IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 16, 2018, accepted March 27, 2018, date of publication April 11, 2018, date of current version May 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2825782

Modeling and Analysis of Protein Synthesis and DNA Mutation Using Colored Petri Nets

JINLIANG YANG[1], (Member, IEEE), HAITAO PU[1], JIAN LIAN[1], JASON GU[2], (Senior Member, IEEE), AND MINGQU FAN[1]

[1]Department of Electrical Engineering and Information Technology, Shandong University of Science and Technology, Jinan 250031, China
[2]Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS B3J 2X4, Canada

Corresponding author: Jinliang Yang (jerryjlyang@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703243, in part by the Shandong Province Science and Technology Major Project under Grant 2015ZDXX0801A02, in part by the Shandong Province Higher Educational Science and Technology Program under Grant J14LN79, in part by the Scientific Research Foundation of the Shandong University of Science and Technology for Recruited Talents under Grant 2014RCJJ024, in part by the SDUST Excellent Teaching Team Construction Plan under Grant JXTD20160512, in part by the Jinan campus of SDUST Excellent Teaching Team Construction Plan under Grant JNJXTD201711, and in part by the Chinese Scholarship Council.

ABSTRACT In molecular biology, protein synthesis is an essential biological process of generating specific proteins in living systems. Many mathematical models are designed to characterize the process of the flow of genetic information from deoxyribonucleic acid (DNA) to protein; however, most of them cannot provide detailed and observed steps to describe and analyze this fundamental biological process. Colored Petri Net (CPN), as a mathematical method widely applied in the process analysis of discrete event dynamic systems, has increasingly become an innovative and efficient theoretical approach for exploring the biological processes. Aiming at describing the entire process of protein synthesis, a CPN-based model has been designed that successfully and intuitively represents the flow of genetic information involving transcription and translation. DNA mutations are permanent alterations in the nucleotide sequence of the DNA strand, while different types of mutations have various effects on the synthesized proteins. Based on the proposed protein synthesis model, we put forward a new CPN model to identify the type of mutation, which is beneficial for analyzing whether the mutation has impacts on the structure or function of the produced protein. The mutation position and bases mutated rate are obtained by contrasting the nucleobases on DNA sequences, while the mutation type is determined via the alignment of the amino acids in the polypeptide chain. The model's effectiveness and accuracy are illustrated by biological mutation examples, indicating this method offers great superiority in modeling and analyzing the complex biological processes.

INDEX TERMS Protein synthesis, modeling, deoxyribonucleic acid (DNA) mutation, type determination, colored Petri Net.

I. INTRODUCTION

Molecular Biology is a branch of science that aims to reveal the essential biological processes in the cell by studying the structure and function of fundamental cellular molecules such as nucleic acids and proteins. As a frontier research in life science, a certain number of researchers have focused their attention on a variety of academic technologies in this area [1], such as biochemistry, bioinformatics, computational biology, genetic engineering, nanotechnology, etc. Protein synthesis is one of the most fundamental biological processes, revealing the process that individual cells produce specific proteins. The Central Dogma of Molecular Biology provides an explanation for understanding the flow of genetic information. The description and modeling for the process of protein synthesis can be implemented on various levels, and a more systematic approach would be beneficial for researchers to understand the complicated biological systems on a microscopic level [2].

The textual representation model provides a clear and complete description of the whole process of genetic information flow. However, the lack of systematic model has been still regarded as the bottleneck among the research fields such as molecular biology, information science, and their discipline-crossing research. Hence, recently many traditional mathematical methods and tools have been applied for modeling and analyzing the cellular biological systems [3], such as gene expression, metabolic processes, genetic regulatory networks, etc. These techniques include numerical calculation, matrix theory, finite state machine (FSM) theory, Hidden Markov Model, ordinary differential equation, Petri net, and so on. In 2006, Zhang *et al.* [4] proposed a numerical calculation formulation of deoxyribonucleic acid (DNA) computation. In 2009, Gao *et al.* [5] put forward a mathematical framework for interpreting the Central Dogma of Molecular Biology based on matrix theory. A FSM model has been designed to regulate gene mutation in 2013 [6]. Although these models provide several mathematical representations of genetic information flow from an overall aspect, it is indispensable to describe the protein synthesis from a more microscopic view, i.e., the model can depict the detailed states and changes of the process in the level of nucleobases and amino acids.

The Hidden Markov Model (HMM) is one of the most successful modeling tools applied in the field of computational biology to solve a surprisingly wide range of multiple biological sequence alignments problems, including gene prediction [7], multiple sequence alignment of protein families [8], prediction of protein secondary structures [9], biological sequence database searching [10], etc. As a powerful tool for modeling and statistical analysis, HMM owns many advantages in computational biology. However, the method still does suffer from a major disadvantage of being unable to describe the biological process, like protein synthesis, mainly because it ignores the relationships among the nucleobases, amino acids or proteins that are involved in the whole process of genetic information transmission.

Ordinary differential equation (ODE) models also play an important role in studying the biological systems by providing accurate mathematical expressions and equations derived from the integration of experimental data and analysis of biological networks [11]. Since the process of protein synthesis consists of a series of standard biochemical reactions, it is possible to accurately build an ODE-based model to describe the process by analyzing experimental data [12]. Although ODE approach owns a reasonable ability to describe dynamical aspects of the biological process, it has disadvantages in representing the whole process of protein synthesis. For example, ODE is deterministic and continuous, which makes it unsuitable to describe the systems that subject to stochastic events [13], such as biological processes involving interaction with the DNA, mRNA or protein.

