Identification of protein patterns in nucleic acid sequences and exploration of synonymous codons in tissue differentiation

G. Anogianakis[1] E. Kapritsos[2], Ch. Makris[2], N. Mpaltas[2], K. Perdikuri [2,3], K. Themelis[2] and A. Tsakalidis [2,3]

[1] Department of Physiology, Faculty of Medicine, University of Thessaloniki, 54124 Thessaloniki, Greece
[2]Computer Engineering & Informatics Dept., University of Patras 26500 Patras, Greece
perdikur@ceid.upatras.gr
[3]Research Academic Computer Technology Institute 61 Riga Feraiou Str., 26221 Patras, Greece

Abstract

In this work we present an automaton, which searches for protein patterns in nucleic acid sequences. Our program allows the exploration of usage of synonymous codons in gene expression and can serve as a tool in the study of tissue differentiation.

Keywords: Aho-Corasick Automaton, Pattern Matching, Synonymous Codons, Biological Sequences, Tissue Differentiation.

1. Introduction

DNA and protein sequences can be seen as long texts over specific alphabets encoding the genetic information of living beings. In the case of DNA sequences the alphabet consists of the four nucleotides, Σ_{DNA} = {a,c,g,t}, while in the case of protein sequences, the alphabet consists of the twenty amino acids. Codons, constitute triplets of nucleotides that code the amino acids, the "words"' in this 4-letter language. One of the corollaries of the DNA coding mechanism is that DNA sequence alterations in a gene can change the structure of the protein that it codes for.

Indeed, given the fact that the genetic alphabet uses four letters and that each word is three letters long we would expect to have sixty four different words. But the language of genetics contains twenty one "semantically different words"' (i.e.: the twenty amino acids and a "terminating codon"' which serves as the punctuation mark). Thus it is evident that there must be many different ways of "spelling"' for at least some, of the words of the language of genetics. The actual correspondences are tabulated in Table 1. For example the codon UCC codes for the amino acid Serine, CCU for Proline etc, but also CCC, CCA and CCG all code for Proline. Since the codons: {CCU, CCC, CCA, CCG} code the same aminoacid they are called **Synonymous Codons**. According to the recent literature Synonymous Codons have significant implications in tissue differentiation and evolution.

Table 1. The 64 possible combinations of the four bases and the codons they represent. X stands for the terminating codon.

1ST BASE	2ND BASE				3RD BASE
	U	C	A	G	
U	PHE	SER	TYR	CYS	U
	PHE	SER	TYR	CYS	C
	LEU	SER	X	X	A
	LEU	SER	X	TRP	G
C	LEU	PRO	HIS	ARG	U
	LEU	PRO	HIS	ARG	C
	LEU	PRO	GLN	ARG	A
	LEU	PRO	GLN	ARG	G
A	ILE	THR	ASN	SER	U
	ILE	THR	ASN	SER	C
	ILE	THR	LYS	ARG	A
	MET	THR	LYS	ARG	G
G	VAL	ALA	ASP	GLY	U
	VAL	ALA	ASP	GLY	C
	VAL	ALA	GLU	GLY	A
	VAL	ALA	GLU	GLY	G

The genomes of species from bacteria [1] to Drosophila [2] show unique biases for particular synonymous codons and, recently, it was shown that such codon preferences exist in mammals [3]. Systematic differences in synonymous codon usage between genes selectively expressed in six adult

human tissues were reported, while the codon usage of brain-specific genes is, apparently, selectively preserved throughout the evolution of human and mouse from their common ancestor [3]. In particular, when genes that are preferentially expressed in human brain, liver, uterus, testis, ovary, and vulva were analyzed, synonymous codon biases between gene sets were found. The pairs that were compared were brain-specific genes to liver-specific genes; uterus-specific genes to testis-specific genes; and ovary-specific genes to vulva-specific genes. All three pairs differed significantly from each other in their synonymous codon usage raising the possibility that codon biases may be partly responsible for determining which genes are expressed in which tissues. Such a determination may, of course, take place at the level of transcriptional control. However, given the relatively low number of genes identified in the human genome, "vis-à-vis" our preconceptions about human structure and function, one is tempted to explore whether other control mechanisms operate (alone, in tandem or in parallel with transcriptional level mechanisms) in tissue differentiation [4].

