BMC Bioinformatics

RESEARCH

Open Access

Ribosomal computing: implementation of the computational method

Pratima Chatterjee[1], Prasun Ghosal[2], Sahadeb Shit[1], Arindam Biswas[1], Saurav Mallik[3,4], Sarah Allabun[5], Manal Othman[5], Almubarak Hassan Ali[6], E. Elshiekh[7] and Ben Othman Soufiene[8*]

*Correspondence:
soufiene.benothman@isim.rnu.tn

[1] Kazi Nazrul University, Asansol, West Bengal, India
[2] IIEST Shibpur, Howrah, West Bengal, India
[3] Department of Environmental Health, Harvard T H Chan School of Public Health, Boston, USA
[4] Department of Pharmacology & Toxicology, University of Arizona, Tucson, AZ 02115, USA
[5] Department of Medical Education, College of Medicine, Princess Nourah bint Abdulrahman University, P.O.Box 84428, Riyadh 11671, Saudi Arabia
[6] Division of Radiology, Department of Medicine, College of Medicine and surgery, King Khalid University (KKU), Abha, Aseer, Kingdom of Saudi Arabia
[7] Department of Radiological Sciences, College of Applied Medical Sciences, King Khalid University, Abha, Saudi Arabia
[8] PRINCE Laboratory Research, ISITcom, Hammam Sousse, University of Sousse, Sousse, Tunisia

Abstract

Background: Several computational and mathematical models of protein synthesis have been explored to accomplish the quantitative analysis of protein synthesis components and polysome structure. The effect of gene sequence (coding and non-coding region) in protein synthesis, mutation in gene sequence, and functional model of ribosome needs to be explored to investigate the relationship among protein synthesis components further. Ribosomal computing is implemented by imitating the functional property of protein synthesis.

Result: In the proposed work, a general framework of ribosomal computing is demonstrated by developing a computational model to present the relationship between biological details of protein synthesis and computing principles. Here, mathematical abstractions are chosen carefully without probing into intricate chemical details of the micro-operations of protein synthesis for ease of understanding. This model demonstrates the cause and effect of ribosome stalling during protein synthesis and the relationship between functional protein and gene sequence. Moreover, it also reveals the computing nature of ribosome molecules and other protein synthesis components. The effect of gene mutation on protein synthesis is also explored in this model.

Conclusion: The computational model for ribosomal computing is implemented in this work. The proposed model demonstrates the relationship among gene sequences and protein synthesis components. This model also helps to implement a simulation environment (a simulator) for generating protein chains from gene sequences and can spot the problem during protein synthesis. Thus, this simulator can identify a disease that can happen due to a protein synthesis problem and suggest precautions for it.

Keywords: Mathematical model, Computational model, Ribosomal computing, Protein synthesis, Ribosome, Stalling, Gene sequence, Mutation, Protein, Simulator

Introduction

Nowadays, the number of diseases is increasing due to the added number of mutations in bacterial, viral, and other micro-organisms [1]. They target to alter the structural and behavioral characteristics of the host body [2]. Any living body's structural

and behavioral characteristics are controlled by gene sequence and functional protein [3]. DNA and RNA contain the gene sequence and transfer the information from the previous to the next generation, but they are often not reactive [4]. On the other hand, protein is an active element and participates in different reactions to specify the characteristics of an organism. The protein synthesis process in a cell establishes the relationship between gene sequence and functional protein. This process translates the gene sequence into a functional protein[5] with the help of several components, *viz* different types of RNAs, proteins, macro-molecules, small molecules, and protein translocation. However, the exciting features and the complex structure of ribosome molecules and protein synthesis have drawn scientists' attention to exploring them.

The crystallization techniques are performed to determine the physical structure of the ribosome and various protein synthesis components [10] to reveal the complex micro-operations of protein synthesis. However, this technique is complex to understand and implement, time-consuming, costly, and can only handled by expert persons. Thus, besides this technique, different mathematical models are developed to characterize the properties of the ribosome and the protein synthesis process [11, 12] for easy understanding of the process. These models have presented the relationship between input and output units through some quantitative parameters, *e.g.* codon of mRNA, ribosome movement, hopping probability of ribosome, of the protein synthesis process. Among these mathematical models, the *ODE-based* [6] model is a straightforward approach. This model is developed based on multiple elongation processes on a single mRNA and gives a clear picture of the status of the accessible codons' position on an mRNA. The immense equation size of this model to represent a complex system restricts the scope of this model. The *statistical* [7] model focuses on analyzing only the kinetics of the protein synthesis. The *TASEP-based* [8] model is developed to predict the hopping probability of ribosome molecules during protein synthesis. This model is beneficial to understand the environment of the process. The *PBN-based approach* [9] is fruitful in analyzing the elongation process using the rule of state distribution. At a glance, the scopes and limitations of these models are available in Table 1. Though various quantitative predictions over protein synthesis have been studied, different protein synthesis components' functional dependencies and effects on protein functionality have not been widely explored. In the proposed

Table 1 Mathematical models of protein synthesis

Model	Scope of the model	Limitation of the model
ODE based Model [6]	It describes polysome structure accurately by introducing different species for every possible combination of ribosome occupancy by an mRNA	Type of equations are complex and lengthy to represent a simple operation
Statistical Model [7]	Clears the concept of codon-based mRNA translation and rate of translation	It focuses only on the kinetics of protein synthesis process
TASEP based	It focuses on ribosome's hopping probability and to study rate limiting	It is restricted and later focuses on codon
Model [8]	factors and traffics of protein synthesis	specific hopping probability
PBN based Approach [9]	It uses a boolean network tool to analysis the steady state distribution of mRNA during protein synthesis	It analyses the quantitative data of protein synthesis

work, we wish to develop a new model that will find out the mathematical functions between gene sequence and protein to make a clear concept of the process that will help to analyze protein motifs, predict drugs, and predict the structure and function of the protein.

Despite this, nonconventional computing has become a promising alternative due to the scaling limit of conventional computing. Most notable nonconventional computing techniques are *Quantum Computing* [13], *DNA Computing* [14, 15], *Ribosomal*, *Peptide Computing* [16], *Optical Computing* [17]. Among these, DNA Computing, Ribosomal Computing, and Peptide Computing are examples of biocomputing, and they are energy efficient and time efficient due to the inherent parallel nature of the micro-operations used to implement a computational algorithm. However, most of them are limited by manual intervention. Ribosomal computing can overcome this limitation by imitating the protein synthesis process for computation. In this computing, the mRNA structure with a stalling motif represents the input, and the amino acids sequence in the protein chain represents the output. The set of mutations acts for conditional control of this computing. In some recent works, logic operations [18], sequential operations [19], and a few arithmetic operations (comparator, shifters, and many more [20]) have been implemented successfully using this computing technique. Depending upon close observation, we could reveal some advantages of this computing. They are the microscopic size of the working unit (mRNA-Ribosome), energy awareness, and inherent parallelism to minimize time. Even though several manually intervened subtasks help to implement the input structure, the input-to-output mapping technology follows automation as it follows the automated process. Thus, this computing includes partial automation. This computing is in the initial stage. However, we wish that it will be changed into full automation by the invention of an automatic input generation unit in the near future. These advantages motivate us to further propel into this computing and find out its applications compared to other existing techniques presently in use for computation. Even though several computational works have been implemented using this computing, a general model does not exist for it. In the present work, we are going to develop this generalized model.

Contribution of the proposed work

A mathematical model describes a complex system in the wrapper of abstraction. The previous mathematical models of protein synthesis explain or predict various critical functions of the process. However, a significant section of the process remains unexplored, such as motif structure, protein folding, the relationship between protein chain and exit gate, the effect of mutation over protein synthesis, the impact of upstream mRNA sequence over protein chain elongation. Conversely, a mathematical model of a computational technique provides a common platform to bind different logic operations. In ribosomal computing, several computational tasks are implemented with varying structures of input and control conditions. Thus, a mathematical model is also required to set a common platform for this computing. The Fig. 1 shows the generic view for implementing the new model protein synthesis. Thus, the enlisted contributions of the present work are as follows.