Petri net (PN) is well-known as an effective tool to model and analyze the discrete event systems [14], [15]. Since the biological processes own typical feature of discrete event systems, PN is regarded as the powerful innovative method for describing and analyzing molecular proce4sses, such as protein production process [16], metabolic pathways [17], and gene regulation [18]. CPN is a significant extension of classical PN and it marks the tokens with different colors represented various properties of objects [19]–[22]. Recently, many researchers focus on designing CPN based models of biological processes, including enzymatic reaction [23], metabolic pathways [24], and gene regulatory networks [25]. Heiner and Gilbert [26] proposed a CPN model for the phase variation in bacterial colony growth. Liu *et al.* [27] constructed a CPN approach for modeling and simulating reaction–diffusion systems in systems biology. Carvalho *et al.* [28] designed a multilevel CPN model to reproduce the dynamics of the steps in the infection process and innate immune response. Pennisi *et al.* [29] built a novel CPN-based methodological approach and applied the approach in the immune system response at the cellular level. Gratie and Gratie [30] presented an algorithm for building a composition CPN models and implemented the data refinement of reaction-based models. Liu *et al.* [31] summarized the application of CPN for multilevel, multiscale and multidimensional modeling of biological systems. Therefore, CPN has increasingly become a useful tool that establishes theoretical foundation for studying the classic biological processes.

DNA mutation is a permanent alteration in the DNA strand which may cause the changes in the sequence of amino acids on the polypeptide chain. Mutations have diverse effects on protein product depending on where they happen and whether they change the biological function of essential proteins. Therefore, mutations have been divided into different types according to the effects of the mutations on chromosome structure, on protein function, or on protein sequence [2]. It is necessary to design an intuitive model for identifying the mutation type by contrasting the DNA sequences. The model will be not only helpful in studying how the mutations influence the genetic information, but also conducive to analyze whether they affect the structure and function of the produced proteins. A CPN model is proposed to determine whether mutations occur by comparing the DNA strand and mutated one [32]. Based on this work, CPN models for describing genetic information transfer [33] and identifying the polarities of amino acids [34] have been designed later. However, the proposed CPN model in [32] used a large number of places and transitions, which does not make good use of the concurrent characteristics of CPN. In 2017, the authors revised the model for determining the mutation type and simplified the process to achieve the synchronization of alignment in both the bases on the DNA strands and the amino acids located on the synthesized polypeptide chains [35]. The work, to some degree, improved the capabilities of the model, reduced the model size and shorten the processing time.

This paper is an effective extension and improvement of the work shown in [35]. First of all, starting with the process of protein synthesis, we established a CPN model to describe and analyze the whole process of gene information transfers from DNA to RNA and to protein in several stages. This provided the theoretical and analytical basis for the analysis of gene mutation and determination of mutation types. Then, we improved the mutation type determination model to obtain the mutation position and nucleobase mutation rate. Mutation type is determined by using comparative analysis of the amino acids sequences. Specifically, we revised some places and transitions and enriched the color sets and the priorities of transitions in the model. We tried to use the actual four types of mutation examples in nature to verify the validity and rationality of the model. In these CPN models, the places and token colors are defined as the states of different stages in the process of protein synthesis. The firing rules and guard functions of transitions are defined according to the state changes or analysis operations during the process of gene information flow. The operation of the models can be divided into different stages, representing a number of steps of the process of protein synthesis.

There are two goals for us to choose CPN as the preferred method to model and analyze the basic biological processes involved in protein synthesis and DNA mutation. Firstly, the model must be able to simulate the various stages of protein synthesis, such as transcription, identifying start codon and stop codon, and translation. Secondly, through analyzing the operation procedure of the model, it can intuitively display the states and reflect the interactions among DNA, mRNA, bases, amino acids and proteins during protein synthesis, which helps to deepen our understanding of the genetic information transmission. In other words, compared to the final simulation results of the model, we pay more attention to the detailed operation process during the whole simulation. Therefore, we use the new syntax form of step-by-step graph description to demonstrate the operation procedure of the proposed model. Although our own reasoning-based analysis process is somewhat different from other CPN simulation software, it provides an effective and intuitive way to illustrate the detailed states of all elements involved in the whole process of protein synthesis.

This paper is organized as follows. Section II introduces the definition and properties of CPN. In Section III, a protein synthesis CPN model is designed for intuitively describing the flow of genetic information from DNA to protein. Examples illustrate and analyze the model's accuracy and effectiveness. In Section IV, we built a mutation type identifying model that contrasts the bases on DNA strands and amino acids on the polypeptide chains respectively. Some biological examples are demonstrated to validate the model can identify the type accurately in a finite number of steps. Finally, the conclusion is summarized in Section V.

II. PRELIMINARIES

The definition and function of CPN are stated in this section. Besides, an example of chemical reaction model is given to illustrate the operation process of a CPN model.

Definition 1 (CPN [20], [21]): A CPN is a 9-tuple, $CPN = (\Sigma, P, T, A, N, C, G, E, I)$, where:

1) $\Sigma = (\Sigma_1, \Sigma_2, \dots)$ is a finite set of *color sets* defined by non-empty types. This set represents all possible color types, operations and functions in the model;
2) $P = (P_1, P_2, \dots, P_m)$ is a finite set of *places*;
3) $T = (T_1, T_2, \dots, T_n)$ is a finite set of *transitions*;
4) $A = (A_1, A_2, \dots, A_s)$ denotes a set of *directed arcs* such that $P \cap T = P \cap A = T \cap A = \emptyset$;
5) N denotes a *node function*, which maps from A into $(P \times T) \cup (T \times P)$;
6) $C = (C_1, C_2, \dots, C_m)$ is a set of *color functions*, and it maps from *places* in P to *colors* in Σ;
7) $G = (G_1, G_2, \dots, G_n)$ is a set of *guard functions*, which maps each *transition* into a Boolean expression: true or false, to decide the transition will be fired or not;
8) E denotes an *arc expression function*, representing the weight function of *directed arcs* in set A;
9) $I = (I_1, I_2, \dots, I_m)$ is an *initialization function*, where I_i represents the number of tokens stored in P_i.

Definition 2 (Transition Enabled and Fired [20], [21]): A Transition T is ***enabled*** if each input place P_i of T is marked with at least $W(P_i, T)$ tokens, where $W(P_i, T)$ represents the weight of the directed arc from P_i to T. The transition T is said to be ***fired*** if and only if the transition is enabled and its guard function evaluates to true.

