

# Bridging P Systems and Genomics: A Preliminary Approach

Solomon MARCUS

Romanian Academy, Mathematics  
Calea Victoriei 125, București, Romania  
E-mail: `solomon.marcus@imar.ro`

**Abstract.** Bringing genomics within the framework of P systems could give to the former the possibility to take profit of the computational capacities of the latter. Moreover, suggestions coming from genomics could enrich the study of P systems with new biological and computational ideas. In what follows, a first attempt is made in this respect.

## 1 “Life is a Surface Activity”

“Life is a surface activity[. . .]. Life is fundamentally about insides and outsides” [5: 260]. Relevant parts of the environment are internalised as an “inside exterior” or “inner outside” (the so-called Uexküll’s *Umwelt* [8]; “the representation of certain environmental features inside an organism by various means” [8: 28]), while the interior becomes externalised as an “outside interior” or “outer inside”, in the form of the “semiotic niche” ([4: 40]), as informed and changed by the inside needs of the organism pertaining to that niche [3: 29]. This inside-outside interplay is made possible by the membrane strictly governing the traffic between them. P systems [7] find their starting point in this biological reality, to which a computational dimension is added. In agreement with the ideas of DNA computing and membrane computing, Wolfram [9] proposed recently to see life as a universal Turing machine, to which Chaitin [2] adds the condition of a high program-size complexity. The project of bridging genomics and P systems could have the slogan: *Life is DNA software + membrane software*.

## 2 P Systems and the Human Genome Project (HGP)

The HGP, as presented, in its computational aspect, by Karp [6], is a good starting point for the problem raised in the title of this article.

Let us recall the notion of a P system in one of its standard representations [1: 18]. A *P system with replicated rewriting* is a construct  $\Pi = (V, T, \mu, M_1, M_2, \dots, M_m, R_1, R_2, \dots, R_m)$ , where  $V$  is an *alphabet* (its elements are called *objects*);  $T$  is contained in  $V$  and it is called the *output alphabet*;  $\mu$  is a *membrane structure* consisting of  $m$  membranes (or regions of a membrane) labeled by  $1, 2, \dots, m$ , such that each membrane except the first is completely contained within another;  $M_1, \dots, M_m$  are finite languages over  $V$ ;  $R_1, \dots, R_m$  are finite sets of *developmental rules* of the form  $a \rightarrow (v_1, tar_1) \parallel \dots \parallel (v_n, tar_n)$ , with  $n \geq 1$ , where  $tar_i$  belongs to  $\{here, out, in\}$ ,  $a \in V$ , and  $v_i \in V^*$ ,  $\leq i \leq n$ . The languages  $M_i$  and the rule sets  $R_i$  are associated with the regions of  $\mu$ , for all  $1 \leq i \leq m$ . When a rule of the form above is used to rewrite a string of the form  $xay$ ,  $n$  strings  $xv_iy$  are obtained and sent to the regions indicated by  $tar_i$ . When  $tar_i = here$ , the string is kept in the same region. When  $tar_i = out$ , the string leaves the current region and goes into the outer one, which for region 1 means out of the system. When  $tar_i = in$ , the string is sent to one of the directly included regions, if any exists, otherwise the rule cannot be applied. A computation is defined as follows: the process starts with the strings present in the initial configuration and proceeds iteratively by applying in parallel the rules in each region to all strings that can be rewritten. If more than one rule can be applied to the same string in the same region, then only one, randomly chosen, rule will be applied. If the chosen rule can be applied in several places of the string, then it is applied in only one, randomly chosen, place. The result, the set of all terminal strings, is collected outside the system, at the end of the halting computation. The language generated by a system  $\Pi$  is denoted by  $L(\Pi)$  and consists of all strings over  $T$  that are sent out of the system during a halting computation.

Our option for the variant investigated in [1] is motivated mainly by the fact that it distinguishes between the input and the output alphabet.

### 3 From Genomics to P Systems and Back

Let us recall that the genome of an organism is its total content of DNA molecules within the chromosomes. Each species has its genome characteristics and each individual within a species has its specific features. The human genome includes about three billion base pairs and about 35,000 genes. Our aim in the following is to identify the ways genomics, i.e., the study of genome, may lead naturally to some P systems. To the extent to which this task is fulfilled, the computational aspects of genomics may take profit from the computational capacities of P systems.

The usual, starting interpretation of the objects forming the alphabet of a P system is to consider them as molecules. The general theory of P systems does not depend on the way we interpret these objects; however, the intuitive representation of them decides to a large extent the type of problems which are investigated.

According to Karp [6], the main problems of genomics are: (a) to sequence and compare the genomes of different species (to sequence a DNA means to decompose it in its successive nucleotide bases); the sequencing of the human genome began in 1990 and was essentially completed in February 2001; (b) to identify the genes and determine the functions of the proteins they encode. Task (a) is mainly of a syntactic nature, while task (b) refers to the semantic dimension of cellular processes. Some other tasks of the HGP were considered, but we leave them aside now.