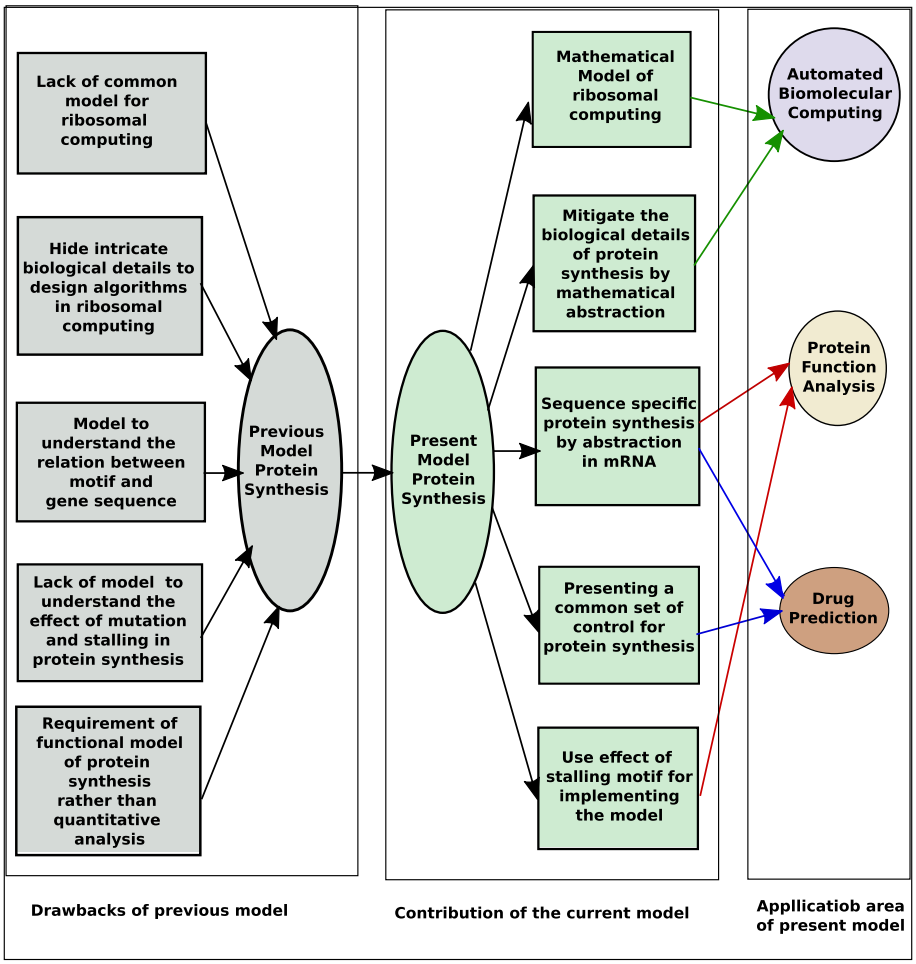


Fig. 1 Goal of the new model of protein synthesis

- Several models of protein synthesis with quantitative parameters are proposed. Still, the functional dependency between different components of protein synthesis is unclear due to the lack of a functional model. The proposed model demonstrates the functional dependency among different components of protein synthesis alternative to quantitative analysis. Therefore, complex biochemical suboperations of protein synthesis can be understood easily and straightforwardly rather than complicated and gigantic equations.
- Nowadays, protein functionality is specified by a motif (a sub-sequence of the protein chain). A protein can contain several motifs in its structure. Motifs are essential for disease prediction. Thus, a model is required to understand the generation of motifs from a gene sequence. In the proposed work, the sequence-specific mRNA structure is considered rather than the codon-specific structure, which helps to understand the relationship between gene sequence and protein function.
- Till now, any model is not present to describe the effect of stalling. The stalling motif is used in the model to perform different tasks. Thus, one can understand the impact of ribosome stalling on protein synthesis.

- Several protein synthesis components and mutation can control protein formation. This fact is not encountered as a single component in any previous model. In the proposed work, the word *'mutation'* is used as a common term for such controls. Thus, the complexity of the mathematical model of protein synthesis is mitigated in the present work.
- Implementing different computational tasks in ribosomal computing requires different input structures and control factors. Therefore, a standard model is desired to unite all of them. The present work provides a mathematical model as a common platform for maintaining unity among different computational tasks of ribosomal computing.
- The designing of algorithms on the biomolecular computing system (ribosomal computing) demands understanding the underlying principles of a biological procedure. So, a mathematical model where the biological information is presented in mathematical notation is needed for this computing system to assuage the complexity. The proposed model provides a mathematical abstraction of biological details. This model benefits computer scientists or developers as they can design various problem-solving algorithms without delving into the intricate biological information of the underlying process.

In the present section, we have explained the necessity of the mathematical model of ribosomal computing. The rest of the paper is organized to demonstrate the proposed model's development and application area. At first, a brief description of the components and the underlying biological process of the ribosomal computing system are described in Sect. "Biological details of ribosomal computing system" for maintaining the flow. Then, the proposed mathematical model is elaborated in Sect. "Proposed mathematical model of ribosomal computing". In this section, the chosen variables are described first to develop the model. Then, the relationships among the variables are defined using some newly developed functions. Section "Complexity of the mathematical model" derives the complexity of the proposed model to verify the accuracy of the model with natural protein synthesis. In Sect. "Applicability of the proposed model", the application area of the proposed work is described. Finally, in Sect. "Conclusion" the conclusion of the work is presented with the limitations of the work.

Biological details of ribosomal computing system

A brief overview of ribosomal computing is required to describe the proposed work further, as the computing imitates the protein synthesis process. Thus, this section demonstrates the protein synthesis operations and their components. First, the elements of protein synthesis are explained. Then, the description of the protein synthesis process is presented.

Details of components

The backbone of ribosomal computing imitates the protein synthesis process. In a cell, the genetic information is translated into a protein chain during this process. This process occurs in ribosome molecules referred to as *Molecular Machine* [21] for its behavior. Three types of RNA play a significant role in this process, though several RNAs

participate in it. The mRNA(messenger RNA) brings the input information from the DNA through the *triplet codon*, and the tRNA (transfer RNA) brings the *amino acid* as the output unit for protein synthesis [22]. The mutation in rRNA (ribosomal RNA) and rProtein (ribosomal protein) also controls the protein synthesis [23]. In Fig. 2, we present three significant components of protein synthesis, *viz* : ribosome, mRNA, and amino-acyle-tRNA-GTP complex. A brief description of all of them is as follows.

1. **Ribosome** is a complex biomolecular assembly. It has two subunits [24]. The upper subunit is larger than the lower one. Each subunit contains rRNA and rproteins [25]. The lower subunit has one mRNA binding site. The upper subunit has three tRNA binding sites. They are A site, P site, and E site. The **rRNA** regulates protein synthesis procedure [26]. Different kinds of mutations in rRNA affect protein synthesis [27, 28]. According to the definition, a mutation is a change in nucleotide bases that can change the behavior of a nucleotide sequence or gene sequence. A mutation that can suppress a gene sequence's behavior is referred to as a suppressor mutation [29].
2. The **tRNA** brings amino acid and is referred to as aminoacyl tRNA [30, 31]. During protein synthesis after reading the mRNA codon, the tRNA is bound with the mRNA codon with its matched anti-codon by hydrogen bond [32, 33]. The **GTP** molecule supplies the energy for the protein synthesis process.

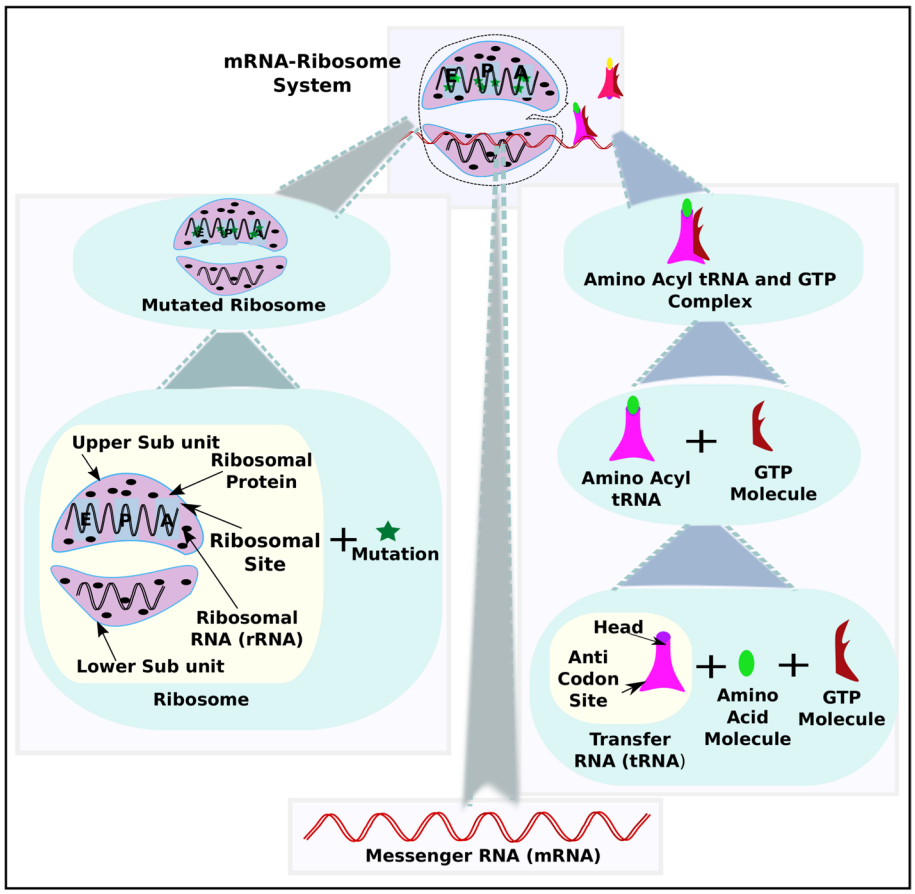


Fig. 2 The complete design of an mRNA-ribosome system

3. The inputs are given to the process through **mRNA** [34, 35]. For ribosomal computing, mRNA generation algorithms have been developed following the DNA sticker model till now. The mRNA can be produced artificially using other methods[36]. However, in the proposed work, we have focused on the working principle of protein synthesis.

Underlying process

At first, the Shine-Dalgarno(SD) sequence of the lower ribosomal subunit binds to the anti-SD sequence of mRNA. Then, the entire unit is added to the upper subunit for completion before the initiation of protein synthesis. However, the proposed model considers an artificial ribosome [37] for protein synthesis. The protein synthesis process occurs in the following sequence of steps. We have instantiated a step of elongation of the protein synthesis process in Fig. 3.

1. Protein synthesis starts by reading the start codon AUG from mRNA at the A site. The Methionine amino acid-carrying tRNA binds with the start codon via hydrogen bonds. This pair enters the ribosome through the A site and directly comes to the *P site*.
2. It reads the triplet codon onwards, following a sequence of steps, until it reads the stop codon UAG in the *A site*.

    (a) Like the start codon, *A site* reads codon from mRNA and binds with appropriate aminoacyl tRNA with the help of anti-codon. Each aminoacyl tRNA brings an elongation factor (Ef-Tu for a prokaryotic cell, EF-G for a eukaryotic cell). The pair (consisting of aminoacyl tRNA and the read codon of mRNA) enters the ribosome through the *A site*. At this instant, the protein chain at the head of the previous tRNA at the *P site* gets held by PTC(Peptidyl Transferase Center).