Definition 3 (Firing Rule): When an enabled transition T is fired, it removes $W(P_i, T)$ tokens from each input place P_i of T, and adds $W(T, P_j)$ tokens to each output place P_j of T, where $W(T, P_j)$ is the weight of the directed arc from T to P_j. That is to say if the transition is fired, the number of tokens either removed or added in the model is equal to the weight of the corresponding directed arcs.

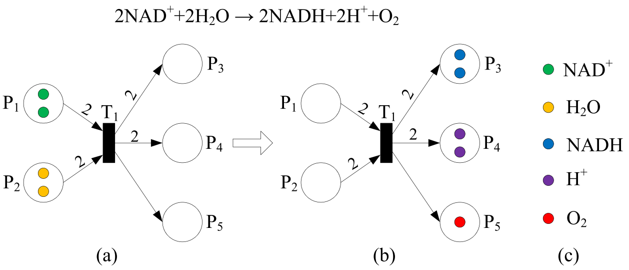


FIGURE 1. CPN model of a chemical reaction (redrawn from [36]).

To illustrate the operation process of CPN model, we refer to a simple model that describes a chemical reaction of relight-induced phosphorylation [36], as shown in Fig. 1. This model has two input places (P_1 and P_2), three output places (P_3, P_4, and P_5), and one transition T_1. Here, T_1 stands for the chemical reaction process; input places and output places can store tokens that represent the different reactants and resultants of this chemical reaction, respectively; and the number marked on the edges means the weight of the directed arcs. Notice that P_1 and P_2 each have two tokens represented by the color dot inside each place initially, as shown in (a), which means T_1 is enabled and can be fired. As the firing of T_1, the tokens stored in the input places are consumed, while the output places are filled by the tokens that denote the resultants of reaction, shown in (b). In Fig. 1(c), different colors of the tokens stand for the reactants and resultants involved in the chemical reaction.

III. CPN MODEL OF PROTEIN SYNTHESIS PROCESS

Proteins are the end product of many metabolic processes, and are regarded as the most fundamental part of life on earth. The process of protein synthesis transcribes the DNA strand to mRNA, and in turn translates the codons on mRNA into a polypeptide chain of the proteins.

A. PROCESS OF PROTEIN SYNTHESIS

Protein synthesis process is generally described as four steps, including transcription, initiation, translation elongation, and termination [37]. Transcription is the first step in which the information in DNA is transferred to mRNA molecule in the nucleus. Then the mRNA moves out of the nucleus and travels to the cytoplasm. Initiation denotes the recognition of *Start Codon* (*AUG*) on mRNA by the ribosome, signaling the start of translation. Translation elongation is the third step in which the mRNA strand is read according to the genetic code, transferring the DNA sequence to a chain of amino acids. With the ribosome moving from the 5'-end to the 3'-end of the mRNA, the polypeptide chain elongates by sequentially adding the amino acids encoded by the codons, resulting in a growing protein. Elongation continues until one of three *Stop Codons* (*UAA*, *UAG*, and *UGA*) appears on mRNA, which signals the termination of translation. Then, the polypeptide chain releases from the mRNA and the ribosome.

B. CPN MODEL OF PROCESS OF PROTEIN SYNTHESIS

Based on the description of genetic information flow, we build a CPN model of protein synthesis, as shown in Fig. 2. In accordance with the steps of protein synthesis, we divide the operation procedure into five stages in the model.

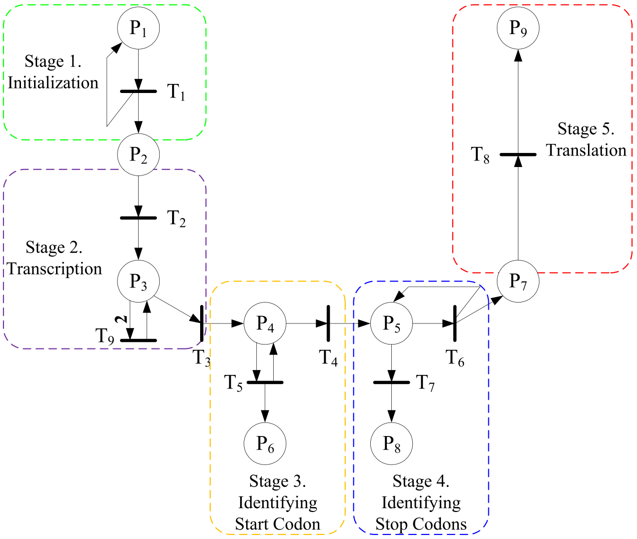


FIGURE 2. CPN model of the protein synthesis process.

1) *Initialization*. The DNA strand is read in the CPN model and stored in P_1, which stands for the genetic information contained in the given DNA strand that is ready to transfer to a corresponding mRNA molecule, indicating the beginning of the process of protein synthesis.

2) *Transcription*. Based on the type of given DNA chain, either coding or template strand, it can be transcribed into mRNA chain in accordance with the base conversion rules. The conversion rule is defined as: i) If a coding DNA strand is read, the conversion rules are $A \rightarrow A$, $C \rightarrow C$, $G \rightarrow G$, and $T \rightarrow U$; ii) If it is a template DNA strand, the rules are defined by $A \rightarrow U$, $C \rightarrow G$, $G \rightarrow C$, and $T \rightarrow A$.

3) *Identifying start codon*. The stage of translation will begin when the start codon is identified on the mRNA chain. Therefore, we define that the model can determine the start codon in this stage, which represents the start of translation.

4) *Identifying stop codons*. The protein elongation carries on until any one of three stop codons is recognized on the mRNA chain. This stage is to determine any stop codons in the mRNA strand, indicating the termination of translation.

5) *Translation*. The last stage depicts the translation process that shows the stepwise addition of amino acids to the growing polypeptide chain, with the next amino acid attaching to the adjacent produced amino acids.

The basic definition of each element and its symbol expression of the proposed protein synthesis CPN model are stated as follows.

