

Formal Language Model for Transcriptome and Proteome Data Integration

Reginaldo Inojosa da Silva Filho[1], Ricardo Luis de Azevedo da Rocha[2], and Claudio Santos Oliveira[1(✉)]

[1] Campus de Ponta Porã, Universidade Federal de Mato Grosso do Sul, Ponta Porã, Mato Grosso do Sul, Brazil
reginaldo.uspoli@gmail.com, claudio.santos@aluno.ufms.br
[2] Departamento de Engenharia da Computação, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil
rlarocha@usp.br

Abstract. We present in this article, for the area of structural and functional genetics, a preliminary theoretical model based on the representation of transcriptomes and proteomes as families of formal languages, in which the phenomenon of translation is described as a artificial language transduction process (present in programming languages compilers), making it possible to unify the transcriptomic and proteomic data.

Keywords: Calculi · Computational simulation · Transcriptome · Proteome · Formal languages

1 Introduction

Computer simulation is a fast and flexible virtual process that can significantly reduce the need for resources and time, which can make systems unfeasible [16]. It does not replace *in vivo* experimentation, but provides important information to achieve scientific and technological goals in critical areas of biology and health [5,13].

Our purpose, in this initial work, is to present a model (basis for the construction of a computational framework) for further simulation in the area of gene expression, more specifically, the mapping between the elements of the transcriptomes and the proteomes (apprehended by *omics*), in order to allow data integration between these two areas, and study their interactions.

Transcriptomic and proteomic are independent areas, however, there are efforts to integrate them [8,15,21]. Our objective is to create a model that integrates both under the same form of representation: **formal languages**. More specifically, we will represent the transcriptomes and the proteomes as families of formal languages, and model the phenomenon of translation as a transduction process of artificial languages.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12253, pp. 727–735, 2020.
https://doi.org/10.1007/978-3-030-58814-4_60

We seek the simplicity necessary to provide an immediate computational implementation, without however making it an *ad hoc* solution. We are inspired by a series of existing approaches that deal with the modeling of genetic processes [1], such as the π-*calculus* and the work of Searls [7,19]. In our model, we will present the biological substrate of the relationship between transcriptomes and the proteomes from the point of view of language processing [9]. Biological information processing is based on the DNA and RNA alphabets of 4 symbols each, and the protein alphabet containing 20 symbols. Biological translation mechanism is represented by the process of transduction between the *"strings of symbols"* [20] formed by such alphabets (RNAs and proteins). Thus, using an analogy with the activity of programming, we can say that a **"source code"** (**transcriptome**) is converted into an **"object code"** (**proteome**), which allows the **"computational architecture"** (**organism**) to execute the instructions contained therein (events that lead to the manifestation of **phenotypic characteristics**).

The work is organized as follows. Section 3 presents the definition both of formal proteome and transcriptome, as well as the main abstractions from biology such as the concept of organism and the establishment of a set of time intervals that serve as a basis for the construction of such definitions. Section 4 presents the different types of transformations that can be applied to the transcriptome. Such transformations take into account only the formal characteristics and, initially, we will not analyze their biological implications. Among these transformations, we highlight the modeling made of the translation phenomenon. Finally, Sect. 5 discusses the results and the next steps to be followed.

2 Background

In this section, we give several definitions and notations required for the adequate discussion of the present article. We assume that the reader has familiarity with basics genetic concepts and computational formal languages theory. The basic computational concepts used in this work (mostly concerning automata theory), as well the pertinent notation, are summarized in Table 1 of Annex; for more details see [11].

2.1 Biological Background

The first genome expression product is the **transcriptome** [6]. It is the complete set of coding transcripts that are active in a given cell organism, tissue, lineage or organelle in a given time interval; constituting the profile (or large-scale pattern) of all mRNAs present in the cell in a given period [12]. The second product of the organism's gene expression is its **proteome**, which is the complete set of all proteins, obtained by the process of **translation** in a given organ, tissue, lineage or cell organelle in a given period of time [4,12].