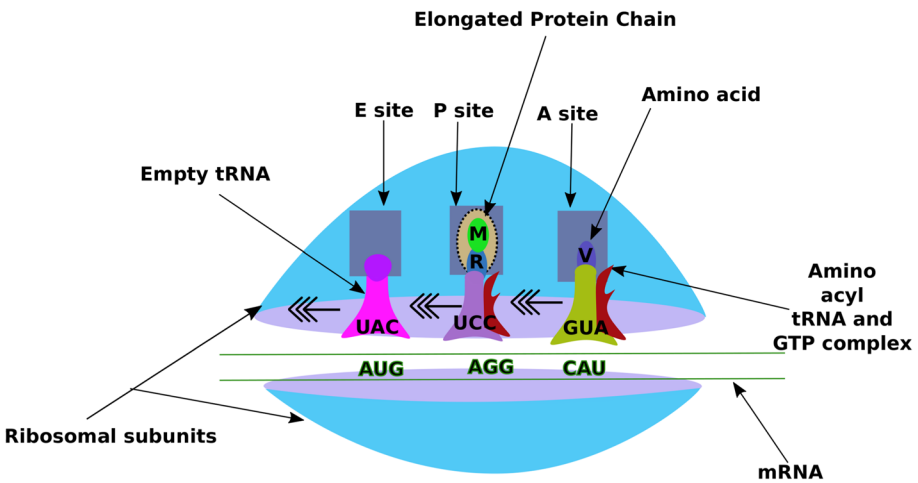


Fig. 3 An instant of elongation of protein synthesis in an mRNA-ribosome system

(b) PTC releases the hold protein chain and gets attached to the amino acid at the head of the tRNA of the *A site*, and another amino acid increases the protein chain. At this instance, GTP hydrolysis occurs, which deforms the overall structure, thus helping the tRNA come to the *P site*. The previous tRNA (now empty) of the *P site* goes to the *E site* and exits from the system.

3. After reading the stop codon (UAG), the *A site* does not choose any further aminoacyl tRNA and stops the protein synthesis procedure. The protein chain is broken from the head of the tRNA and departs from the ribosome molecule.

Proposed mathematical model of ribosomal computing

This section presents the internal working principle and the properties or relationships among different components of ribosomal computing with mathematical notation. The protein synthesis components are considered as variables. On the other hand, the biochemical operations (microoperation) are represented by mathematical functions, which represent the roles of these variables in protein synthesis. The dependency diagram describes each function in this section and illustrates the relationships among microoperations. However, the model is divided into three parts for ease of discussion.

1. In Sect. "Mathematical formulation" .each component of the mRNA-Ribosome system and various functions performed by these components have been identified and represented by suitable mathematical representations.
2. In Sect. "Illustrative evaluation of the model", we verified the proposed model using an example of an XOR operation. First, the structure of different variables for XOR is shown in this section. Then, we demonstrated the mapping between the input and output parameters of the XOR operation.
3. In Sect. "Complexity of the mathematical model", we have validated the model by comparing it with the natural process. This section shows that the model is simple enough to implement as we avoid the chemical reaction in detail. The proposed model follows a particular operation of protein synthesis.

Mathematical formulation

The biological macro-molecules and the biochemical elements of a biological system are considered variables that can be used to develop a mathematical model of that biological system. The equations defining the variables' roles and relationships are called functions. Similarly, we have considered the components of protein synthesis as variables and the atomic biochemical operations as functions to develop the model of ribosomal computing as it emulates the behavior of protein synthesis. A new level of abstraction is constructed for the development of the model, as the goal of this model is to demonstrate a generalized framework for ribosomal computing and to establish a relationship between gene sequence and protein functionality. In the proposed work, the variables are defined before constructing functions. In practical scenarios, the entire protein synthesis process is divided into three suboperations: initiation of protein synthesis, elongation of the protein chain, and protein synthesis termination. Conversely, similar suboperations

occur for the translation of each triplet codon. However, the proposed model does not emphasize the suboperations of the entire process but the suboperations for translating each triplet codon. We have observed from the literature survey that each codon translation can be divided into four suboperations. Thus, we consider four suboperations, *viz* : codon reading, condition checking, decision making, and tRNA binding and elongation for the proposed model.

Variables

Variables are symbolic names associated with a value that someone can change and use appropriately in a mathematical expression. The model of ribosomal computing is constructed with three variables. Among them, two are input variables, and one is the output variable. One input variable is the information variable (mRNA), which carries the genetic information needed for protein synthesis. The other input variable (set of mutations) controls protein synthesis. The output variable is the protein chain. The detailed descriptions of the three variables above, mRNA, mutation, and protein chain, for the proposed mathematical model are elaborated as follows.

- *mRNA* In the practical model of protein synthesis, the mRNA is a sequence of four types of nucleotide bases (A, U, G, and C) with ribose sugar. In an mRNA reading frame, three consecutive nucleotide bases (triplet codons) bind an mRNA codon with an amino acyl-tRNA containing the correct anti-codon. Therefore, it is clear that a set of triplet codons constructs a set of amino acid sequences during protein synthesis. We have observed that some small sets of amino acids create protein motifs defining protein functionality. Thus, let us consider the structure of mRNA as a four-layer abstraction rather than a single chain of nucleotide bases. The abstrac-

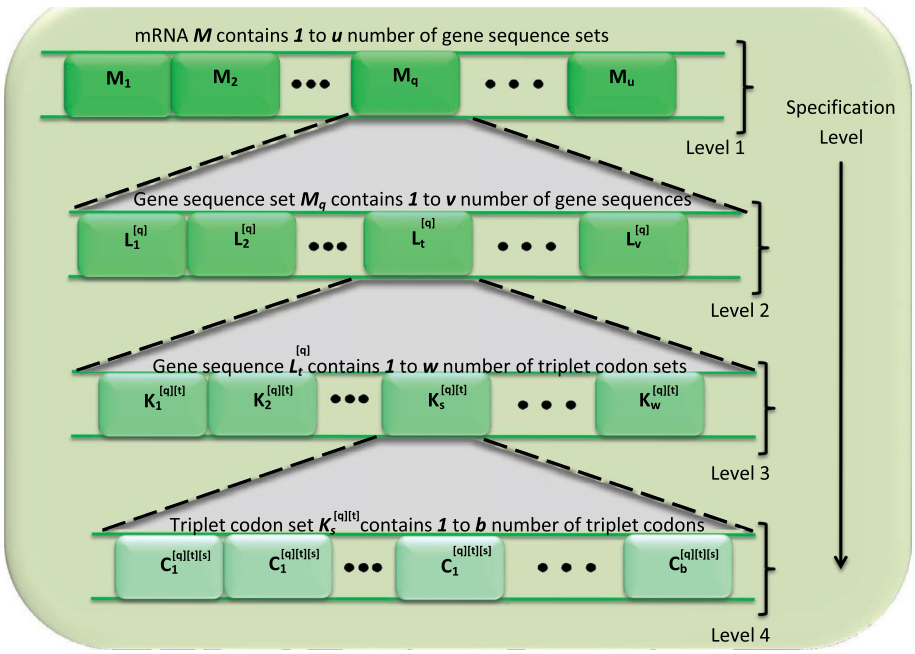


Fig. 4 The abstract structure of mRNA for the proposed model

tions are presented in the top-to-bottom level view in Fig. 4. In the first (top) level of abstraction, let us assume that the mRNA consists of some gene sequence sets. In the second level, we suppose each gene sequence set is constructed with some gene sequences. In the third level of assumption, let us further presume that each gene sequence consists of a triplet codon set. In the last level, let us assume that each triplet codon set consists of some triplet codon, and each triplet codon is a character.

Let us assume that the mRNA M contains u number of gene sequences set. Mathematically, M is an array of strings. So, the representation of mRNA M is depicted in Eq. 1.

$$M = \{M_1, M_2, M_3, \cdots, M_u\} \tag{1}$$

Then, we consider that $M[q]$ is the q^{th} gene sequence set for mRNA M where $1 \leq q \leq u$ and $M[q]$ is further composed into v number of the gene sequence. Therefore, we demonstrate $M[q]$ gene sequence set in Eq. 2. In Eq. 2 $L_{\#}^{[q]}$ represents each gene sequence.

$$M[q] = \{L_1^{[q]} L_2^{[q]} \ldots L_v^{[q]}\} \tag{2}$$

Again, we consider that $L[t]$ is t^{th} gene sequence of $M[q]$ gene sequence set where $1 \leq t \leq v$ and it contains w number of triplet codon set. Now, we have given the representation of $L[t]$ sub-string with w number of triplet codon set in Eq. 3 where $K_{\#}^{[q][t]}$ represents a triplet codon set.

$$M[q][t] = L[t] = \{K_1^{[q][t]} K_2^{[q][t]} \cdots K_w^{[q][t]}\} \tag{3}$$

Let us consider that $K[s]$ is the s^{th} triplet codon set of $L[t]$ gene sequence sequence of $M[q]$, where $1 \leq s \leq w$. The $K[s]$ triplet codon set consists of b number of codons. We demonstrate $K[s]$ triplet codon set in Eq. 4, where $C_{\#}^{[q][t][s]}$ represents a triplet codon.

$$M[q][t][s] = L[t][s] = K[s] = \{C_1^{[q][t][s]} C_2^{[q][t][s]} ... C_b^{[q][t][s]}\} \tag{4}$$

So, a triplet codon of an mRNA is represented by a four-dimensional variable. One point to note is that the length of each triplet codon is three nucleotide bases. Each mRNA is composed of many triplet codons. So, the total length of an mRNA in terms of the number of nucleotides is $3 \times f(u, v, w, b)$, where the function $f(u, v, w, b)$ gives the count (position in mRNA) of triplet codon. However, the mRNA for this model contains five types of regions, as an mRNA consists of five types of fields in ribosomal computing. Figure 5 illustrates the structural division of an mRNA for ribosomal computing. u is $n + 4$ for performing the n bit operation when all the

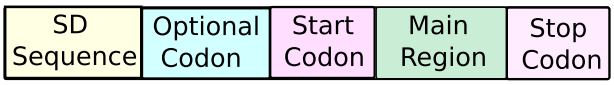


Fig. 5 General structure of input template (mRNA of ribosomal computing)

fields are present. Now, we show that the abstraction, as mentioned above, reflects the structure of an mRNA of ribosomal computing.