Assuming that some kind of parsimony principle governs the development of control mechanisms at the DNA transcription level (a very strong but necessary assumption that is required in order to limit and focus the subsequent discussion), there are two obvious mechanisms that can be used to link synonymous codon choice and tissue-specific gene expression:

- The first mechanism depends on local transfer RNA abundance. The tRNA pools in the brain, e.g., may differ from the pools in liver, and so if the codon usage of a gene is calibrated to the tRNA pools that exist in the brain, that gene will be translated more efficiently in brain.
- The second mechanism would make use of the different chemical affinities between different tRNAs coding for the same amino acid and the underlying DNA structure. In other words, if certain codons have larger affinities with their corresponding tRNAs than their synonymous codons, it stands to reason that they will be expressed more readily. A corollary of this argument is that differentiation will occur when the appropriate genes find themselves in an energetically appropriate environment to be expressed.

In order to resolve the question of whether the second mechanism that was proposed as the link between synonymous codon choice and tissue-specific gene expression has any theoretical (or practical) merit, it is necessary to, first, associate each codon with a value reflecting its potential for expression. To illustrate this point let us assume, e.g., that a gene is coding a peptide with the sequence:

ALA-ALA-ALA-ALA-ALA-ALA-ALA-ALA-ALA-ALA.

Let us, further, assume that codon GCU has twice the affinity for its corresponding tRNA than codon GCC has for its own corresponding tRNA. Similarly, codon GCC has twice the affinity for its corresponding tRNA than codon GCA has for its own corresponding tRNA and, finally that codon GCA has twice the affinity for its corresponding tRNA than codon GCG has for its own corresponding tRNA. It is evident that, if chemical affinities alone were the determining factor and if gene expression was a linear function of the product of its codons' affinities for their corresponding tRNAs, then a gene represented by the sequence:

GCUGCUGCUGCUGCUGCUGCUGCUGCU,

would be expressed 2^{30} times more readily than the gene represented by the sequence:

GCGGCGGCGGCGGCGGCGGCGGCGGCG,

despite the fact that these two hypothetical genes are synonymous.

The example above serves as an illustration of the approach that is required to resolve the problem of amino acid coding redundancies, codon synonyms and their possible involvement in tissue differentiation.

In this work we have developed an automaton, which searches for protein patterns in DNA fragments and further explores the usage of synonymous codons in gene expression serving as a tool in the study of tissue differentiation.

The structure of the paper is as follows. In Section 2 we give some basic definitions needed through the paper while in Section 3 we present the methodology followed. Finally in Section 4 we conclude and discuss our future research in the area.

2. Preliminaries

We start with a formal definition of the problem outlined in the above paragraph.

Problem. *Given a DNA fragment x, find whether a protein p is encoded inside the fragment and the exact codons which code the protein.*

Although the above definition looks like a trivial pattern matching problem this is rarely the case, since as we already mentioned a protein sequence may have more than one equivalent encodings based on the use of synonymous codons. Thus in our problem the given pattern of a protein may have more than one representations, which are all equivalent.

A simplistic solution of the problem would be to produce all possible encodings of a given protein $p=\{p_1, p_2, \ldots p_n\}$ using synonymous codons and reducing the problem to a *multi-pattern matching* problem. Taking into consideration that the length of a protein may have up to 300 amino acids and that every amino acid is encoded with at most six different ways this could yield to 6^{300} possible patterns. Such an approach is not efficient having in mind that a DNA

fragment could be millions of nucleotides long. Moreover in most problems of Bioinformatics and Computational Molecular Biology we seek for efficient solutions both in space and time. Thus we have to come up with a more efficient solution for the above problem.

