DNA sequence analysis

Rafał Biedrzycki[1]

[1] Warsaw University of Technology, Institute of Electronic Systems, Warsaw, Poland,
email: R.Biedrzycki@elka.pw.edu.pl

Abstract. This paper presents a brief survey of computational approaches to the DNA sequence analysis. The basic biological background is presented. The various types of algorithms for pattern construction and gene finding are presented with special attention paid to the application of global optimization methods.

1 Introduction

Bioinformatics is a quite new field of research bringing together biology, mathematics and computer science. One of the main research areas of bioinformatics is genome sequencing and annotation. There is a vast amount of data available to the public, including DNA sequences and annotations. This paper is aimed at introducing the basic concepts of a branch of bioinformatics, which is oriented on analysis of DNA sequences. The use of evolutionary computation for performing the analysis is also discussed.

1.1 Biological background

The basic biological macromolecules: DNA, RNA, and proteins have a form of chains composed of small building blocks. In case of DNA and RNA, there are four such elements, called nucleotides, whereas proteins are composed of twenty aminoacids. Those molecules can be represented as strings over alphabets of four and twenty letters, respectively. Each nucleotide consists of three parts: a base molecule (a purine or a pyrimidine), sugar, and one or more phosphorate groups. The purines are: adenine (A) and guanine (G), and the pyrimidines are cytosine (C) and thymine (T). Every DNA strand has its head called 5' end and its tail called 3' end. To be more stable, DNA strand is connected with other with hydrogen bonds. Adenine bonds exclusively with thymine (A-T) and guanine with cytosine (G-C). The connected strands have opposite directions and are composed of complementary parts called base pairs (bp). In most of the biological systems, DNA forms classic double helix called B-DNA. In certain conditions, it can become supercoiled or even reverse direction of its twist (Z-DNA). It is one of the reasons why the sequence based automatic annotation will never be perfect because there is more information available to the cell than just a pure sequence. Notice also that free parts of DNA or RNA sequence could bind with itself forming loops and other shapes which influence molecular operations performed on the sequence.

DNA is composed of sequences that encode proteins or other cell products (called coding DNA) and the rest (called non-coding DNA). Most of the coding DNA encodes proteins – essential parts of all living organisms. The synthesis of a protein is a two-step

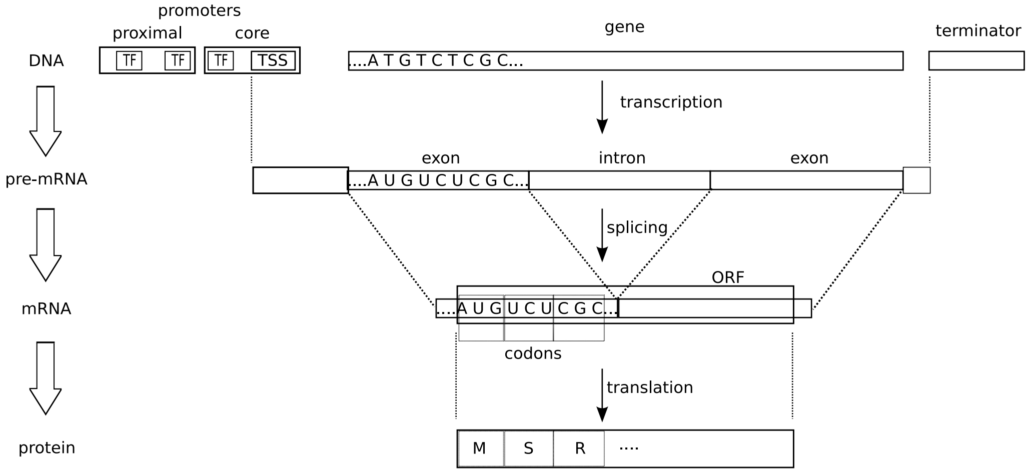

Figure 1. Transcription and translation in eukaryotes

decoding process. The sketch of that process we could see in Figure 1. The fist stage is the transcription from DNA to RNA; the second stage is the translation of RNA to proteins. The initiation of the protein production depends on complicated regulatory mechanisms. Each gene has its own regulatory sequences in addition to explicitly coding region. A universal regulatory region is called promoter which provides region recognized by the transcription machinery. The promoter consists of Transcription Factors (TF) – short sequences that are binding sites for regulatory proteins. Some sites have a positive influence on the initiation of the transcription process (enhancer) and some of them are blocking the transcription (silencers). The gene transcription process starts from the binding the transcription machinery to the promoter region and determining the transcription start site (TSS). Transcription stops when a special stop sequence is met. In higher organisms (eukaryotes) transcription produces pre-mRNA – the sequence that is composed of regions (subsequences) that are responsible for information coding (exons) and those which are not (introns).