1. *SD sequence* In Eq. 1, M_1 represents the Shine Dalgarno sequence (SD sequence). This field is obvious for each operation. We observe that researchers have found the *UAAGGAGGU* sequence in *E. coli* to be an SD sequence. *E. coli* is suitable for the practical experiment for its simplest structure [38]. We consider this sequence for the proposed model to be an SD sequence. M_1 consists of 9 nucleotide bases (three triplet codons) for the prokaryotic ribosome[38]. Therefore, the value of u,v, and w is 1, and the value of b is 3 for this field. So, M_1 can be represented in Eq. 5.

$$M_1 = C_1^{[1][1][1]} C_2^{[1][1][1]} C_3^{[1][1][1]} = \underbrace{UAAGGAGGU}. \tag{5}$$

2. *Optional codon* This user-defined field is reserved for future use. This field is optional for an operation. The M_2 symbol in Eq. 1 represents the optional codon. We assume that there is one gene sequence and one triplet codon set. So, the value of u is 2, and the value of v and w is 1. If p number of codons is present in this field, the value of b is p. This field is represented as in Eq. 6.

$$M_2 = C_1^{[2][1][1]} C_2^{[2][1][1]} \dots C_p^{[2][1][1]}. \tag{6}$$

3. *Start codon* M_3 substring in Eq. 1 represents the start codon. It is an obvious field and is present in all operations. The start codon M_3 trivially consists of only a single triplet codon *AUG*, as in Eq. 7. As a result the value of u is 3 and value of v, w and b is 1.

$$M_3 = C_1^{[3][1][1]} = \underbrace{AUG} \tag{7}$$

4. *Main region* The main region combines input and output codon sequence. n number of such combinations are required for n bit operation. The combinations are shown in the under brace section as $M = M_1 M_2 M_3 \underbrace{M_{3+1} M_{3+2} \cdots M_{3+n}} M_{3+n+1}$. Now, we represent each n number combination in Eq. 1. The input codon sequence signifies the number of codons representing the input data; the output codon sequence signifies the number of output codons. In each combination, a special sequence may be present, or they may combine with the input sequence in addition to the input and output sequence. So, we can say that v is at most 3 for each input–output combination, as three types of codon sequences are present in each combination. Now, we consider that the input codon sequence consisted of w_1 number of triplet codon sets, the output gene sequence consisted of w_2 number of triplet codon sets, and the special gene sequence consists of w_3 number of triplet codon sets. The values of w_1, w_2, and w_3 may not equal. For example, we consider an input–output combination in Eq. 8. In the combination of Eq. 8, the input codon sequence consisted of three gene sequence types. The input gene sequence consists of three types of triplet codon sets, which are *SecM*, *IP*, and *SecA*. The output gene sequence con-

sists of two triplet codon sets, which are OP_1 and OP_2. The special gene sequence consists of SP_1 and SP_2. Therefore, the value of w_1, w_2 and w_3 are 3, 2 and 2 for the combination of Eq. 8.

$$\underbrace{\underbrace{SecM}\ \underbrace{IP}\ \underbrace{SecA}|}_{\text{Input gene sequence}}\ \underbrace{\underbrace{OP_1}\ \underbrace{OP_2}}_{\text{Output gene sequence}}\ |\ \underbrace{\underbrace{SP_1}\ \underbrace{SP_1}}_{\text{Special gene sequence}} \tag{8}$$

Now, we consider that the entire input sequence (w_1 number of triplet codon set) consists of x number of triplet codon in such a way that ($x = x_1 + x_2 + \cdots + x_{w_1}$), where x_1 is a number of triplet codons for first triplet codon set. Similarly, we assume the entire w_2 number of triplet codon set of output sequence consisted of y number of triplet codons in such a way that ($y = y_1 + y_2 + \cdots + y_{w_2}$), where y_1 is a number of triplet codons for first triplet codon set of output sequence. Similarly, we assume the entire w_3 number of triplet codons set in a special sequence consisting of z number of triplet codons in such a way that ($z = z_1 + z_2 + \cdots + z_{w_3}$), where z_1 is a number of triplet codons for first triplet codon set of special sequence. So, the total number of triplet codons in each input–output combination is $(x + y + z)$. Therefore, in Eq. 9 we demonstrate the input–output combination for M_{ip}th gene sequence set of mRNA M where $(3 + 1) \leq M_{ip} \leq (3 + n)$.

$$\begin{aligned} M_{ip} = & \underbrace{C_1^{[i][1][1]} \cdots C_{x_1}^{[i][1][1]}\ C_1^{[i][1][2]} \cdots C_{x_2}^{[i][1][2]}\ C_1^{[i][1][w_1]} \cdots C_{x_{w_1}}^{[i][1][w_1]}}_{} \\ & \underbrace{C_1^{[i][2][1]} \cdots C_{y_1}^{[i][2][1]} \cdots\ C_1^{[i][2][w_2]} \cdots C_{y_{w_2}}^{[i][2][w_2]}}_{} \\ & \underbrace{C_1^{[i][3][1]} \cdots C_{z_1}^{[i][3][1]} \cdots\ C_1^{[i][3][w_3]} \cdots C_{z_{w_3}}^{[i][3][w_3]}}_{} \end{aligned} \tag{9}$$

5. *Stop codon* In Eq. 1, the last sub-string represents stop codon. The value of u is $(3 + n + 1)$ The value of v, w and b is 1. The Eq. 10 represents the stop codon for this model.

$$M_{(3+n)+1} = C_1^{[(3+n)+1][1][1]} = \underbrace{UAA} \tag{10}$$

- *Mutation* In ribosomal computing, mutations are used as a control variable for the biochemical process. They represent the conditions for the computing technique. In the proposed model, we consider each mutation as an object; thus, the set of mutations is an array of objects. Each mutation object is defined by three properties: the activator, the deactivator, and the function. More than one activator or deactivator may present for a mutation, and it may perform more than one function. We assume that some gene sequences in mRNA act as activators or de-activators of the corresponding mutation, though the reality is experimentation-specific. Figure 6 depicts the abstraction layer of mutations for the proposed model. In Eq. 11, the mutation array is represented symbolically as CM, where CM_i represents i^{th} mutation. In this mutation array, r number of mutations is present.

$$CM = \{CM_1, CM_2, \ldots CM_r\} \tag{11}$$

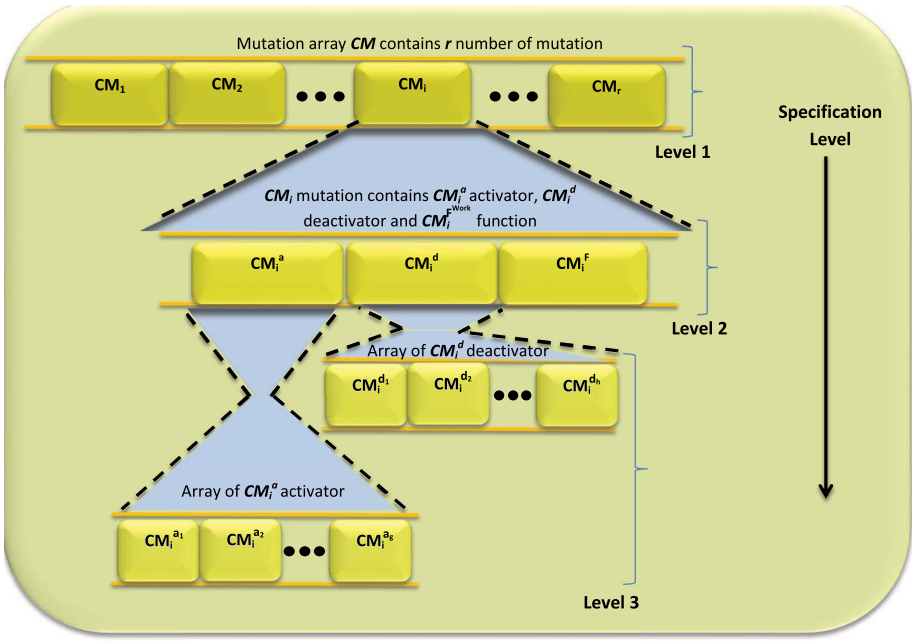


Fig. 6 Structure of mutation array for the mathematical model of mRNA-Ribosome system

In Eq. 12, we show that each i^{th} of mutation array CM consists of these properties. CM_a^i represents the activation condition, CM_d^i represents the deactivation codon and CM_i^F represents the functions of the mutation.

$$CM_i = \{CM_i^a, CM_i^d, CM_i^F\} \tag{12}$$

In Eq. 13, the array of CM_i^a is represented. This array consists of a set of substrings, each representing an individual activator of the mutation CM_i. This array contains g number of elements. So, the array element $CM_i^{a_j}$ signifies the j^{th} activator of i^{th} mutation, where $(1 \leq j \leq g)$.

$$CM_i^a = \{CM_i^{a_1}, CM_i^{a_2}, \cdots, CM_i^{a_g}\} \tag{13}$$

Similarly, CM_i^d is an array of substrings representing a set of deactivators for this mutation in Eq. 14. Let us consider h number of deactivators present for i^{th} mutation. Since $CM_i^{d_j}$ represent j^{th} deactivation sequence for i^{th} mutation, where $(1 \leq j \leq h)$.

$$CM_i^d = \{CM_i^{d_1}, CM_i^{d_2}, \cdots, CM_i^{d_h}\} \tag{14}$$

CM_i^F contains a set of rules for mutation CM_i. The rules are either (a) selection of an output codon, (b) skipping of an output codon, or (c) activation of another mutation according to the property of the mutation. In function, F_{Work}, the elongation is ensured according to the enlisted properties of mutation CM_i in CM_i^F and the detected mRNA codon (D).

- *Protein chain* The protein chain is an array of amino acids. Equation 15 shows the array of amino acids. We consider that the chain PC contains n amino acids. Figure 7 depicts the representation of the protein chain.