One could argue that the above problem could be easily solved using a variation of the programs such as BLAST or FASTA, two widely used tools for searching protein and DNA databases for sequence similarities. Generally BLAST programs have been written to compare protein or DNA queries with protein or DNA databases in any combination, with DNA sequences often undergoing conceptual translation before any comparison is performed [5]. This translation step is the basic object of our work, reporting at the same time the use of synonymous codons. In our methodology to be presented in the following section we build a sequential Aho-Corasick automaton to solve the previously defined problem. The Aho-Corasick automaton is a deterministic finite automaton together with a failure function. A formal definition follows.

An Aho-Corasick automaton is a six-tuple $(Q, \Sigma, g, f, q_0, F)$, where:

- Q is a set of states;
- Σ is a finite input alphabet;
- $g : Q \times \Sigma \rightarrow Q \cup \{fail\}$ is the forward transition;
- $f : Q \rightarrow Q$ is the failure transition;
- q_0 is the initial state;
- F is a set of final states and is a subset of Q.

The failure function has the following property: Suppose that in the transition graph of the g function, state q_i represents the string s_i and state q_j represents the string s_j. Then, $f(q_i) = q_j$ if and only if s_j is the longest proper suffix of s_i that is also a prefix of some keyword $x \in X$. The search procedure in an Aho-Corasick automaton is defined by the forward transition function for moving from one state to another according to the current symbol being read from the input text.

3. Methodology

As previously described a simplistic solution to our problem would be to produce all possible encodings of a protein pattern and reduce the problem to a multi-pattern matching problem. But such an approach is not efficient in practice due to the exponential number of possible patterns. Thus we have to come up with a different approach.

3.1. Naïve Approach

A naive approach would be to translate the given DNA fragment in the equivalent protein sequence. In more detail we have to "read"' the DNA fragment in reading frames of *length* = 3 (probably skipping nucleotides at the beginning to produce the required amino acids) and check if the given protein sequence appears inside the fragment. The following example clarifies the naive approach.

Example. *Given the DNA fragment X=AUGCAUAGGCUACUCUAG, find whether protein y = CysIleGlyTyrSer is encoded inside the fragment.*

Solution:

Step-1: Read "AUG"=MET ≠ Cys → skip the first character;
Step-2: Read "UGC"=Cys → 1st amino acid found, proceed a reading frame;
Step-3: Read "AUA"= Ile → 2nd amino acid found, proceed a reading frame;
Step-4: Read "GGC"= Gly → 3rd amino acid found, proceed a reading frame;
Step-5: Read "UAC"= Tyr → 4th amino acid found, proceed a reading frame;
Step-6: Read "UCU"= Ser → 5th amino acid found, report "Success"'

The drawback of the above approach is that in a case of a mismatch the process starts again by skipping only one character, since we do not know where the starting coding position is located inside the DNA fragment. This naïve approach is not efficient in practice.

3.2. A sequential Aho-Corasick Automaton

As we have already mentioned we aim at locating in a DNA fragment x the places where a given protein p appears. We will describe how to solve this problem by employing the Aho-Corasick automaton. The fact that each of the amino acids that constitute a protein can be coded with many different words (triplets of bases), makes the application of the Aho-Corasick automaton a non-trivial procedure. We choose to proceed as follows: assume that p is composed from the amino acid sequence $a_1, a_2, \ldots, a_t$; for every amino acid a_i we create an Aho-Corasick automaton A_i that accepts each of the code words that code a_i and we connect every final state of the automaton with an ε-transition to the initial state of A_{i+1}.

The initial state of the produced automaton is the initial state of A_1 while the final states of the automaton are the final states of A_t. Having built the automaton we can place as its input the string

representation of the DNA fragment x and traverse the various states by reading sequentially the characters of x. If we reach the final state while reading the k-th character of x, we report that the amino acid sequence has appeared at the *k-3t* position.

By maintaining appropriate auxiliary information during the transitions from one automaton to the other we can also produce statistical data concerning the different code words that codify the various amino acids. Moreover, we can assign to the occurrence of every coding triplet a weight depicting its thermodynamic significance thus testing the initial hypothesis that synonymous codons point towards a thermodynamic theory of tissue differentiation.