The Translation process involves ribosomes, mRNAs, tRNAs and amino acids. In the cytoplasm, amino acids must join with their respective tRNAs.

$\mathbb{N}$	**the set of natural numbers**
$card(\mathcal{Y})$	**the cardinality of** $\mathcal{Y}$, (number of elements in the set)
Σ	**an alphabet of symbols**
$\alpha \in \Sigma$	**a symbol of the alphabet**
$\Sigma^* = \bigcup_{k=0}^{\infty} \Sigma^k$	**the reflexive transitive closure over** Σ
$L \subseteq \Sigma^*$	**a language over** Σ
$w \in L$	**a word of** L
$\mathbf{L}$	**a language with repetition words**
$\|w\| = n$ with $n \in \mathbb{N}$	**the lenght of the word** w
$\Sigma^{\leqslant k}$	**the set of all words with lenght less than or equal to** k

Fig. 1. Notation for technical preliminaries related to computational theory.

During translation, the tRNA molecules bind to the ribosome. The ribosome, in turn, processes the entire coding sequence for the mRNA. As this reading takes place, a series of tRNA connections and disconnections are made with the ribosomes, which in concert bring the amino acids that will compose the resulting protein formed.

In the mRNA coding sequence, **each group of three nucleotides corresponds to a given amino acid**. Such a group is called **codon**, which is chemically recognized by the tRNA carrying the corresponding amino acid. Since there are four nucleotides that make up an RNA, there are 64 different codons (Table 1), where three of them do not encode amino acids, but serve as a stop sign for translation. Since there are only 20 amino acids (Table 1 of Annex, first column), **different codons can encode the same amino acid**. Because of this, it is said that the genetic code is **degenerate** [17].

The insertion and connection of amino acids occurs in the same order in which their respective codons are recognized, in a phenomenon known as **principle of collinearity** [14]. Thus, the linear sequence of the nucleotides determines the primary structure in a protein, forming, in parts, the resulting protein sequence, until it reaches the codon that serves as the termination signal.

2.2 Computational Background

Given two alphabets Σ_1 and Σ_2 distinct, the function $R : \Sigma_1^* \to \Sigma_2^*$ is **transduction** [2, p. 65], whose graph $\mathcal{R}$ is described as:

$$\mathcal{R} = \{(u, v) \in \Sigma_1^* \times \Sigma_2^* \quad | \quad R(u) = v\}$$

The function R can be performed by a machine called **transducer** [2, p. 77], defined as:

$$\mathfrak{R} = (\mathcal{Q}, \Sigma_1, \Sigma_2, q_0, \mathcal{F}, \partial)$$

Where $\mathcal{Q}$ is a set of states, Σ_1 is the constituent alphabet of the input strings, Σ_2 is the alphabet of the output strings, q_0 is the initial state, $\mathcal{F} \subseteq Q$ is the set

Table 1. Mapping between codons and amino acids [10]

Amino acid name	Codons
Alanine	GCA GCC GCG GCU
Arginine	AGA AGG CGA CGC CGG CGU
Aspartic acid	GAC GAU
Asparagine	AAC AAU
Cysteine	UGC UGU
Glutamic acid	GAA GAG
Glutamine	CAA CAG
Glycine	GGA GGC GGG GGU
Histidine	CAC CAU
Isoleucine	AUA AUC AUU
Leucine	UUA UUG CUA CUC CUG CUU
Lysine	AAA AAG
Methionine	AUG
Phenylalanine	UUC UUU
Proline	CCA CCC CCG CCU
Serine	AGC AGU UCA UCC UCG UCU
Threonine	ACA ACC ACG ACU
Tryptophan	UGG
Tyrosine	UAC UAU
Valine	GUA GUC GUG GUU
Stop	UAA UAG UGA

of final states and the transition relationship function ∂ is defined as:

$$\partial \subset \mathcal{Q} \times \Sigma_1 \times \Sigma_2 \times \mathcal{Q}$$