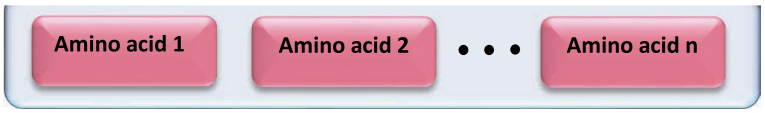


Fig. 7 Structure of protein chain for the mathematical model of mRNA-Ribosome system

$$PC = \{aa_1\ aa_2\ aa_3\ \ldots aa_n\} \tag{15}$$

Functions

We have already discussed the abstract representation of variables for the proposed computational model of ribosomal computing. Now, we are going to elaborate on the intricate relationship between the input and output variables of protein synthesis by analyzing the functionality of this model. We have observed that each mRNA codon is translated into an amino acid following four types of suboperations. They are (a) read (means detection of an mRNA codon), (b) condition checking (related to selection of mutation), (c) decision making (that means selection of an output codon), and (d) binding of tRNA and elongation (signify attachment of amino acid to the protein chain). In the proposed model, we assume four functions for allocating each of these individual suboperation to each of them. These four functions are named as F_{Im}, F_{Muta}, F_{Work} and F_{Bind}. Figure 8 shows each function's reflection in the practical protein synthesis model. The allocation of each function for each suboperation is as follows.

1. Read (Detection of mRNA codon) operation is assigned to function F_{Im}.
2. Checking the condition for the process (mutation selection) is allocated to function F_{Muta}.
3. The decision-making operation (selection of output codon) is appointed to function F_{Work}.
4. Binding of tRNA and elongation (amino acid attachment to the protein chain) is allotted to function F_{Bind}.

The interacting relationships of all these functions are demonstrated in Fig. 9. In a practical scenario, each suboperation for codon translation consists of several biochemical reactions. In the proposed model, these biochemical reactions are referred to as microoperations. Thus, we allocate one or more microoperations to accomplish the task of each function and establish the relationship with the practical model. This way, we perform the event validation technique to validate the proposed model. Therefore, the implementation and validation details of each function of the model are as follows.

- F_{Im}: This function detects the mRNA sequences and introduces the way of the elongation process. The Eq. 16 presents the relationship among the variables for the function F_{Im}. In Eq. 1, the M is the mRNA, and D is the input codon.

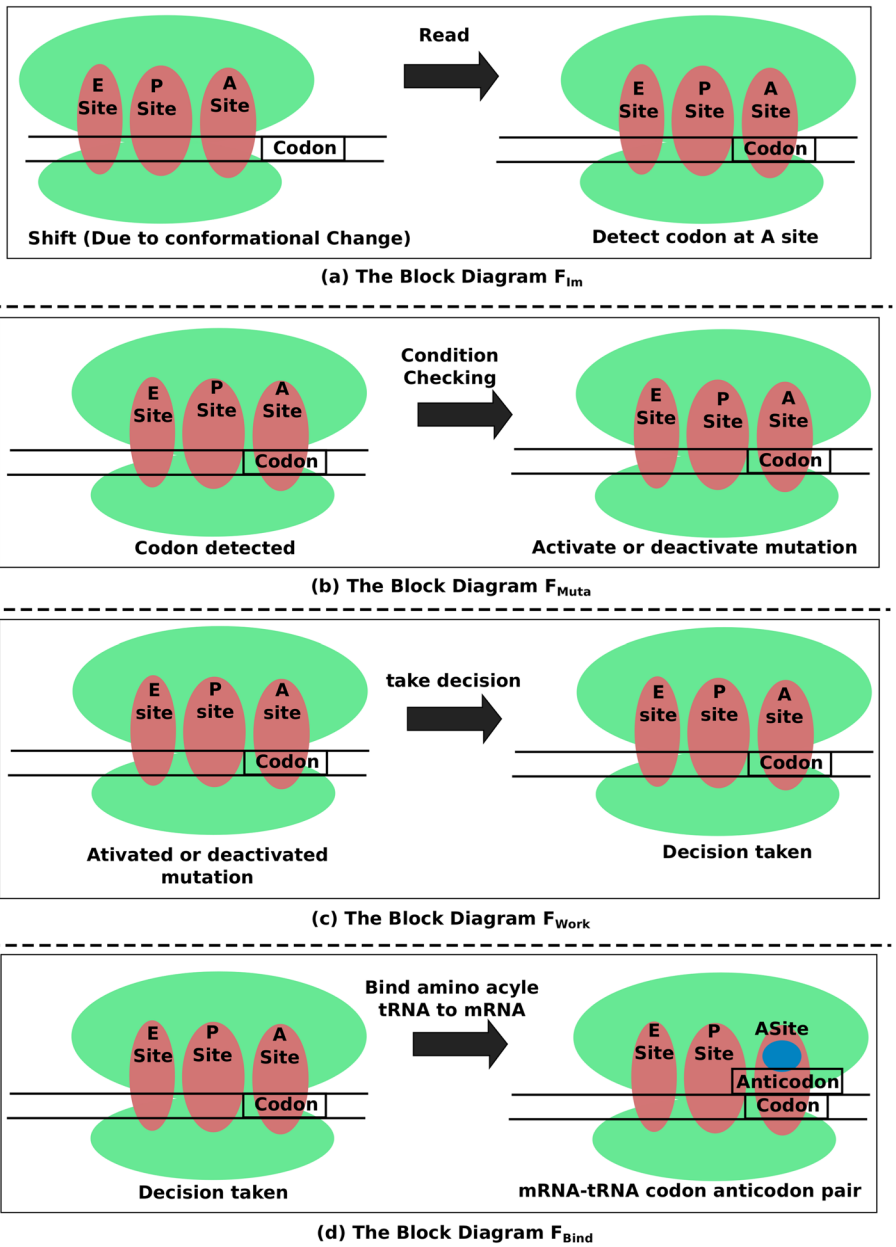


Fig. 8 The block diagram of all functions in practical protein synthesis

$$F_{Im}(M) \rightarrow D \tag{16}$$

The goal of the proposed model is the correlation between gene sequence and proteins. Thus, we assume that the function F_{Im} detects a triplet codon set rather than one triplet codon and returns the set as the form of instruction to develop in this model. In Algorithm 1, the working process of the function F_{Im} is discussed step by step. In Algorithm 1, each triplet codon set of a gene sequence of mRNA M is read and stored in a two-dimensional array, $D[t][s]$ (, where t denotes the position of gene sequence and s is a position of triplet codon set) as a triplet codon set. We

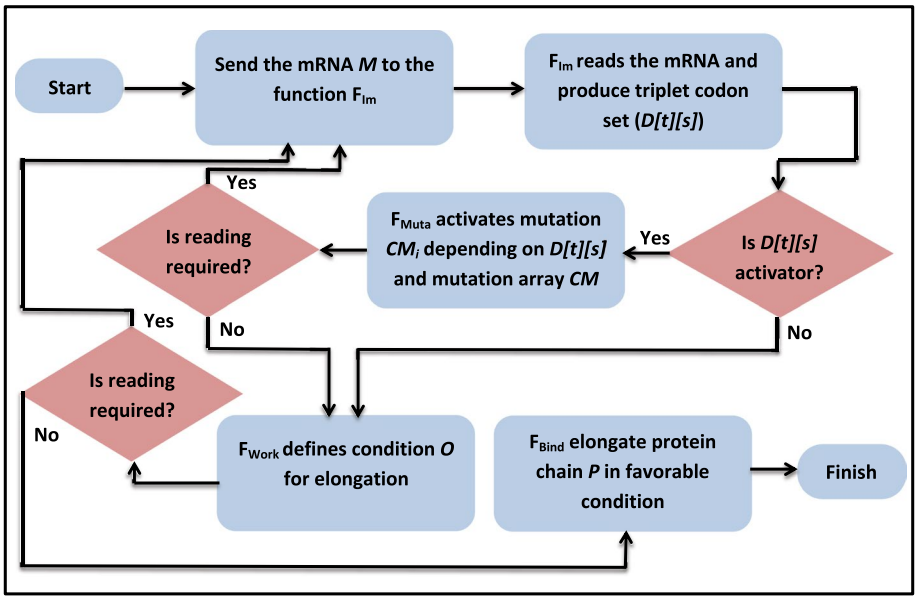


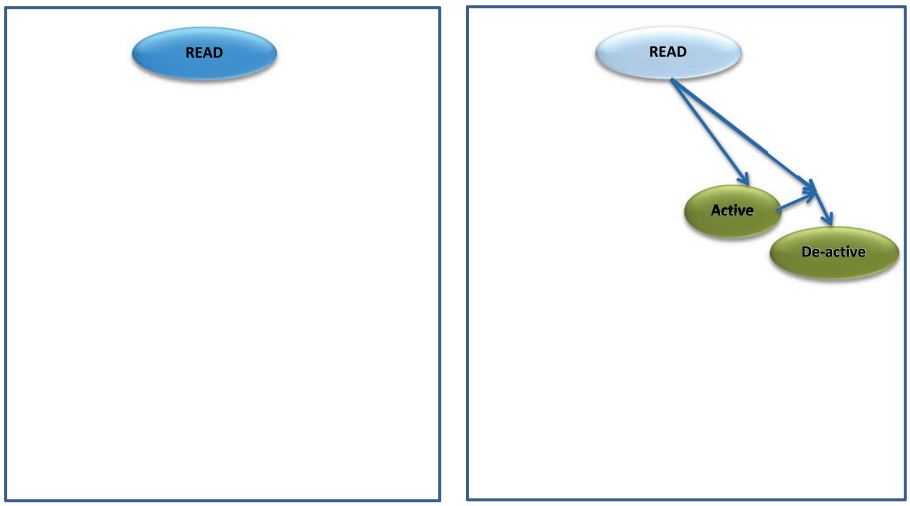
Fig. 9 The block diagram of the workflow of the mathematical model of mRNA-Ribosome System

have already mentioned that the mRNA is a four-dimensional variable. Among these four dimensions, a two-dimensional variable (with dimensions v and w) represents a triplet codon set of a gene sequence set. The F_{Im} function detects one triplet codon set from a gene sequence set at a time and generates the output as a two-dimensional array. Now, we consider the mRNA of Eq. 1, where $M[q]$ is the q^{th} gene sequence set of mRNA M. Again, let us consider each output array of the function F_{Im} contains v number of gene sequence in $M[q]$ gene sequence set, such as $L_1^{[q]}, L_2^{[q]}, L_3^{[q]}, \cdots, L_v^{[q]}$ according to Fig. 4. Thus, the outer loop of Algorithm 1 is repeated v times. In the next step, the gene sequence $L^{[t]}$ contains w number of triplet codon set such as $K_1^{[q][t]} K_2^{[q][t]} K_3^{[q][t]} \cdots K_w^{[q][t]}$ as in Fig. 4. Thus, the inner loop of Algorithm 1 is repeating w times. F_{Im} function accepts a gene sequence set of the reporter mRNA $M[q]$ as input and generates a two-dimensional array $D[t][s]$ as output to instruct the next function. Here q, t and, s are loop variables.