The next operation is the splicing process. During splicing, the mRNA is produced by removing introns from the pre-mRNA by the spliceosome; mRNA (messenger RNA) is a sequence that encodes a protein. The next stage of the protein synthesis process is translation. The translation machinery finds an appropriate start codon (codon is a triple of nucleotides) and translation is performed until a stop codon is met. The sequence starting from the start codon and terminated by the end codon is called Open Reading Frame (ORF). The translation is performed codon by codon. Although four letters of DNA/RNA allow for building 64 possible codons, only 20 aminoacids are known. Therefore, some aminoacids are encoded by more than one codon. If such situation takes place, the codons responsible for the synthesis of the same aminoacid usually differ in the last position letter. The frequencies of all codons are not equal in coding sequences. Therefore, since codons are triplets, some regular patterns of a periodical type are reported to occur[11].

1.2 Sequencing

The first successful sequencing of a bacterial organism (H. Influenzae) was performed in 1995; the complete DNA sequence is 1,7 million nucleotides long. The genome of yeast (S. Cerevisiae) has 12 million nucleotides, and the human genome contains 3 billion nucleotides [2, 6, 12]. It may seem surprising but the human genome is not the longest one. The owner of the longest known genome is Amoeba and its length is $6.7 * 10^{11}$. The genomes, after the sequencing, are stored in databases available to the public. One of such databases is GenBank. From its inception, GenBank has doubled in size about every 18 months [1]. At the beginning of the year, 2006 GenBank contained sequences of more than 205000 named organisms, and more that 10^{11} base pairs of sequence data. The process of sequencing genomes from different organisms continues and will probably be continued because all species are interesting for the science (estimates of currently existing species range from 5 million to 50 million, so there is still a lot to be sequenced).

1.3 Annotation

Successful sequencing is not the end of genome project – it is rather the beginning. The second step is gene annotation which consists in finding and describing functional elements in the genome.

So far, the annotation process is performed in a collective way by the community annotation via the Internet [4]. That kind of annotation needs www servers with dedicated tools and supporting experimental evidences. There exist libraries of such evidences which collect DNA that came from the reverse transcription from mRNA, so they are composed of exons only. Such long sequences are called cDNA; short sequences, caused by the errors in reverse transcription, are called EST. Internauts, after reading tutorial, are able to use web tools to put their annotations based on cDNA, EST, and their general knowledge. Those annotations are later accepted or rejected by experts – so called curators of the database. This procedure needs great effort of many experts but still for some annotations no supporting sequences are known.

2 Analysis of DNA sequences

There is a vast amount of different problems connected with DNA sequence analysis but we are able to find something in common. For every problem defined here the underlying task is how to efficiently search through a certain space of solutions and how to escape from the local optima of the objective function defined over that space. This holds even in the problem of classification of DNA sequences where the problem is an efficient search in the space of all possible attributes.

The another problem connected with classifiers is how to asses their quality. One popular approach is to split available data into two subsets: the training set and the test set. Having labeled test set allows to compare predictions of the classifier with the real class values. As a result of that comparisons we get a matrix in which we could find true positives (correctly predicted positive examples), true negatives (correctly predicted negative examples), false positives (predicted positive should be negative), and false negatives (predicted negative should be positive). There is a large number of methods of assessing classifiers performance that are based on elements from that matrix.

2.1 Pattern representation

Every problem connected with the computational DNA analysis, annotation being an example, is at least connected with the issues of pattern representation, building, and matching.

Several notations for describing patterns have been used [10], but most of them are the versions of a standard notation used in regular expressions. The basic pattern notation is just a sequence of characters, each of them denoting a single aminoacid or a group of amino acids. Such short sequences with biological significance are called motifs. The more powerful representation introduces ambiguous symbols. An ambiguous symbol is the expression that allows more than a single nucleotide in certain position. In the commonly used notation, alternatives for considered position are enclosed in square brackets, e.g. A[TC] matches sequences AT and AC. The special case of that patterns are patterns with so called "don't care" symbols or gaps, e.g. AxTxxG. Such symbol represents any letter from the alphabet. A flexible gap is defined by two numbers: the minimum and the maximum number of "don't cares" allowed, e.g. [CG]x(2,3)T matches to the sequences of the length 4 or 5 that ends with T and starts with C or G. That kind of patterns is used, for instance, in PROSITE database (database of protein families described by common pattern). Such patterns are quite powerful but hard to effectively acquire from sequence.