For a transducer $\mathfrak{R}$, a **computation** is a sequence of consecutive transitions

$$\Delta = (\delta_1, \delta_2, ..., \delta_n)$$

where:

$$\delta_i = (q', \alpha_i, \beta_i, q'') \text{ and } \delta_{i+1} = (q'', \alpha_{i+1}, \beta_{i+1}, q''')$$

with $\delta_i \in \partial$ and $1 \leq i \leq n$. Computation can also be represented by the notation:

$$\Delta^{(u,v)} = p \xrightarrow{\frac{u}{v}} s$$

where the word $u = \alpha_1\alpha_2 \ldots \alpha_n \in \Sigma_1^*$ is the computing entry and $v = \beta_1\beta_2 \ldots \beta_n \in \Sigma_2^*$ is the output word, for $p, s \in \mathcal{Q}$.

A computation is **successful** if $p = q_0$ and $s \in \mathcal{F}$ [3, p. 18]. Thus, for a computing entry u, the processing $\mathfrak{R}(u)$, calculated by the transducer, is the concatenation produced as a result of the computation. In this way, we have to:

$$\mathfrak{R}(u) = \{v \in \Sigma_2^* \mid \exists(\Delta^{(u,v)} \text{ is successful})\}$$

3 Transcriptome and Proteome as Formal Languages

Although formal (artificial) languages do not encompass the complexity of human language, they are useful not only for the study of linguistic themes, but also for several computational purposes [18]. In general, from a linguistic point of view, an occurrence experienced by someone (real or not) is expressed in a language (natural or artificial) and then transformed into a record, which can be passed on and transformed over time. Something similar is done to define the abstractions for transcriptomes and proteomes: we describe a biological phenomenon in terms of a (formal) language, establishing some assumptions and relaxing some restrictions in relation to the physical phenomenon, building an functional (in the mathematical sense) articulation of possible relationships, which can be "expressed" in the form of computer program and executed under different forms, circumstances and modes (based on experimental or hypothetical data). This abstract construction begins with the description of the prerequisites that serve as support for the model, described in the two items below:

1. (**First prerequisite**) We define the set, called $\mathcal{O}$, as the set of all organisms with transcriptomes and proteomes.
2. (**Second prerequisite**) Both, the RNA and the primary structure of proteins can be described as strings of characters.

Every element of $\mathcal{O}$ is represented by the ordered pair $o = (\mathcal{C}, \mathcal{I})$ where:

- $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ (with $k \in \mathbb{N}$) is the **set of containers**. Each container is a set of a specific type of tissue, cell or cellular organelle, present in the organism $\mathcal{O}$, according to what the analysis at the moment requires. Such an abstraction gives the freedom to treat only the elements relevant to the analysis.
- $\mathcal{I} = \{i_1, i_2, \ldots, i_p\}$ (com $p \in \mathbb{N}$) is called **set of expressiveness intervals**, which consists of a **partition** of an interval Δt of the organism's lifetime.

Given the second prerequisite, we have the definition of the following two alphabets:

- $\Sigma_{\text{RNA}} = \{A, G, C, U\}$ composed of the symbols that represent the four ribonucleotides. A string $s \in \Sigma_{\text{RNA}}^{\leqslant n}$ represents the coding segment of an mRNA.
- $\Sigma_{\text{AM}} = \{A, R, D, N, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$, composed of letters that represent each of the 20 amino acids, according to Table 1 of Annex. A string $p \in \Sigma_{\text{AM}}^{\leqslant m}$, represents the primary structure of a protein.

The natural numbers m and n in $\Sigma_{\mathrm{AM}}^{\leqslant m}$ and $\Sigma_{\mathrm{RNA}}^{\leqslant n}$ are restrictions that prevent the occurrence of proteins or mRNA strings with infinite size.