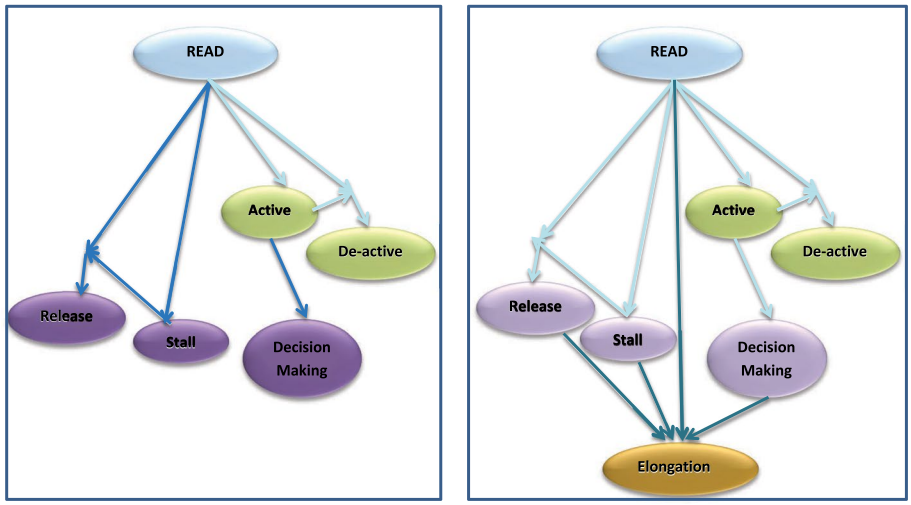
Validation with practical model(I)

The function F_{Im} performs the read operation. The initial stage of translating an mRNA codon is performed in this operation. In the practical scenario of protein synthesis, it is referred to as reading microoperation [39].

Read Among all microoperations, only *read* operation can shift mRNA by a codon sequence (depending on the operation, one or more triplet codon sequences can shift). This operation returns the read triplet codon and advances to the next one to read at the next instant. This microoperation returns a gene sequence for the function F_{Im} of the proposed model. As the read operation detects a triplet codon, every other operation depends on this read microoperation. The dependency dia-



a) The dependency diagram of function F_{Ig} b) The dependency diagram of function F_{op}



c) The dependency diagram of function F_{work} d) The dependency diagram of function F_{tRNA}

Fig. 10 Building of Dependency Graph for all the functions in the mathematical model of mRNA-Ribosome System

gram in Fig. 10a depicts this microoperation. Any other microoperation can start only after obtaining the read codon.

- F_{Muta}: The function F_{Muta} takes the gene sequence D from function F_{Im} after reading the mRNA M and an array of mutations CM as inputs. Then, the function produces the index of the activated mutation as output if the current gene sequence turns out to be an activator of some mutation. In other scenarios, this function generates-*1* as output. The Eq. 17 demonstrates the relationship between the variables. The function F_{Muta} takes two variables, mRNA codon (D) and mutation array (CM), and generates a specific condition (E) as output.

$$F_{Muta}(D, CM) \rightarrow E \tag{17}$$

The function F_{Muta} checks the condition of the translation of the gene sequence. The Algorithm 2 demonstrates the function F_{Muta} in step by step manner. Algorithm 2 takes D and CM as input. D is the read mRNA codon, and CM is the mutation array. This function checks whether the input triplet codon sets satisfy a condition of activation or deactivation of a mutation. In the Algorithm 2, the index of a mutation is denoted by temporary variable i. However, if there is a match, the output variable(E) modifies its contents with the value of i.

Validation with practical model(II)

This function is used to activate or deactivate some conditions that can regulate the elongation process of the proposed model. In the actual model, the ribosomal protein and RNA regulate the elongation of the protein chain. The Fig. 10b depicts the dependency diagram for this function in an actual model. Therefore, the description of the microoperations for the function of F_{Muta} are as follows.

- *Activation Activation* operation activates a proper mutation by reading the appropriate input codon from reporter mRNA where it is applicable. So, the activation operation also depends on the *read* operation.
- *Deactivation* If a specific mutation is activated, the deactivation of that mutation can occur upon reading the appropriate deactivating sequence. So, the deactivation operation depends on both the *read* and the *activation* operation.

- F_{Work}: This function decides the elongation process of the protein chain. The decision depends on the currently activated condition acquired from function F_{Muta}. The function F_{Work} takes a gene sequence (D) and the output (E) of the function F_{Muta} and generates the output decision O. Equation 18 shows the relationships among the variables.

$$F_{Work}(D, E) \rightarrow O \tag{18}$$

The working technology of the function F_{Work} is stated in Algorithm 3. This function executes one of the following operations depending on the value of E. (a) If the value of E is -1, the content of D[t][s] may transfer to the output O. (b) Otherwise, we test each codon from the output gene sequence of D[t][s] according to the activated condition and set the index for transferring the codon to output O. O is a structure variable and consists of two variables. It contains a flag (flag1: a boolean variable set if 1; clear if 0) to indicate the decision. The default value of *flag*1 is 1. However, for a particular situation, it can be changed depending on the value of E. The other element of O is the two-dimensional array to hold the triplet codon set.

Validation with practical model(III)

This function takes the decision to perform one of three microoperations. It may decide to elongate, stall, or release the protein synthesis process. The Fig. 10c contains the dependency diagram for this function. The microoperations for this function are as follows.

- *Stall and release* During protein synthesis, *stall* and *release* operations get executed upon reading special gene sequences *SecM* and *SecA*, respectively, from mRNA. After *stall* operation, no amino acid can get attached to the protein chain until matching *release* is executed. Like other operations, these operations come into action after certain sequences are read. Hence, they depend on the *read* operation.
- *Decision-Making* Depending on the number of possible output digits, currently, activated mutation selects one of the digits. We have already specified that each of these digits is associated with a triplet codon and one amino acid to attach to the protein chain. This selection is a property of activating mutation. The system performs one of the two operations for each codon in the output gene sequence. The operations are skipped or selected. In skip operation, the system reads the codon but does not attach any amino acid. On the other hand, the system reads triplet codon and attaches the amino acid to the protein chain for the select operation. These operations depend on two microoperations, *read* and *activation*, as are shown in the Fig. 10c.

• F_{Bind}: The function F_{Bind} elongates the protein chain. This function takes the output O of the function F_{Work} and elongates the protein chain with suitable amino acid (a) in a favorable condition. One point to note is that O is the set of triplet codons (gene sequence), and P is the protein chain (set of amino acids). The index t defines the position of a triplet codon in a triplet codon set. Equation 19 depicts the relationship between the variables.

$$F_{Bind}(O) \rightarrow A \tag{19}$$

Through this function, we want to show that a ribosome brings appropriate tRNA to the ribosomal *A site* to attach the desired amino acid to the peptide chain in a defined condition. The function F_{Bind} is the inherent property of the corresponding ribosomal computing. Algorithm 4 shows the steps of this function.

Validation with practical model(IV)

This function imitates the property of microoperation elongation. The dependency diagram for this function is depicted in Fig. 10d. The description of microoperation elongation is given below.

- *Elongation* During protein synthesis, the protein chain gets elongated by a single amino acid after each step of *read*. Elongation can also occur if the read codon is not within *stall* and matches the codon for *release* operations. Moreover, elongation occurs on behalf of the selected output codon from the output gene sequence in ribosomal computing. However, the elongation process depends on *read*, *stall*, *release*, *decision-making*. *Elongation* augments the size of a protein chain by a unit amino acid.

Algorithm 1 Implementation algorithm for function F_{Im}.

Algorithm 1: Implementation algorithm for function F_{Im}.

Input: Input Numbers: reporter mRNA $M = M[q]$
Output: Gene sequence: $D[t][s]$

```
incr ← 1;
for t ← 1 to v by incr do
    Initialize D as blank;
    for s ← 1 to w do
        /*Insert each triplet codon set of M[q] gene sequence in D*/
        D[t][s] ← Detect(M[q][t][s]);
    end
    return D;
end
```

Algorithm 2 Implementation of F_{Muta}.

Algorithm 2: Implementation of F_{Muta}.

Input: D, CM
Output: E

```
r ← Number of mutations in CM;
g ← Number of activator in CM;
h ← Number of de-activator in CM;
E ← −1 ;
/*Check each element of triplet codon set for each element of mutation array*/
for i ← 0 to r do
    /*Check for activator conditions in the subarray CM_r^a of array CM*/
    for j ← 0 to g do
        if D is an element of CM_r^a then
            /*Store the position of activator mutation in E*/
            E ← i;
        end
    end
    /*Check for de-activator conditions in the subarray CM_r^d of array CM*/
    for j ← 0 to h do
        if D is an element of CM_r^d then
            /*Store the position of deactivator mutation in E*/
            E ← i;
        end
    end
end
return E;
```

Algorithm 3 Implementation of F_{Work}.