Another approach is a matrix pattern representation. The basic idea is to use a matrix of numbers containing scores for each nucleotide at each position of a fixed-length subsequence. There are two types of weight matrices: a position frequency matrix (PFM) and a position weight matrix (PWM). PFM records the position-dependent frequency of each nucleotide, and PWM contains logarithmic weights for computing a match score.

Patterns are also modeled by Markov Models. The k-order Markov Model is the probabilistic model that takes into consideration k previous elements in the sequence. The higher order of the model, the better ability to describe DNA, but the more parameters to be estimated (so the harder to compute). In practice k ranges from 0 to 5. Unfortunately even such a sophisticated model cannot capture distant relationships which often occur in DNA sequences.

2.2 Gene finding

Gene finding is the process of identification of DNA subsequences that are biologically functional. This especially includes protein coding genes identification, and splice sites detection. There are two types of splice sites: a donor site (start of the intron) and an acceptor site (the intron end).

Most of the algorithms like those described in [11] decompose the splice site detection problem into two tasks: finding a donor site and finding an acceptor site. Saeys [11] proposed an approach based on attributes induction and then classification. He aligned all training sequences to the start in a certain site, e.g. a donor site. After that, he generated position dependent features using the schema: if a nucleotide T is in the right first position of the donor site, then the attribute T1 is set to 1, and attributes A1, C1, G1 are set to 0. In a similar way he also defined a large set of position independent attributes. He also performed the transformation of the DNA sequence into the Fourier spectrum trying to capture periodicity in DNA, and, as the result of that, he get another set of additional attributes. He also used counters of AT/TG dinucleotide percentage upstream, and other features.

Having huge amount of features he used several feature selection techniques to reduce the number of features. After that, he applied a few classification algorithms. Although he defined such large set of attributes, he still did not cover all the important DNA features. DNA and RNA can fold into the secondary structure. That kind of folding depends mainly on complementary matches of distant parts of a sequence which could not be captured by his approach. Defining large set of simple attributes needs a great computational effort. Moreover, a classification algorithm cannot be successfully applied to such large sets of attributes. The feature selection techniques unfortunately do not guarantee that important features will be preserved in the final feature set.

Another popular approach is represented by the Genscan algorithm [3]. Genscan models the gene using finite state automata. Each state corresponds to important functional sequence fragment (intron, exon, promoter, intergenic region and others) and is modeled by the Hidden Markov Model (HMM). This approach has also its drawbacks. Predicted gene number may be incorrect, the algorithm was originally developed for the human/vertebrate sequences, which results in lower accuracy for other types of sequences. Internal exons are predicted more accurately than initial and terminal exons. Another constraints comes form using HMMs, e.g. the algorithm cannot capture long-distant interactions in the analyzed DNA sequence.

2.3 Application of pattern search methods

In [9] the authors face up the problem of constructing planted motifs (motifs of specified length with specified maximal number of point substitutions). A motif is represented by the position weight matrix. In general, the motif scoring function is based on the information contents of that motif and is computed using the matrix. The underlying idea of that algorithm is to find some promising initial motifs by the random search and then to use a local algorithm to find a local maximum of the logarithmic motif scoring function and some other local maxima nearby. The first stage of the algorithm is called a global phase. In that phase, random search with threshold is used (patterns with quality below specific threshold are rejected). The second stage is the refinement stage when a local search algorithm is used to find a local maximum according to the Expectation Maximization (EM) criterion. The third phase is the exit phase when the algorithm determines the exit points from the current local maximum x_{max}. To be able to determine the exit points, the eigenvectors of the Hessian matrix of the scoring function are used as the directions of escape from the local attraction basin. Then for every direction d_i, a direction search method is used to find a minimum $x_{i,\mathrm{min}}$ along that direction in order to leave the attraction basin of the current local maximum x_{max}. For each escape direction, a new local maximization process is started with a start point $x_{i,s} = x_{i,min} + \varepsilon d_i$, where ε is a small positive number. The same methodology is then applied to each local maximum neighboring to x_{max}. The authors report that on average, results obtained by exploring neighboring local maxima are better than original local maximum.