A natural question arises: Which are (if they exist) the P systems accounting for the above tasks (a) and (b)? Referring to P systems with replicated rewriting, a first idea is to work with an alphabet  $V$  including both the types of nucleotide bases and the types of amino acids, while the output alphabet  $T$  contained in  $V$  will be the set of various types of amino acids. The P system we are looking for should describe the process leading from DNA to its segmentation in nucleotide bases, from this segmentation to the identification of genes, which are privileged substrings of DNA, carrying the genetic information, and finally from genes to protein functions (the latter being hypothetically related to the protein sequencing, i.e., to their decomposition in amino acids). So, the membrane structure should consist of several regions, such as: a region of nucleotide bases, a region of genes, a region of amino acids, a region of DNAs, a region of proteins, all of them contained in the initial region represented by the cell. We are already faced with a necessary extension of the relation “contained in”, used in the definition of a P system. Besides its usual meaning, when we refer, for instance, to the fact that DNA is included in the cell, we consider also the substring–string relation, as a variant of “contained in”, accepting so that the region of nucleotide bases is contained in the region of DNA (meaning that any element of the former region is a substring of an element of the latter); similarly, the region of genes is contained, in this acceptance, in the region of DNAs; the region of amino acids is contained in the region of proteins, while the region of codons is contained in the region of RNAs and all are contained in the cell.

Another aspect deserving a reconsideration is the interior–exterior distinction, involved in the structure of a P system. In the light of the ideas exposed in the first section, it should be replaced by a four-steps organization: interior, exterior interior, interior exterior, and exterior, according to Hoffmeyer’s approach. This means that some formal rules should be identified in order to distinguish a living system from its Umwelt, its Umwelt from its semiotic niche – sometimes called the *ecological niche* – and the ecological niche from its environment.

### 4 A Difficult Task: The Developmental Rules

The most difficult task is to identify the developmental rules associated with the considered regions. Just in this respect, the work done within the framework of HGP is essential [6]. The rules leading to the decomposition of DNA in nucleotide bases are of a chemical nature and recall the phonemic segmentation problem in descriptive linguistics. The rules identifying the genes are a mixture of chemistry, biology, and combinatorial and comparative operations; they lead to approximations and to probabilistic statements, rather than to deterministic exact statements. In this respect, a measure of similarity between strings over the input alphabet or over the output alphabet is needed. This task is fulfilled by the introduction of the notion of an alignment of a pair of strings  $\langle x, y \rangle$  as a new pair  $\langle x', y' \rangle$ , where  $x'$  and  $y'$  have the same length and  $x'$  is obtained from  $x$  and  $y'$  is obtained from  $y$

by inserting occurrences of the special space symbol  $-$ . For instance, if  $x = acbcd$  and  $y = abbdcdc$ , then a possible alignment of  $x$  and  $y$  is given by

$$\begin{aligned}x' &= a-cbc-db, \\y' &= ab-bdcdc.\end{aligned}$$

A symmetric scoring function  $f$  is defined, that maps pairs of symbols from the alphabet  $\{a, b, c, d, -\}$  to the real numbers. In respect of  $f$ , each individual column from the eight columns appearing in the representation of  $x'$  and  $y'$  has a score. The total score of the considered alignment is, by definition, the sum of scores for all columns; it expresses the similarity between  $x$  and  $y$  in respect to  $f$ . In order to make the total score, as an adequate measure of similarity between  $x$  and  $y$ , it is necessary to make a right choice of the mapping  $f$ . For instance, it is natural to select for  $f(\langle u, u \rangle)$  a value strictly higher than zero, for any object  $u$  different from  $-$  in the alphabet, because matched symbols must increase the score of the alignment. However, one takes  $f(\langle -, - \rangle) = 0$ . It is also natural to oblige  $f(\langle u, - \rangle)$  to be strictly negative, for any symbol  $u$  different from  $-$ , in order to penalize misalignments. When  $a, b$  are amino acids,  $f(\langle a, b \rangle)$  indicates the frequency with which  $a$  replaced  $b$  in evolutionarily related strings. The global alignment problem asks for the optimal alignment of two strings  $x$  and  $y$  in respect to a given scoring function. As a measure of similarity, *optimal* means here the highest value possible, in contrast with other measures, which are looking for the size of dissimilarity, such as the Hamming distance.

For strings which are not globally similar, a kind of local alignment is investigated, which is weaker than the global one. The idea is to look for the alignment between consecutive substrings, chosen as desired, of  $x$  and  $y$ . From two strings one can move to  $n$  strings and define their multiple alignment, the score being the sum of the scores of its induced pairwise alignments. The problem to find a maximum-score multiple alignment of a set of strings is **NP-hard** (it is not the case for  $n = 2$ ). See Karp [6: 547] for more details.