Algorithm 3: Implementation of F_{Work}.

Input: $D[t][s]$, E, *decision* flag1
Output: $O flag1, index$

```
flag1 ← 1;
incr ← 1;
o ← blank;
t_o ← blank;
if E is −1 then
    O ← D[t][s];
end
else
    flag ← E;
    for incr ← 1 to w do
        /*t_o is the output gene sequence*/
        t_o ← t;
        O ← D[t_o][incr];
    end
end
O ← flag
return O;
```

Algorithm 4 Implementation of function F_{Bind}.

Algorithm 4: Implementation of function F_{Bind}.

Input: Output triplet codon O
Output: Protein chain P

```
if t is t_o then
    /*Append the amino acid a to the protein chain P*/
    P ← a;
end
else
    Perform the gene sequence-specific task;
end
```

Illustrative evaluation of the model

Now, we verify the proposed model by illustrating an example. We discuss the XOR operation provided in [18]. At first, we consider that ribosomal computing performs operations in binary mode to perform this operation. Let us assume $B_1B_2B_3$ codon in mRNA and amino acid L in protein chain is used to represent bit 0 and codon $B_4B_5B_6$ in mRNA and amino acid A in protein chain represent 1. Now, let us take two input numbers 1100 and 1001 in DNA sticker model for XOR operation. Therefore, the mRNA for the XOR operation is given in 20.

Table 2 Properties of different Mutations required for XOR Gate [40]

Mutations	Activator	De-activator	Action
M_2	$B_1B_2B_3B_1B_2B_3$	SM_2	Select current codon, Skip next codon
M_3	$B_1B_2B_3B_4B_5B_6$	SM_3	Skip current codon, Select next codon
M_5	$B_4B_5B_6B_1B_2B_3$	SM_5	Skip current codon, Select next codon
M_8	$B_4B_5B_6B_4B_5B_6$	SM_8	Select current codon, Skip next codon

$$
\begin{aligned}
&SD|\ AUG|\\
&\quad SecMB_4B_5B_6B_4B_5B_6SecA\ B_1B_2B_3\ B_4B_5B_6\ SM|\\
&\quad SecMB_4B_5B_6B_1B_2B_3SecA\ B_1B_2B_3\ B_4B_5B_6\ SM|\\
&\quad SecMB_1B_2B_3B_1B_2B_3SecA\ B_1B_2B_3\ B_4B_5B_6\ SM|\\
&\quad SecMB_1B_2B_3B_4B_5B_6SecA\ B_1B_2B_3\ B_4B_5B_6\ SM|\\
&\quad UAA
\end{aligned}
\tag{20}
$$

We give the mutation set for performing the xor operation in Table 2.

The following steps are performed for the xor operation on mRNA20.

1. At first, mRNA20 binds to ribosome using SD sequence to prepare the mRNA-Ribosome system.
2. Then starts codon AUG is read by the function F_{Im}, and no mutation is activated. It initializes the protein synthesis by attaching *Methionine* amino acid to the protein chain.
3. The next sequence set is read by F_{Im}. The input sequence of the first codon set of this sequence ($B_4B_5B_6B_4B_5B_6$) activates mutation M_8 by the function F_{Muta}. Then, the $B_1B_2B_3$ output codon is selected by the function F_{Work}. The function F_{Bind} attaches the amino acid L to the protein chain. So, the protein chain is ML
4. Similarly, the function F_{Im} reads next sequence set. In this set, input codon sequence $B_4B_5B_6B_1B_2B_3$ activates the mutation M_5 for function F_{Muta}. Then the second output codon $B_4B_5B_6$ is selected by the function F_{Work}. The amino acid A is attached to the protein chain by the function F_{Bind}. So, the protein chain is MLA
5. After that, function F_{Im} reads next sequence set. The F_{Muta} function activates mutation M_2 for $B_1B_2B_3B_1B_2B_3$ input sequence of the set. Then, function F_{Work} chooses the output codon $B_1B_2B_3$. The function F_{Bind} attaches amino acid L to protein chain and the chain is $MLAL$
6. Then after, the next sequence set is read by F_{Im}. The function F_{Muta} activates M_3 for input sequence $B_1B_2B_3B_4B_5B_6$. The output codon $B_4B_5B_5$ is selected by F_{Bind}, and amino acid A is attached to the protein chain by F_{Bind}. So the protein chain is $MLALA$
7. After reading the stop codon UAA in a similar manner, the system stops the protein synthesis. Then, the protein chain is detached from the ribosome. The M is discarded from the chain, the final output chain is $LALA$, and its binary value is 0101. It is the correct result of the XOR operation between the input 1100 and 1001.

Complexity of the mathematical model

The mathematical model is not the replica but the downsized replica of an actual model. The proposed model focuses on the internal working principle of the protein synthesis process for a single ribosome. Several computational works are performed using ribosomal computing. In the proposed model, we have illustrated a general abstraction for all computational work of ribosomal computing. During the development of the model, we have observed the following points:

- We have illustrated the direct relationship between input (mRNA) and output (protein chain).
- This model is not developed to detail molecular analysis for simplicity, but the functional level abstraction of the components of protein synthesis is adequately explained. The model differs from the actual model at some points of observation, although it becomes simple by avoiding chemical details.
- We have considered the mRNA a four-dimensional array, but it is a single-dimensional array of triplet codons in the actual model. Several amino acids may be present in the protein chain, but we consider only the output amino acid for simplicity. However, the distance of an output codon is measurable.
- To design the model, we have considered different conditional control factors of protein synthesis. We believe conditions are an array of mutations. Though mutation has some other significance in the biological domain, we have considered each conditional change as a mutation to keep the model uniform and simple.
- The cause and effect of the ribosome stalling motif on protein synthesis is one of the significant conditions for implementing the model. We have ignored many other controlling factors to keep the model simple. After a detailed observation of the experimental work, we infer that mutations in a gene sequence or some macrolide drugs or antibiotic drugs can stall protein synthesis. The stalled process can start again for some gene sequences or the presence of some other biological substances [41]. Staling may arise due to the protein chain interaction with the output tunnel of the ribosome or the ribosome's conformation to a protein chain. During the stalling, the protein chain is not elongated. So, some triplet codons may not be translated for this. In developing this model, we can observe the significance of ribosome stalling for protein synthesis.

Applicability of the proposed model

A mathematical model of biomolecular computing (ribosomal computing) is studied and developed by modeling the functional operations of the actual biochemical process (Protein synthesis). The proposed model finds its applicability manifold in both biological and computing domains. A few prominent application areas may be mentioned as follows.

Biomolecular computing

The mathematical model of ribosomal computing is developed by defining the variables and functions required for a mathematical representation of the protein synthesis process in a cell. The straightforward representation of the variables and functions manifests the working principle of this computing. Therefore, the proposed model may be utilized to address, analyze, and solve conventional computational problems in an alternative way, a new biomolecular computational technique.

- *Useful to design algorithm by hiding biological details* In the present model, the biological information of ribosomal computing is enveloped by mathematical symbols and equations. The properties of the biological elements are converted into math-

ematical rules. Therefore, one can develop a new problem-solving algorithm using ribosomal computing to avoid biological complexity.

- *Quantitative data evaluation for a computing system* In the proposed model, each bit operation is performed by processing a unit amount of information. This quantitative measurement is useful for quantitative data analysis of the system. One can analyze the performance parameters such as execution time, power requirement analysis for this computing system.
- *Developing a standard biological computational unit* The proposed mathematical model provides a general framework for ribosomal computing. Developing a common biological computational unit that can perform different computational tasks is useful [42]. As this is a biological unit, it is compatible with a biological organism. Thus, this computational unit is suitable as a calculative unit in those technologies that can control a biological unit or process externally. This unit can be applied in calculating the amount of drug release and cumulative drug release of multiple drug delivery networks.

Protein analysis

The proposed model clearly shows the internal functional structure of mRNA and other components of protein synthesis and reveals the working technology of protein synthesis. Therefore, anyone can further explore the individual component of the system if they wish. For example, suppose someone wants to test the effect of different mutations on protein synthesis. In that case, concentration on the composition of the mutation, the behavior of the mutation, and its applying position may be of prime importance for discussion. However, the other components of the system remain unchanged. This model focuses on protein formation by applying control conditions over the mRNA gene sequence. Therefore, it can specify the relation of the control condition over mRNA and the properties of a protein. The characteristic properties of a protein depend on the structure and function of the protein.

- *Protein structure prediction* Different types of stalling motifs may break the translation at the point induced and generate a breakage in the chain. This condition leads to the folding of the protein chain incorrectly. Now, suppose anyone simulates the model in a computer and checks the relationship between each fragment of the protein chain and the combination of mRNA gene sequence and condition. In that case, they will get a list of protein fragments with their generating conditions. Then, one can combine different fragments in all possible combinations and record the structure for all of them. One can obtain all possible protein structures from this record for different conditions. If someone performs a practical protein synthesis process, he can easily predict a protein's structure from the experimental scenario's available conditions.
- *Protein function prediction* The function of a protein depends on its structure. The correlation between different components and the association between gene sequence and protein structure can predict the structure. The current model leads

us to this analysis in a similar manner. So, we can predict the function as well as the structure.

Medical system

Abnormal protein synthesis is a fundamental cause of several diseases, and some antibiotic drugs are used as remedies. Protein synthesis and its different components are major areas of focus for the proposed model. This model can also analyze the cause and solution of abnormal protein synthesis. Thus, the model is significant in the medical system.

- *Drug design* The proposed mathematical model can further be extended to implement a ribosome simulator. This simulator is helpful for testing different conditions of protein synthesis. Thus, from this testing, one can understand the cause and effect of abnormal protein synthesis in the living body. The possible solution of abnormal protein synthesis can also be predicted. Thus, from the proposed model, one can understand the cause and solution of abnormal protein synthesis. This cause is a crucial feature of drug design.