Thus, the **Formal Transcriptome** (T) and the **Formal Proteome** (Π) can be defined by the functions:

$$T : \mathcal{C} \times \mathcal{I} \rightarrow \mathcal{L}_{\Sigma_{\mathrm{RNA}}} \tag{1}$$

$$\Pi : \mathcal{C} \times \mathcal{I} \rightarrow \mathcal{L}_{\Sigma_{\mathrm{AM}}} \tag{2}$$

Whose codomain $\mathcal{L}_{\Sigma_{\mathrm{RNA}}}$ and $\mathcal{L}_{\Sigma_{\mathrm{AM}}}$ are **families of indexed and non-empty languages**, so that for $\mathbf{max} = card(\mathcal{I}) \times card(\mathcal{C})$, they are specified as:

$$\mathcal{L}_{\Sigma_{\mathrm{RNA}}} = \{\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_j\} \text{ where } (j \leq \mathbf{max}) \text{ and } \forall(\mathbf{L}_i \in \mathcal{L}_{\Sigma_{\mathrm{RNA}}})(\mathbf{L}_i \subset \Sigma_{\mathrm{RNA}}^{\leqslant n})$$

$$\mathcal{L}_{\Sigma_{\mathrm{AM}}} = \{\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_z\} \text{ where } (z \leq \mathbf{max}) \text{ and } \forall(\mathbf{L}_k \in \mathcal{L}_{\Sigma_{\mathrm{AM}}})(\mathbf{L}_k \subset \Sigma_{\mathrm{AM}}^{\leqslant m})$$

where each member language, both in $\mathcal{L}_{\Sigma_{\mathrm{RNA}}}$ and $\mathcal{L}_{\Sigma_{\mathrm{AM}}}$, there may be repeated words.

The modeling presented in this section will allow the definition of a transformation on formal transcriptomes and proteomes.

4 Transformation over Formal Transcriptome

The transformation on formal transcriptomes is related to considerations about what the phenomenon of translation **computes**. First, we have to capture the main characteristics of the translation process in the form of **postulates**. They are:

Postulate 1. *From the Table 1 of Annex, the presence of* ***codons*** *(which we will call k_i) in the coding section of the mRNAs, allows to describe it as:*

$t = k_1 k_2 \ldots k_l$ *with* $k_i \in \Sigma_{RNA}^3$*, where l is the lenght of computation time.*

Postulate 2. *The fact that the genetic code is* ***degenerate*** *makes it possible to describe the equivalence table between codons and amino acids as being a total function D with 64 elements in the domain and 21 elements in the codomain (since stop codons are taken into account):*

$$D : \Sigma_{RNA}^3 \rightarrow \Sigma_{AM}$$

Postulate 3. *The principle of* ***collinearity*** *makes it possible to associate states to the processing steps of the codons by the ribosomes, so that we can represent function D through a set $\mathcal{D}$ of 64 computations:*

$$\mathcal{D} = \{\Delta^{(GCA,A)}, \Delta^{(GCC,A)}, \Delta^{(GCG,A)}, \Delta^{(GCU,A)}, \ldots, \Delta^{(UAA,\varepsilon)}, \Delta^{(UAG,\varepsilon)}, \Delta^{(UGA,\varepsilon)}\} \tag{3}$$

Postulate 4. *Reading the coding portion, done in a single direction, makes it possible to associate an initial state and stop codons (Table 1 of Annex, last line) and also makes it possible to establish a state q_f final. So for $q \in Q$, we have the following specifications for computations involving stop codons:*

$$\begin{aligned} \Delta^{(UAA,\varepsilon)} &= q \xrightarrow[\varepsilon]{UAA} q_f \\ \Delta^{(UAG,\varepsilon)} &= q \xrightarrow[\varepsilon]{UAG} q_f \\ \Delta^{(UGA,\varepsilon)} &= q \xrightarrow[\varepsilon]{UGA} q_f \end{aligned}$$

Now we can state the proposition that is the center of the modeling proposed in this article.