1) COLOR SETS

In the process of protein synthesis, there exist many statuses and features of the various stages. Accordingly, to attribute the different status and characteristics, we define six types of color sets in the model. The name of color sets and their definitions are shown in Table 1.

2) PLACES

The places are defined on the basis of the diverse states and various stages involved in the genetic information transfer. Table 2 gives the detailed definition of the defined nine places in the model.

3) TRANSITIONS

According to the state changes or analysis occurred in protein synthesis, nine transitions are designed in the CPN model. Each transition's definition, input and output places, as well as its firing rule and guard function, are listed in Table 3.

4) WEIGHT OF DIRECTED ARCS

Once a transition is fired, the weight of directed arcs determines how many tokens have been taken from the input places and distributed into the output places. The weight of directed arc from P_3 to T_9 is defined as 2, i.e., $W(P_3, T_9) = 2$. While for the other directed arcs, we define their weights equal to 1 in this model, that is $W(P_1, T_1) = W(T_1, P_1) = \cdots = W(T_8, P_9) = 1$, which means only one token can be transferred via each directed arc among them at each time.

5) INITIAL MARKING

In the initial state of the model, only one token is stored in place P_1, which indicates the original DNA strand. Besides, the other places do not have any tokens at the beginning.

TABLE 1. Color sets definition and symbol expression in CPN model of protein synthesis.

Name	Symbolic notation	Definition
Base_DNA	$X_{DNA} = \{X_i \mid X_i \in \{A,C,G,T\}\}$	X_i represents the different bases on a DNA strand.
Base_mRNA	$X_{mRNA} = \{X_j \mid X_j \in \{A,C,G,U\}\}$	X_i denotes the different bases on a mRNA strand.
Codon	$\{X_iX_{i+1}X_{i+2}\}$	$\{X_iX_{i+1}X_{i+2}\}$ denotes a codon that encoded an amino acid.
Position	i or $j \in INT$	$\{X_i, i\}$ is a compound color type *Base_DNA*×*Position*, where X_i indicates the base and i denotes its position on the DNA strand; Similarly, $\{X_iX_{i+1}X_{i+2}, j\}$ is a compound color type *Codon*×*Position*, where $X_iX_{i+1}X_{i+2}$ represents an amino acid and j is its position on the polypeptide chain.
DNA or *mRNA strand*	$\{X_1X_2\ldots X_P\}$	$\{X_1X_2\ldots X_P\}$, where $X_i \in X_{DNA}$, stands for a DNA strand; similarly, it also can express a mRNA strand, where $X_i \in X_{mRNA}$.
Null	$\{\ \}$	It stands for the token's color is null, meaning it is a colorless token.

TABLE 2. Places definition and symbol expression in CPN model of protein synthesis process.

Name	Definition
P_1	A token with color $\{X_1X_2\ldots X_P\}$ ($X_i \in X_{DNA}$) can be stored in P_1, indicating a given DNA strand.
P_2	P_2 is defined to store a colored token $\{X_i, i\}$, where X_i indicates the base and i denotes its position on the DNA strand.
P_3	P_3 can store no more than two tokens with the color defined in *Base_mRNA*, constituting a mRNA strand.
P_4	P_4 can store one token, expressing an entire mRNA strand.
P_5	P_5 is used to store one token, which stands for a fragment of mRNA strand because some bases before the start codon have been removed from the whole mRNA strand.
P_6	P_6 can store the bases before start codon on the mRNA strand, which means that this fragment cannot take part in the translation process.
P_7	P_7 can store a token with color $\{X_iX_{i+1}X_{i+2}, j\}$, where $X_iX_{i+1}X_{i+2}$ represents an amino acid and j is its position on the polypeptide chain.
P_8	P_8 is the place that can store the bases after the first stop codon of the mRNA strand, showing the end of the translation process.
P_9	P_9 can store a number of tokens having the color $\{X_iX_{i+1}X_{i+2}\}$, which stands for a group of amino acids along the polypeptide chain.

TABLE 3. Transitions definition and symbol expression in CPN model of protein synthesis.

Name	Input place	Output place	Definition
T_1	P_1	P_1, P_2	Transition T_1 is used to select the bases from P_1 one by one and then transfer it to P_2. Specifically, with the firing of T_1, it chooses the first base of the DNA sequence and outputs it to P_2 by a colored token $\{X_i, i\}$, where i stands for the base's ordinal number on the DNA strand. At the same time, the remaining bases move back to P_1. We define its guard function evaluates to true if the color of token stored in P_1 is not *Null.*
T_2	P_2	P_3	T_2 has the function of converting the bases on a DNA sequence and forming a mRNA strand, representing the process of transcription. The conversion rule is different depending on the type of given DNA strand is coding or template strand. When T_2 is fired, the converted base will be transmitted to P_3 by a colored token. The guard function of T_2 always evaluates to true.
T_3	P_3	P_4	The role of T_3 is defined to determine the transcription process is finished, i.e., all bases on the DNA strand are already converted and formed a complete mRNA strand. When T_3 is fired, it will output the mRNA to P_4. The guard function of T_3 evaluates to true if the color of tokens in both P_1 and P_2 is *Null*.
T_4	P_4	P_5	T_4 is used to identify the start codon (*AUG*) of the mRNA strand. It will be fired on condition that the first three bases constitute the start codon. Then the mRNA strand is transfered to P_5; If the condition is not satisfied, T_4 cannot be fired. The guard function of T_4 evaluates to true if P_4 contains a colored token while the first three bases can compose the start codon.
T_5	P_4	P_4, P_6	The function of T_5 is to remove the first base from the mRNA strand represented by a colored token stored in P_4. With the firing of T_5, it delivers the first base of the mRNA sequence to P_6 via a token. At the same time, the other bases within the mRNA strand return to P_4. Its guard function evaluates to true on condition that the color of token stored in P_4 is not *Null*.
T_6	P_5	P_5, P_7	T_6 is defined to determine any stop codon appears on the mRNA strand. T_6 will be fired if the first three bases cannot form one of the stop codons. As T_6 is firing, it transmits the first three bases to P_7 by a colored token $\{X_iX_{i+1}X_{i+2}, j\}$, and meanwhile the remaining bases move back to P_5. If it is the stop codon, T_6 cannot be fired. The guard function of T_6 evaluates to true under the following two conditions: 1) P_6 has a colored token; 2) the first three bases cannot compose the stop codon.
T_7	P_5	P_8	T_7 is used to separate the stop codon together with the following bases from the mRNA sequence that stored in P_5. As T_7 is firing, it delivers a segment of mRNA sequence to P_8 by a colored token. The guard function of T_7 evaluates to true if the input place P_5 stores a colored token and the first three bases can form any stop codon.
T_8	P_7	P_9	The role of transitions T_8 is to assemble the produced amino acids together in sequence and make up an amino acid chain. If T_8 is fired, it will use a token to transmit the codon of amino acid to P_9 and lay it after the existing codons stored in P_9. The guard function of T_8 always evaluates to true.
T_9	P_3	P_3	The function of T_9 is to combine the bases into a mRNA strand. T_9 will be fired when two colored tokens appeared in P_3. With the firing of T_9, it will combine the colors of the two tokens by placing the second color following the original one, and then output a colored token back to P_3.