## 5 Exons, Introns, and Codons

Since gene finding and determining the functions of the proteins they encode were a basic task of HGP, discovering the rules in the P system accounting for them appears to be important. In this respect, we should perhaps distinguish between prokaryotes (whose cells do not have a distinct nucleus) and eukaryotes, whose cells include nuclei and organelles. In the former case, each gene consists of a single contiguous string of nucleotide bases. Things are more interesting in higher eukaryotes, where a gene consists of two or more substrings called exons, that code for parts of a protein; exons are separated by introns, which are noncoding substrings. Some rules in the hypothetical associated P system should register the process of alternative splicing; the different possibilities of parsing a gene into exons and introns and the way the same gene can code for different proteins. In the transcription process from DNA to RNA, the string of exons and introns is transcribed into a pre-mRNA transcript, after which the introns are removed and the exons are spliced together to form the mRNA that leads to a ribosome, which in its turn is translated into protein.

Should exons and introns be included in the input alphabet of the P system and amino acids and protein functions into the output alphabet? Should these types of entities lead to different regions of the P system? Obviously, the input-output distinction corresponds here to the syntactic-semantic distinction, where *syntactic* is related to processes anterior to RNA-protein translation, while *semantic* is related to proteins and their functions. The input alphabet should also include all the signals indicating the exon-intron boundaries, as well as the beginning of the first exon and the end of the last exon of a gene. The entities involved in this operation are 64 types of codons (substrings of length equal to three), which correspond by the genetic code (dictionary) to twenty types of amino acids. Like morphemes in natural languages, which can be lexical or grammatical, some codons (whose number is 61) code for different types of amino acids, while three of them (TAA, TAG, TGA) are stop codons; they indicate the end of the translation process. The codon ATG is both lexical and grammatical, depending on its position; it may code an amino acid, but also the start of an exon. The complete picture of distribution of codons within exons and introns and of the

distribution of nucleotide bases in respect to exon–intron boundaries is a mixture of deterministic and statistic aspects. The dynamic-programming Viterby algorithm gives the most likely evolution [6].

## 6 Phylogenetic Trees

Besides the problem of the structure of genomic sequences there is the problem of the evolution in time of a genetically related group of organisms. A P system depending on the parameter time should account for this process, where we are dealing with a phylogenetic tree whose leaves represent the existant species, while the internal nodes represent some postulated speciation events in which a species divides into populations that follow separate evolutionary paths and become distinct species. Karp [6: 540] makes clear the difference in approaching the evolutionary tree before and after the era of genomics. Before this era, each species was described by means of some morphological characteristics, such as presence or absence of hair, fur, number and type of teeth, etc. Within the framework of genomics, the trees are mainly constructed by comparison of related DNA or protein sequences in the considered species, where for each species and each character a character state is given. Under this second aspect should phylogenetic trees be considered in the P systems perspective. Species with similar character states should be close together in the tree. We reach in this way a problem of optimisation of a distance in the tree. Irrespective the way in which this optimisation problem is formulated, it proves to be **NP**-hard [6]. For instance, one can define the distance between two species as the sum of the lengths of the edges on the path between the two species in the tree. This fact gives rise to an opposite problem: Given a distance function  $d$  defined on pairs of species, construct a corresponding tree and a set of edge distances, such that the resulting distance approximates  $d$  as closely as possible. Phylogenetic trees should form a distinct region in the associated P system, while distances and their optimisation in these trees should code significant facts in the output alphabet.

We are only at the first steps of a problem that could involve a lot of technical difficulties.

## References

1. J. Aguado, T. Bălănescu, T. Cowling, M. Gheorghe, M. Holcombe, F. Ipate, P systems with replicated rewriting and stream Eilenberg machines, *Fundamenta Informaticae*, 49, 1-3 (2002), 17–33.
2. G.J. Chaitin, Meta-mathematics and the foundations of mathematics, *Bulletin of the EATCS*, 77 (June 2002), 167–179.
3. C. Emmeche, K. Kuhl, F. Stjernfelt, Reading Hoffmeyer, rethinking biology, *Tartu Semiotic Library* 3, Tartu University Press, 2002.
4. J. Hoffmeyer, Surfaces inside surfaces, *Cybernetics and Human Knowing*, 5, 1 (1998), 33–42.
5. J. Hoffmeyer, The biology of signification, *Perspectives in Biology and Medicine*, 43, 2 (2000), 252–268.
6. R.M. Karp, Mathematical challenges from genomics and molecular biology, *Notices of the American Mathematical Society*, 49, 5 (May 2002), 544–553.
7. Gh. Păun, *Membrane Computing. An Introduction*, Springer-Verlag, Berlin, 2002.
8. J. Uexküll, The theory of meaning, *Semiotica*, 42, 1 (1982) [1940], 25–82.
9. S. Wolfram, *A New Kind of Science*, Wolfram Media Inc., 2001.