Proposition 1. *For each and every member of the set $\mathcal{O}$ of organisms that have a transcriptome T and a proteome Π formal, there is a transformation M for which:*

$$M(T) = \Pi \tag{4}$$

Proof. Let's build a transducer $\mathfrak{R}_{\text{RNA/AM}}$ to model the translation phenomenon, so that, for T and Π of a particular member body of $\mathcal{O}$, we have the graph:

$$\mathcal{R}_{\text{RNA/AM}} = \{(T(c,i), \Pi(c,i)) \quad | \quad \mathfrak{R}_{\text{RNA/AM}}(T(c,i)) = \Pi(c,i)\}$$

com $T(c,i) \in \Sigma_{\text{RNA}}^{\leqslant n}$ and $\Pi(c,i)) \Sigma_{\text{AM}}^{\leqslant m}$

From the above postulates and using concatenation, union and Kleene's star operations, we built the $\mathfrak{R}_{\text{RNA/AM}}$ transducer shown in Fig. 2, for which:

$$\forall(c \in \mathcal{C}) \forall(i \in \mathcal{I}) \exists \Delta^{(T(c,i),\Pi(c,i))} \textbf{successful}.$$

Figure 2 presents the transducer for some codons. We have also suppressed the transitions and intermediate states related to the recognition of each of the nucleotides that make up the codons. Instead, the elements of the set were used $\mathcal{D}$, presented in the expression 3, through the use of a transition graphic with a differentiated arrow, as can be seen between the states q_0 and q_x, for example.

In possession of the transducer $\mathfrak{R}_{\text{RNA/AM}}$, we can develop a method to build Π from T for any $\mathcal{O}$.

$$\bigcup_{card(\mathcal{C})} \bigcup_{card(\mathcal{I})} \bigcup_{t \in T(c,i)} \mathfrak{R}_{\text{RNA/AM}}(t) = \Pi$$

However, we can use a more compact notation in place of expression

$$\bigcup_{card(\mathcal{C})} \bigcup_{card(\mathcal{I})} \bigcup_{t \in T(c,i)} \mathfrak{R}_{\text{RNA/AM}}(t)$$

replacing it with $M(T)$.

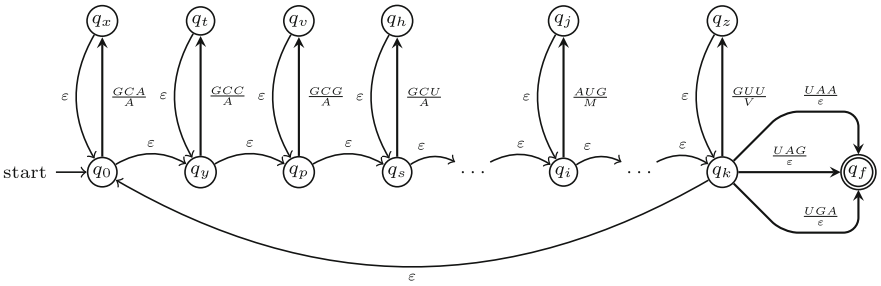

Fig. 2. The transducer $\mathfrak{R}_{\text{RNA/AM}}$

5 Conclusion

When delimiting (by means of formal languages) the aspects of both transcriptomes and proteomes, we stipulate the criteria to show that, although they are heterogeneous in constitution, both are governed by an internal uniformity expressed by the formal structure that describes them, expressed through the developed model.

With absolutely reasonable considerations about the nature of the translation phenomenon, abstractly presented as a language processor, we arrive at the logical conclusion described by the equation $M(T) = \Pi$, which relates transcriptomics and proteomics information.

Thus, based on transcriptomics data, we can infer information about organism proteomics aspects, through *in silico* experiments, going beyond a simple database architecture, entering the field of inference and allowing the automatic generation of a **transcriptome - proteome mapping**, even in situations that have information gaps.

Throughout this preliminary article, we emphasize our approach to focus attention on formal linguistic aspects resulting from computational modeling, placing their biological meanings in the background and verifying the properties resulting from symbolic relations.