So the initial marking of all places in the model is $M_0 = \{1, 0, 0, 0, 0, 0, 0, 0, 0\}$.

6) PRIORITY OF TRANSITIONS

During the operation of CPN model, a situation may occur where the tokens stored in one place may make two or more transitions to be enabled and fired simultaneously. Therefore, to avoid the transitions conflict in this case, we need to define the firing conditions for these transitions to ensure which one can be fired, that is, we define the priority of the transition.

1) In the stages of initialization and transcription, if the transition T_1 and T_2 are enabled simultaneously, we

TABLE 4. State changes of the protein synthesis CPN model in example 3.1.

Step	Distribution of tokens	Transition Enabled	Fired (in sequence)
1	$P_1:\{GATGCTTAAACAGTAAG\}$	T_1	T_1
2	$P_1:\{ATGCTTAAACAGTAAG\}$; $P_2:\{G,1\}$	T_1,T_2	T_2,T_1
3	$P_1:\{TGCTTAAACAGTAAG\}$; $P_2:\{A,2\}$; $P_3:\{G\}$	T_1,T_2,T_3	T_2,T_1
4	$P_1:\{GCTTAAACAGTAAG\}$; $P_2:\{T,3\}$; $P_3:\{G\}\{A\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
5	$P_1:\{CTTAAACAGTAAG\}$; $P_2:\{G,4\}$; $P_3:\{GA\}\{U\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
6	$P_1:\{TTAAACAGTAAG\}$; $P_2:\{C,5\}$; $P_3:\{GAU\}\{G\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
7	$P_1:\{TAAACAGTAAG\}$; $P_2:\{T,6\}$; $P_3:\{GAUG\}\{C\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
8	$P_1:\{AAACAGTAAG\}$; $P_2:\{T,7\}$; $P_3:\{GAUGC\}\{U\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
9	$P_1:\{AACAGTAAG\}$; $P_2:\{A,8\}$; $P_3:\{GAUGCU\}\{U\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
10	$P_1:\{ACAGTAAG\}$; $P_2:\{A,9\}$; $P_3:\{GAUGCUU\}\{A\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
11	$P_1:\{CAGTAAG\}$; $P_2:\{A,10\}$; $P_3:\{GAUGCUUA\}\{A\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
12	$P_1:\{AGTAAG\}$; $P_2:\{C,11\}$; $P_3:\{GAUGCUUAA\}\{A\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
13	$P_1:\{GTAAG\}$; $P_2\{A,12\}$; $P_3:\{GAUGCUUAAA\}\{C\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
14	$P_1:\{TAAG\}$; $P_2:\{G,13\}$; $P_3:\{GAUGCUUAAAC\}\{A\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
15	$P_1:\{AAG\}$;$P_2:\{T,14\}$; $P_3:\{GAUGCUUAAACA\}\{G\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
16	$P_1:\{AG\}$; $P_2:\{A,15\}$; $P_3:\{GAUGCUUAAACAG\}\{U\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
17	$P_1:\{G\}$; $P_2:\{A,16\}$; $P_3:\{GAUGCUUAAACAGU\}\{A\}$	T_1,T_2,T_3,T_9	T_2,T_1,T_9
18	$P_1:\{\}$; $P_2:\{G,17\}$; $P_3:\{GAUGCUUAAACAGUA\}\{A\}$	T_2,T_3,T_9	T_2,T_9
19	$P_1:\{\}$; $P_3:\{GAUGCUUAAACAGUAA\}\{G\}$	T_3,T_9	T_9
20	$P_1:\{\}$; $P_3:\{GAUGCUUAAACAGUAAG\}$	T_3	T_3
21	$P_1:\{\}$; $P_4:\{GAUGCUUAAACAGUAAG\}$	T_4,T_5	T_5
22	$P_1:\{\}$; $P_4:\{AUGCUUAAACAGUAAG\}$; $P_6:\{G\}$	T_4,T_5	T_4
23	$P_1:\{\}$; $P_5:\{AUGCUUAAACAGUAAG\}$; $P_6:\{G\}$	T_6,T_7	T_6
24	P_1, P_6 as above; $P_5:\{CUUAAACAGUAAG\}$; $P_7:\{AUG,1\}$	T_6,T_7,T_8	T_8,T_6
25	P_1, P_6 as above; $P_5:\{AAACAGUAAG\}$; $P_7:\{CUU,2\}$; $P_9:\{AUG\}$	T_6,T_7,T_8	T_8,T_6
26	P_1, P_6 as above; $P_5:\{CAGUAAG\}$; $P_7:\{AAA,3\}$; $P_9:\{AUG\}\{CUU\}$;	T_6,T_7,T_8	T_8,T_6
27	P_1, P_6 as above; $P_5:\{UAAG\}$; $P_7:\{CAG,4\}$; $P_9:\{AUG\}\{CUU\}\{AAA\}$	T_6,T_7,T_8	T_7,T_8
28	$P_1:\{\}$; $P_6:\{G\}$; $P_8:\{UAAG\}$; $P_9:\{AUG\}\{CUU\}\{AAA\}\{CAG\}$		

define the priority rule of them as $T_2 > T_1$, which is to ensure the place P_2 can store only one token.