Conclusion

An mRNA ribosome system or ribosomal computing system has recently seen its dawn. However, it has already shown great promise via implementations of logic gates, sequential circuits, and some arithmetic operations. These works imitate an automated process for performing the tasks. Thus, these works demonstrate that automatic computation is possible in the ribosomal computing system. However, designing operations in this biological system is complicated due to the many molecules and complexities involved in the protein synthesis process. For this reason, we have presented a mathematical model in the proposed work. Although some errors may occur in the protein synthesis procedure practically [43], we consider the proposed model for an ideal system to avoid representation complexity. The proposed work aims to simplify the daunting task by carefully analyzing the various units and their protein synthesis functions. At the same time, the model represents them with simple mathematical symbols and formulas. Finally, the verification and validation of the proposed model demonstrate the reliability and complexity of ribosomal computing. Therefore, the proposed work carefully represents the ribosomal computing system in a mathematical wrapper and expects to give this promising system a significant boost in usability and applicability.

Author contributions

Conceptualization, P.C., and P.G.; Methodology, S.S. and A.B.; Software, P.C. and S.S.; Validation, S.M., S.A., B.O.S. and M.O.; Formal Analysis, A.H.A. and E.E.; Investigation, M.O. and B.O.S.; Resources, S.M. and A.R.A.; Visualization, P.C. and A.H.A.; Supervision, S.M. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding

This research was financially supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R393), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/549/45.

Availability of data and materials
The datasets used during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent statement
Not applicable.

Competing Interests
The authors declare that they have no Conflict of interest.

Received: 30 March 2024 Accepted: 23 September 2024
Published online: 03 October 2024

References

1. Plaza PI, Blanco G, Lambertucci SA. Implications of bacterial, viral and mycotic microorganisms in vultures for wildlife conservation, ecosystem services and public health. Ibis. 2020;162(4):1109–24.
2. Karim N, Afroj S, Lloyd K, Oaten LC, Andreeva DV, Carr C, Farmery AD, Kim ID, Novoselov KS. Sustainable personal protective clothing for healthcare applications: a review. ACS Nano. 2020;14(10):12313–40.
3. Weischenfeldt J, Symmons O, Spitz F, Korbel JO. Phenotypic impact of genomic structural variation: insights from and for human disease. Nat Rev Genet. 2013;14(2):125–38.
4. Li M, Wang IX, Li Y, Bruzel A, Richards AL, Toung JM, Cheung VG. Widespread RNA and DNA sequence differences in the human transcriptome. Science. 2011;333(6038):53–8.
5. Lodish, Harvey F. Translational control of protein synthesis. Annu Rev Biochem. 1976;45(1):39–72 (**Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA**).
6. Barnes DJ, Chu DF. Introduction to modelling for biosciences. London: Springer; 2010.
7. Mehra A, Hatzimanikatis V. An algorithmic framework for genome-wide modeling and analysis of translation networks. Biophys J. 2006;90(4):1136–46 (**Elsevier**).
8. Garai A, Chowdhury D, Chowdhury D, Ramakrishnan TV. Stochastic kinetics of ribosomes: single motor properties and collective behavior. Phys Rev E: Stat, Nonlinear, Soft Matter Phys. 2009;80(1): 011908.
9. Zhao Y-B, Krishnan J. Probabilistic Boolean network modelling and analysis framework for mRNA translation. IEEE/ACM Trans Comput Biol Bioinf. 2015;13(4):754–66 (**IEEE**).
10. Khatter H, Myasnikov AG, Mastio L, Billas IML, Birck C, Stella S, Klaholz BP. Purification, characterization and crystallization of the human 80S ribosome. Nucl Acids Res. 2014;42(6):e49–e49 (**Oxford University Press**).
11. Heinrich R, Rapoport TA. Mathematical modelling of translation of mRNA in eucaryotes; steady states, time-dependent processes and application to reticulocytest. J Theory Biol. 1980;86(2):279–313 (**Elsevier**).
12. von der Haar T. Mathematical and computational modelling of ribosomal movement and protein synthesis: an overview. Comput Struct Biotechnol J. 2012;1:1–7.
13. Tour JM. Molecular electronics. Synth Test Compon, Acc Chem Res. 2000;33(11):791–804.
14. Boneh D, Dunworth C, Lipton RJ, Sgall J. On the computational power of DNA. Discret Appl Math. 1996;71(1–3):79–94.
15. Ehrenfeucht A, Harju T, Petre I, Rozenberg G. Characterizing the micronuclear gene patterns in ciliates. Theory Comput Syst. 2002;35(5):501–19.
16. Balan MS, Krithivasan K, Sivasubramanyam Y. Peptide computing-universality and complexity. Berlin: Springer; 2001. p. 290–9.
17. Nolte DD. Mind at light speed: anew kind of intelligence. New York: Simon, and Schuster; 2001.
18. Pratima C, Mayukh S, Prasun G. Computing in ribosomes: performing boolean logic using mrna-ribosome system, VLSI (ISVLSI), 2016 IEEE computer society annual symposium on, 2016; pp. 260–265, USA.
19. Pratima C, Mayukh S, Prasun G. Computing in ribosomes: implementing sequential circuits using mRNA-ribosome system, nanoelectronic and information systems (iNIS), In: 2016 IEEE International Symposium on, 2016. pp. 230–235, India.
20. Pratima C, Prasun G. Realizing all logic operations using mRNA-ribosome system as a post Si alternative. Nanoelectronic, and information systems (iNIS), In: 2017 IEEE international symposium on, 2017. pp. 40–45, India.
21. Spirin AS. Ribosome as a molecular machine. FEBS Lett. 2002;514:2–10.
22. Merrick WC, Pavitt GD. Protein synthesis initiation in eukaryotic cells. Cold Spring Harb Perspect Biol. 2018;10(12):a033092 (**Cold Spring Harbor Lab**).
23. Lee K, Holland-Staley CA, Cunningham PR. Genetic approaches to studying protein synthesis: effects of mutations at Ψ516 and A535 in Escherichia coli 16S rRNA. J Nutr. 2001;131(11):2994S-3004S (**Elsevier**).
24. Hardesty B, Kramer G. Structure, function, and genetics of ribosomes. Berlin: Springer Science Business Media; 2012.
25. Yusupov MM, Yusupova GZ, Baucom A, Lieberman K, Earnest TN, Cate JHD, Noller HF. Crystal structure of the ribosome at 5.5 Å resolution. Science. 2001;292(5518):883–96.
26. Noller HF. Evolution of protein synthesis from an RNA world. Cold Spring Harb Perspect Biol. 2012;4(4):a003681 (**Cold Spring Harbor Lab**).
27. Kozak M. Point mutations define a sequence flanking the AUG initiator codon that modulates translation by eukaryotic ribosomes. Cell. 1986;44:283–92.

28. Bayfield MA, Thompson J, Dahlberg AE. The A2453-C2499 wobble base pair in Escherichia coli 23S ribosomal RNA is responsible for pH sensitivity of the peptidyltransferase active site conformation. Nucl Acids Res. 2004;32:5512–8.
29. Hodgkin J. Genetic suppression, WormBook: the online review of C. elegans biology. 2005;1–13.
30. Giegé R, Puglisi JD, Florentz C. tRNA structure and aminoacylation efficiency. Prog Nucl Acid Res Mol Biol. 1993;45:129–206.
31. Ibba M, Söll D. Aminoacyl-tRNA synthesis. Annu Rev Biochem. 2000;69(1):617–50.
32. Sprinzl M, Horn C, Brown M, Ioudovitch A, Steinberg S. Compilation of tRNA sequences and sequences of tRNA genes. Nucl Acids Res. 1998;26:148–53.
33. Selmer M, Dunham CM, Murphy FV, Weixlbaumer A, Petry S, Kelley AC, Weir JR, Ramakrishnan V. Structure of the 70S ribosome complexed with mRNA and tRNA. Science. 2006;313:1935–42.
34. Pelham, Hugh RB, JACKSON, Richard J. An efficient mRNA-dependent translation system from reticulocyte lysates. Eur J Biochem. 1976;67:247–56.
35. Yusupova GZ, Yusupov MM, Cate JHD, Noller HF. The Path of Messenger RNA through the Ribosome. Cell. 2001;106:233–41.
36. Nagata S, Hamasaki T, Uetake K, Masuda H, Takagaki K, Oka N, Wada T, Ohgi T, Yano J. Synthesis and biological activity of artificial mRNA prepared with novel phosphorylating reagents. Nucl Acids Res. 2010;38(21):7845–57 (**Oxford University Press**).
37. Orelle C, Carlson ED, Szal T, Florin T, Jewett MC, Mankin AS. Protein synthesis by ribosomes with tethered subunits. Nature. 2015;524:119.
38. Chen H, Bjerknes M, Kumar R, Jay E. Determination of the optimal aligned spacing between the Shine-Dalgarno sequence and the translation initiation codon of Escherichia coli m RNAs. Nucl Acids Res. 1994;22(23):4953–7.
39. Alberts B. Molecular Biology of the Cell. Milton Park: Taylor and Francis Group; 2018.
40. Chatterjee P, Ghosal P. Computing in ribosome: logic gates implementation using the mRNA-ribosome system. CSI Trans ICT. 2017;6:39–50.
41. Ramu H, Mankin A, Vazquez-Laslop N. Programmed drug-dependent ribosome stalling. Mol Microbiol. 2009;71(4):811–24 (**Wiley Online Library**).
42. Chatterjee P, Ghosal P. Realization of arithmetic operations using a combined computational unit in ribosomal computing. J Inst Eng (India): Ser B. 2023;104(2):461–73.
43. Wohlgemuth I, Pohl C, Mittelstaet J, Konevega AL, Rodnina MV. Evolutionary optimization of speed and accuracy of decoding on the ribosome. Philos Trans R Soc B: Biol Sci. 2011;366(1580):2979–89.

Publisher's Note