The space complexity of the whole automaton is O(t) while the time complexity is $(O(|x|^2))$ where |x| is the length of the DNA fragment x. In order to speedup the algorithm we chose in our implementation to replace the ε-transitions from a final state of A_i to the initial state of A_{i+1}, with the transitions that exist from the initial state of A_{i+1} to the next state/states of A_{i+1}; thus making the storage of the initial state of A_{i+1} obsolete. This procedure is depicted in the following figure, where we link the final states of the automaton for the amino acid Ile with states of the automaton for the amino acid Ser.

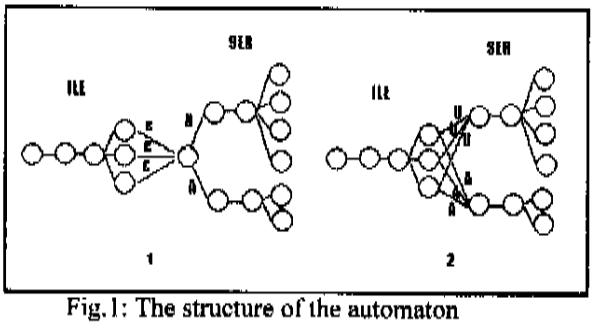

Fig. 1: The structure of the automaton

Let us now consider the time complexity. One of the main ingredients of the classical Aho-Corasick algorithm is the use of the failure function. The failure function follows the logic that in the processed sequence there could exist a smaller sequence that lead to another path of the tree (where the tree root is obviously the initial state of the automaton). So, instead of rescanning the whole input sequence, we can move, with the use of the appropriate failure function, to the state where we can continue reading the sequence from our previous stop point. In this way every symbol of the input sequence is scanned only once and the time complexity becomes linear to the number of characters of the input sequence.

In our use of the automaton the construction of such a function is not possible, since more than one possible query paths can lead to a given state. Hence no failure function can exist since we do not know which the already read characters are. It could be however possible to create multiple failure transitions for every state according to the path that we have traversed in order to reach our point; but the number of these transitions is exponentially increased as we move from one amino-acid to the other and so a choice like that could be disastrous. Hence, in our implementation we choose not to use any failure transition; we just maintain the information for the first time that we met one of the amino acids that codify the first amino acid in the input sequence. In this way we escape useless scans on the input sequence and improve the average case in the time complexity of the algorithm. However asymptotically the search time is still bounded by $O(|x|^2)$.

4. Conclusions

In this paper we have reduced a problem taken from the Computational Molecular Biology era into a pattern matching problem and presented an efficient solution for the identification of protein patterns in nucleic acid sequences using a sequential automaton. Our future research is focused in the experimentation with various data sets in order to test the initial hypothesis and in the incorporation of a dynamic failure function in the automaton (which will be computed online according to the read characters), to improve the time complexity of our algorithm.

5. References

[1] T. Ikemura, "Codon usage and tRNA content in unicellular and multicellular organisms", *Mol Biol Evol* **2(1)**, pp. 13-34, 1985.

[2] J.R. Powell and E.N. Moriyama, "Evolution of codon usage bias in *Drosophila*", *PNAS* **94** pp. 7784-7790, 1997.

[3] J.B. Plotkin, H. Robins and A. J. Levine, "Tissue-specific codon usage and the expression of human genes". *PNAS* **101**, pp. 12588-12591, 2004.

[4] G. Anogianakis, A. Anogeianaki, V. Papaliagkas, "Do Synonymous Codons Point Towards a Thermodynamic Theory of Tissue Differentiation", *In the Proc. of the International Conference of Computational Methods in Science and Engineering 2004*, pp. 692-695, 2004.

[5] S. Altschul, T. Madden, A.Schäffer, J. Zhang, Z. Zhang, W. Miller and D.Lipman, „Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Research*, **25(17)**, pp. 3389-3402, 1997.