2) For transition T_3 and T_9, we define their priority rule as $T_9 > T_3$, which guarantees that the bases are listed sequentially in the process of transcription before forming the whole mRNA strand.
3) In the stage of identifying start codon, the priority rule of transitions T_4 and T_5 is defined as $T_4 > T_5$.
4) In the stage of identifying stop codons, the priority of T_6 and T_7 is $T_6 > T_7$.

C. EXAMPLE AND RESULT ANALYSIS

E. coli ribonuclease H gene is an extensively studied gene which is used to explore the relationships between structure and enzymatic activity, especially as model systems to study protein folding. To illustrate the effectiveness and accuracy of the proposed model, we select a segment of DNA sequence of the *E. coli* ribonuclease H gene [38] which has 760 base pairs as a representative biological example. To simplify the operation, here we select the first four codons beginning with the start codon *ATG* and then we fill the stop codon *TAA* to the DNA sequence. Meanwhile, we add one base before the start codon and after the stop codon, respectively, to demonstrate the model's ability to identify the start codon and stop codons.

Fig. 3 shows a segment of DNA coding strand of the *E. coli* ribonuclease H gene. The relative mRNA and amino acids sequence along the polypeptide chain are obtained.

1) EXAMPLE 3.1

Assume a fragment of DNA coding strand of *E. coli* ribonuclease H gene with 17 bases (as shown in Fig. 3). Use the proposed model to illustrate the entire process that the genetic information within DNA is transcribed to mRNA and then translated to an amino acid chain.

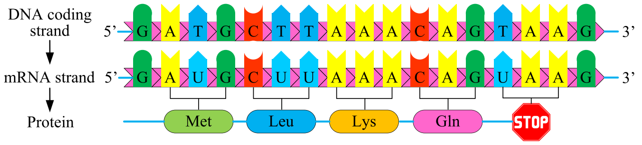

DNA coding strand
5'
G A T G C T T A A A C A G T A A G
3'
mRNA strand
5'
G A U G C U U A A A C A G U A A G
3'
Protein
Met
Leu
Lys
Gln
STOP

FIGURE 3. DNA coding strand, relative mRNA and polypeptide chain in Example 3.1.

The state changes of the protein synthesis CPN model in Example 3.1 are shown in Table 4. At each step, we can know the detailed information about the tokens distributed to all places in the model, as well as the transitions that could be enabled or fired during a certain stage.

2) RESULT ANALYSIS

As shown in Table 4, the stage of initialization and transcription are proceeding simultaneously, from step 1 to step 20. The stage of determining the start codon is processed in steps 21-22. Identifying the stop codon and translation are carried out at the same time from step 23 to 28. The mRNA chain is translated into a chain of amino acids via the gene reading frame until the stop codon *UAA* is identified.

Through the transfer of tokens in the model, the process of protein synthesis has been described intuitively. In the final state of the CPN model, there is one colorless token stored in P_1. The colored token $\{G\}$ stored in P_6 represents the base before the start codon on the mRNA chain. P_8 also has one token with color $\{UAAG\}$, and it is a segment of mRNA which contains the stop codon along with the bases behind it. If this mRNA segment still has the start and stop codon, we can let it reread in P_1 then the model will continue working and produce a new polypeptide chain. Four colored tokens are listed sequentially in P_9, including $\{AUG\}$, $\{CUU\}$, $\{AAA\}$, and $\{CAG\}$. They represent a group of amino acids along the polypeptide chain that transformed from the given DNA coding strand.

The operation and results of the model demonstrate that the proposed model can completely simulate the whole process of protein synthesis, while the operation procedure is also consistent with the actual biological process of genetic information transfer. The model provides a more intuitive characterization and description of this complex process, showing the feasibility and superiority of the Petri Nets applied in modeling and analysis the processes in biological systems.

IV. CPN MODEL OF MUTATION TYPE DETERMINATION

DNA mutation is defined as an alteration that may affect a single nucleotide pair or larger segments in the sequence of DNA. Changing nucleotide sequences most often results in the alterations in amino acids types or sequences along the polypeptide chain that may produce nonfunctioning proteins, which leads to the genetic variation of biological functions and the potential to develop the disease.

A. TYPES OF DNA MUTATION

DNA mutations can affect the gene expression in numerous ways resulting in varying influences on health, like neutral, beneficial or harmful effects. In view of the impact on protein sequence, DNA point mutations can be generally categorized into four types: missense mutation, nonsense mutation, frameshift mutation and synonymous mutation [39].

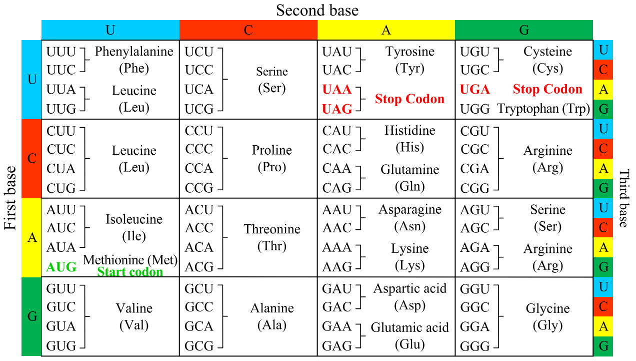
1) *Missense mutation.* A missense mutation is a genetic alteration in which a change or substitution occurs in a DNA base pair that alters the genetic code, resulting in one amino acid being substituted by another one at that position in the product protein.

2) *Nonsense mutation.* This type of mutation alters the nucleotide sequence in one DNA base pair so that a normal amino acid is substituted by a stop codon, which signals the termination of the process of translation and stops protein production. A nonsense mutation may produce a truncated or incomplete protein product, which quite often lacks the functionality of the normal protein.