In [7] the author introduces the algorithm Pratt2 for discovering patterns that are compatible with the PROSITE pattern notation. The user has to supply a training sequence, a maximum flexible gap length and some other parameters. The first step of the algorithm is the generation of a pattern graph. The nodes in that graph represent pattern components, and the edges represent wildcard regions. Assume that we have training sequence ATCG so we will build a graph with nodes labeled A, T, C, G. Assume that the maximum gap length is 1, so from the node A we have the following edges: to T labeled 0 (no gap) and to C, labeled 1 (maximum gap length). Analogically we can construct edges from T and from C. The patterns are scored according to their information contents. The pattern search procedure is a recursive, depth-first search that starts from an empty pattern and finds all high scoring patterns conserved in at last k sequences, where k is a user supplied parameter. The pattern is grown by extending the current one, according to the pattern graph.

Let us come back to the example sequence. Assume that pattern AT was grown. That pattern is extended to ATC and ATxG according to pattern graph. Those basic patterns are extended by allowing more flexibility to: ATx(0,1)C and ATx(0,1)G, ATx(1,2)G. In the Pratt2 method, a branch and bound search strategy is applied to prune the search tree. Two techniques have been used to estimate scores of subtrees of the search tree. One, slower method is based on dynamic programming and the other, a faster one, on the results seen so far, e.g. if during the search we have seen the node V and now we are in node U, and if it is possible to extend the pattern with the node V, we could at most achieve pattern with a score that is the sum of its components. If that score is lower than the current best score we could prune this search path. Another interesting idea of the authors is greedy postprocessing algorithm. The patterns produced by the main algorithm can be refined by the substitution of wildcard positions by an ambiguous pattern components, e.g. pattern AxG could be replaced by A[AT]G. The refinement is accepted when the resulting pattern still matches at last k sequences. The Pratt2 algorithm was used to find patterns that characterize protein families. They use 1148 sets of sequences from the PROSITE database. For the 28 sets PRATT2 failed to allocate sufficient memory. Computations for the hardest set („three very long sequences”) took 50 minutes.

3 Global optimization for DNA pattern analysis

In previous chapter, gene finding problem was introduced. One of the attempts to solve that problem was based on classification of sequences. A representative approach that falls into that category was introduced by Saeys [11]. He generated a large number of attributes and tried to find a reasonable number of good attributes. The underlying problem is how to effectively search the space of possible subsets of attributes. Since it can be expected that the objective function has numerous local minima, it is possible to use a global optimization method. Saeys chose Estimation of Distribution Algorithm (EDA) for this task. In his approach, an individual is represented as a binary string of the length equal to the number of features. Zero or one at specific position in that string means that the attribute which corresponds to this position is excluded or included into the subset of attributes. To speed up the computations the author constrained the upper limit of the size of feature subset. The resulting subset of features is evaluated by training and testing the classification model. The fitness function is proportional to the classification accuracy. The author extended the algorithm by deriving a feature ranking from a probability distribution. Features having higher value of probability are considered more important. Saeys used his method to analyze thousands of potential features to determine which of them have the strongest impact on the classification ability and he interpreted them as a biologically meaningful attributes. He reported that when he used the best 10% of features ranked by EDA based technique the classification accuracy was 10% better than those achieved using by replacing EDA by a Weighted Naive Bayes Method (WNBM).

A more straightforward approach to the bio-pattern analysis is represented by [10]. The author tackles the problem of finding a sequence pattern common to a particular protein family. The author first introduces a stochastic regular expressions language for DNA (SRE-DNA). The language allows for ambiguity and flexible gaps; besides, every element in the pattern is assigned a probability (e.g. A^{+p} means that sequence consists A, and A may be repeated with probability p). For each sequence it is possible to compute the probability that the sequence matches the pattern. The scoring function of the pattern is computed as a difference between the sum of probabilities of fitting to the positive examples (protein sequences from the considered family) and the sum of fitting probabilities for negative ones. The author uses grammatical genetic programming system (DCTG-GP) to evolve motifs. He provides simple grammar by defining what is expression, guard and skip (e.g. $expression ::= guard|guard : expression|expression^{+p}$). An evolutionary algorithm searches the space of motifs constrained by the grammar rules and some additional constraints (e.g. iteration ranges). The basic parameters of the algorithm are: generations - 100; population size 2000; tournament size - 7; probability of crossover - 0.9; probability of mutation - 0.1. The ratio of true positive examples is between 67 and 100% depending on protein family and grammar variation used. Unfortunately there is no comparison to the result of other algorithms. The author does not provide algorithm speed, stated only that only six runs were performed because the algorithm was slow.