Continuing this idea, we can consider and imagine a mapping that is "inverse" to that resulting from the translation phenomenon, that is: morphism $M(\Pi) = T$, which, although not related to any biological phenomenon, could very well be useful in computationally simulated analyses. It would allow the application of **inverse problems** for questions of genetic expressiveness, that is, to answer the question *"given a desired phenotypic characteristic, which transcripts would produce it?"* It is important to note that in such a morphism M', for an input, there are two or more outputs. The implications of this fact and the resulting extensions in the model, as well the influence of other factors (as probability and heuristics, for example), are themes for future work.

References

1. Bartocci, E., Lió, P.: Computational modeling, formal analysis, and tools for systems biology. PLoS Comput. Biol. **12**(1), e1004591 (2016)
2. Berstel, J.: Transductions and Context-Free Languages. Springer, Heidelberg (2013)
3. Berstel, J., Perrin, D., Reutenauer, C.: Codes and Automata, vol. 129. Cambridge University Press, Cambridge (2010)
4. Bessant, C., et al.: Proteome informatics. Royal Society of Chemistry (2016)
5. Butcher, E.C., Berg, E.L., Kunkel, E.J.: Systems biology in drug discovery. Nat. Biotechnol. **22**(10), 1253–1259 (2004)
6. Cellerino, A., Sanguanini, M.: Transcriptome Analysis: Introduction and Examples from the Neurosciences, vol. 17. Springer, Heidelberg (2018). https://doi.org/10.1007/978-88-7642-642-1
7. Datta, S., Mukhopadhyay, S.: A grammar inference approach for predicting kinase specific phosphorylation sites. PLoS One **10**(4), e0122294 (2015)
8. Fan, M., et al.: Integration of deep transcriptome and proteome analyses of salicylic acid regulation high temperature stress in ulva prolifera. Sci. Rep. **7**(1), 1–19 (2017)
9. Gross, F.: The impact of formal reasoning in computational biology. In: Bertolaso, M., Sterpetti, F. (eds.) A Critical Reflection on Automated Science. HPHST, vol. 1, pp. 139–155. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-25001-0_7
10. Hartwell, L., Goldberg, M.L., Fischer, J.A., Hood, L.E., Aquadro, C.F.: Genetics: from Genes to Genomes. McGraw-Hill, New York (2008)
11. Hopcroft, J.E.: Introduction to Automata Theory, Languages, and Computation. Pearson Addison Wesley, Boston (2007)
12. Kahl, G.: The Dictionary of Genomics, Transcriptomics and Proteomics. Wiley, Hoboken (2015)
13. Kitano, H.: Systems biology: a brief overview. Science **295**(5560), 1662–1664 (2002)
14. Klug, W.S., Cummings, M.R., Spencer, C.A., Palladino, M.A.: Concepts of Genetics. Benjamin Cummings, San Francisco (2014)
15. Kumar, D., Bansal, G., Narang, A., Basak, T., Abbas, T., Dash, D.: Integrating transcriptome and proteome profiling: strategies and applications. Proteomics **16**(19), 2533–2544 (2016)
16. Nutaro, J.J.: Building Software for Simulation: Theory and Algorithms, with Application in C++. Wiley Online Library, Hoboken (2011)
17. Pierce, B.A.: Genetics: A Conceptual Approach. Macmillan, New York (2012)
18. Rozenberg, G., Salomaa, A.: Handbook of Formal Languages: Volume 3 Beyond Words. Springer, Heidelberg (2012)
19. Searls, D.B.: The language of genes. Nature **420**(6912), 211–217 (2002)
20. Sempere, J.M.: On compensation loops in genomic duplications. Int. J. Found. Comput. Sci. **31**(01), 133–142 (2020)
21. Zhu, W., et al.: Integration of transcriptomics, proteomics and metabolomics data to reveal the biological mechanisms of abrin injury in human lung epithelial cells. Toxicol. Lett. **312**, 1–10 (2019)