3) *Frameshift mutation.* A frameshift mutation is regarded as an alteration that results from the addition or deletion of a single or several nucleotides on the DNA strand, causing a shift in the codon reading frame. Frameshift mutation usually yields a completely different amino acid chain from the original one and results in a nonfunctional protein.

4) *Synonymous mutation.* Although a base substitution occurs in the DNA strand, this mutation does not change the encoded amino acid in a produced protein. The reason is that an amino acid can be encoded by multiple genetic codons. Due to the redundancy in the genetic code, synonymous substitutions usually occur in the third position of a codon.

There are 20 different types of amino acids that can be combined to make a protein, each having its own physical and chemical properties. The genetic code is a set of instructions that link groups of nucleotides in a mRNA to amino acids in a protein. According to the genetic codes, 64 triplets are coded for the 20 amino acids and three stop codons, implying that some amino acids are encoded by multiple codons, called the redundancy of the genetic code. Synonymous codons of the 20 different types of amino acids are shown in Fig. 4.



Second base

First base	U		C		A		G		Third base
U	UUU UUC UUA UUG	Phenylalanine (Phe) Leucine (Leu)	UCU UCC UCA UCG	Serine (Ser)	UAU UAC UAA UAG	Tyrosine (Tyr) Stop Codon	UGU UGC UGA UGG	Cysteine (Cys) Stop Codon Tryptophan (Trp)	U C A G
C	CUU CUC CUA CUG	Leucine (Leu)	CCU CCC CCA CCG	Proline (Pro)	CAU CAC CAA CAG	Histidine (His) Glutamine (Gln)	CGU CGC CGA CGG	Arginine (Arg)	U C A G
A	AUU AUC AUA AUG	Isoleucine (Ile) Methionine (Met) Start codon	ACU ACC ACA ACG	Threonine (Thr)	AAU AAC AAA AAG	Asparagine (Asn) Lysine (Lys)	AGU AGC AGA AGG	Serine (Ser) Arginine (Arg)	U C A G
G	GUU GUC GUA GUG	Valine (Val)	GCU GCC GCA GCG	Alanine (Ala)	GAU GAC GAA GAG	Aspartic acid (Asp) Glutamic acid (Glu)	GGU GGC GGA GGG	Glycine (Gly)	U C A G

FIGURE 4. Synonymous codons of 20 amino acids.

B. CPN MODEL OF MUTATION TYPE DETERMINATION

In section III, we built a CPN model of the protein synthesis process that describes the flow of genetic information. Based on the proposed model, we further put forward a CPN model to analyze the normal and mutated DNA strands, and then determine the type of mutations, as shown in Fig. 5.

This model firstly carries out a base sequence alignment between the original and mutated DNA strands for determining the position of the mutations as well as the base mutation rate. After that, the model contrasts the codons of amino acids along the two synthesized polypeptide chains such that the mutation type can be identified by using the mutation type determining rules. Seven stages are defined to show the operation procedure of this model, including *initialization*, *transcription*, *base contrasting*, *identify start codon*, *identify stop codons*, *translation*, and *codons alignment and mutation type analysis*. Except for stage 3 and stage 7, the other five stages are the same as the definition stated in the model of protein synthesis. Now we focus on the two added stages.

Stage 3: *Bases contrasting*. The bases from the normal and mutated DNA strands are contrasted sequentially in this stage, which generates a base alignment judgment result. So the position of mutations is obtained and the base mutation rate between the two DNA strands can be calculated.

Stage 7: *Codons alignment and mutation type analysis*. Judgment result of amino acids is achieved by contrasting the amino acid codons along the polypeptide chains translated from the normal and mutated DNA strands. We can then analyze the mutation type according to the determining rules.

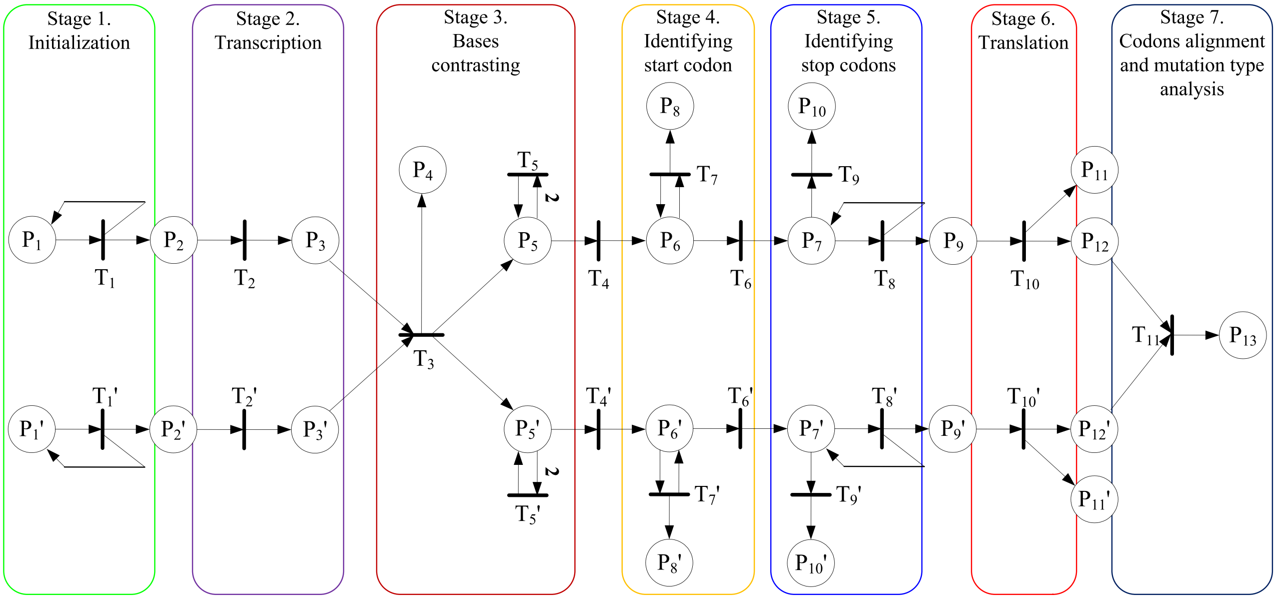


FIGURE 5. CPN model of DNA mutation type determination.