In [5] the authors tackle the problem of generating regular expression based classifiers that classify proteins into two classes: those proteins that will be transported into nucleus, and the others. They tried genetic programming with a vector based and a tree based representation. They used population of 2000 individuals and a tournament selection scheme. The Matthews Correlation Coefficient (MCC) was used as a fitness function: $MCC = \frac{tp*tn-fn*fp}{\sqrt{(tn+fn)(tn+fp)(tp+fn)(tp+fp)}}$. True positives ($tp$) were correctly predicted positive examples, true negatives (tn) were correctly predicted negative examples, and so on. The authors wrote that after some improvements of the algorithm the training process may take no more than 36 hours. The authors conclude that the linear representation is as good as the tree based one considering accuracy, and it allows for faster execution of the algorithm.

Koza wrote a series of articles connected with bioinformatics, e.g. [8]. He and his coauthors were interested in the problem of classification of proteins to their cellular location. The base of that problem is to find appropriate motifs. They introduce term ,,programmatic motif" that is an extension of the conventional concept of a motif. The programmatic motif is the program produced by genetic programming using arithmetic functions, conditional operations, logical operations, named and indexed memory, iteration and recursion. They also use MCC as a fitness measure. They use genetic algorithm with the population of 320000 individuals distributed between 64 demes. They write that they chose the largest population size that could be run and stored in a reasonable amount of time and in the available memory space. Each run of the program took about 3 days (year 1998 - 64 of 80MHz Power PC 601 processors). The number of generations they used was from 7 to 26. They report 83% accuracy comparing to 76% achieved by human-created algorithm.

4 Conclusions

The common problems with articles connected to DNA analysis is the fact that the reported results are hardly comparable, since no agreement is made about the benchmark problems. Sometimes the point is in a slight different problem formulation or result representation. As for the evolutionary computation based approaches mentioned above, common drawback is the extremely long computation time they need to get a solution.

For all the bio-problems defined here the underlying task is how to efficiently search through a certain space of solutions and how to escape from the local minima of the objective function defined over that space. That problems is deeply studied in the community of scientists interested in evolutionary algorithms and global optimization. There is still possibility to improve existing algorithms and to develop new ones. There is a huge amount of real-word data just waiting for use.

Bibliography

[1] D.A. Benson *et al.* Genbank. *Nucleic Acids Research*, 34:D16, 2006.

[2] A. Brazma *et al.* Pattern discovery in biosequences. In *ICGI*, pages 257–270, 1998.

[3] C.B Burge and S. Karlin. Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.*, 8(3):346–354, 1998.

[4] C.G. Elsik *et al.* Community annotation: procedures, protocols, and supporting tools. *Genome Res*, 2006.

[5] A. Heddad *et al.* Evolving regular expression-based sequence classifiers for protein nuclear localisation. In G.R. Raidl *et al.*, editor, *EvoWorkshops2004*, volume 3005 of *LNCS*, pages 31–40, Coimbra, Portugal, 2004. Springer.

[6] L. Hunter, editor. *Artificial intelligence and molecular biology.* American Association for Artificial Intelligence, Menlo Park, CA, USA, 1993.

[7] I. Jonassen. Efficient discovery of conserved patterns using a pattern graph. *Computer Applications in the Biosciences*, 13(5):509–522, 1997.

[8] J. Koza *et al.* Using programmatic motifs and genetic programming to classify protein sequences as to extracellular and membrane cellular location. In V.W. Porto *et al.*, editor, *Evolutionary Programming VII*, volume 1447, San Diego, California, USA, 25-27 1998. Springer.

[9] C.K. Reddy *et al.* Refining motifs by improving information content scores using neighborhood profile search. *Algorithms for Molecular Biology*, 1:23+, 2006.

[10] B.J. Ross. The Evolution of Stochastic Regular Motifs for Protein Sequences. *New Generation Computing*, 20(2):187–213, 2002.

[11] Y. Saeys. *Feature selection for classification of nucleic acid sequences.* PhD thesis, Ghent University, Belgium, 2004.

[12] Wikipedia. Genome — wikipedia, the free encyclopedia, 2007. [Online; accessed 26-February-2007].