TABLE 5. Definition and symbol expression of two new color sets in DNA mutation type determination CPN model.

Name	Symbolic notation	Definition
Judgment result of mutated base	$\{s_1,s_2,\ldots,s_i,\ldots,s_P\}$, where $s_i \in \{0,1\}$	s_i denotes the base contrasting result and its position order along the two DNA sequences. Here s_i=0 means the two bases are the same, while s_i=1 represents the bases are different and may result in a mutation.
Judgment result of amino acid	$\{X_iX_{i+1}X_{i+2}, k\}$, where $k \in \{0,1\}$	It is a compound color type. $\{X_iX_{i+1}X_{i+2}, k\}$ indicates an amino acid with the judgment result of the two amino acids within the polypeptide chains translated from the normal and mutated DNA strand respectively. Here k=0 expresses the two amino acids are the same, and k=1 stands for they produce different amino acids.

TABLE 6. Places definition and symbol expression in DNA mutation type determination CPN model.

Name	Definition
P_1 (P_1')	Both P_1 and P_1' can store one colored token $\{X_1X_2\ldots X_P\}$, standing for the normal and mutated DNA strands are read in the model.
P_2 (P_2')	P_2 and P_2' can contain a token with color $\{X_i, i\}$, denoting nucleobase X_i and its position number in the DNA sequence.
P_3 (P_3')	Only one token can be stored in P_3 and P_3', where the token's color should be in the color set of *Base_mRNA*.
P_4	P_4 can store a token with the base position order and its comparison result along the DNA strand, which is obtained by contrasting each base on the normal and mutated DNA strands.
P_5 (P_5')	P_5 (P_5') can store no more than two tokens with the color defined in *Base_mRNA*.
P_6 (P_6')	P_6 (P_6') is defined to store one token, representing an entire mRNA strand.
P_7 (P_7')	P_7 (P_7') can store one token, denoting a mRNA fragment because some bases before the start codon have been removed from the mRNA strand.
P_8 (P_8')	P_8 and P_8' are used to store the bases before the start codon in the normal and mutated DNA strands, respectively.
P_9 (P_9')	P_9 (P_9') can store a token $\{X_iX_{i+1}X_{i+2}, j\}$, where $X_iX_{i+1}X_{i+2}$ indicates a codon and j is its sequence number along the produced polypeptide chain.
P_{10} (P_{10}')	P_{10} and P_{10}' are places to store the bases after the first stop codon of the two mRNA strands,
P_{11} (P_{11}')	P_{11} (P_{11}') is used to store some tokens having the color of *Codon*, which constitutes the product polypeptide chain with a group of amino acids.
P_{12} (P_{12}')	P_{12} and P_{12}' can store one token with a group of *Codon* colors like $\{X_iX_{i+1}X_{i+2}\}$, indicating all synonymous codons encoded for one amino acid.
P_{13}	P_{13} is defined to store a number of tokens with the color from the set *Judgment result of amino acid*, such as $\{X_iX_{i+1}X_{i+2}, k\}(k \in \{0,1\})$, which means an amino acid in the polypeptide that translated from the mutated DNA sequence and the judgment result obtained via codon alignment.

Next, we will give the detailed definition and symbol expression of the proposed CPN model of DNA mutation type determination and introduce the operation procedure of the model.

1) COLOR SETS

Based on the diverse states and features appearing in the above mentioned numerous stages within protein synthesis and mutation type determination, eight color sets are defined in the model, where *Base_DNA*, *Base_mRNA*, *Codon*, *Position*, *DNA or mRNA strand*, and *Null* are defined in Table 1. The definition of two new color sets, *Judgment result of mutated base and Judgment result of amino acid*, is demonstrated in Table 5.

2) PLACES

Accordingly, we define 24 places in the model to stand for the various states and stages. The detailed definition and symbol expression of each place are shown in Table 6.

TABLE 7. Transitions definition and symbol expression in DNA mutation type determination CPN model.

Name	Input place	Output place	Definition and function
T_1 (T_1')	P_1 (P_1')	P_1, P_2 (P_1', P_2')	T_1 (T_1') is defined to choose one base from P_1 (P_1') at each time. For example, as T_1 (T_1') is firing, it will deliver the first nucleobase of the DNA sequence to P_2 (P_2') via a token $\{X_i, i\}$ and output the remaining bases back to P_1 (P_1'). The guard function of T_1 (T_1') evaluates to true on condition of the place P_1 (P_1') contains a colored token.
T_2 (T_2')	P_2 (P_2')	P_3 (P_3')	T_2 (T_2') is designed for converting the nucleobases from DNA to mRNA strand that represents the process of transcription. Based on whether the type of reading DNA sequence is coding or template strand, the transition has different conversion rules. When T_2 (T_2') is firing, it will transmit the converted base to P_3 (P_3') via a token.
T_3	P_3, P_3'	P_4, P_5, P_5'	T_3 has the function that it contrasts two input nucleobases and provides a comparison result. When T_3 is fired, it will compare the bases $\{X_i\}$ in the normal DNA strand and $\{X_i'\}$ from the mutated one. If the two bases are the same, T_3 will deliver the same base to P_5 and P_5' by a token $\{X_i\}$ separately. At the same time, the judgment result $\{s_i=0\}$ will be transferred to P_4. If they are different bases, T_3 will output the base $\{X_i\}$ to P_5 and deliver $\{X_i'\}$ to P_5' via a token. Simultaneously, the judgment result $\{s_i=1\}$ will be transmitted to P_4, representing the order number of bases that have mutations. The guard function of T_3 evaluates to true if the bases $\{X_i\}$ and $\{X_i'\}$ have the same i.
T_4 (T_4')	P_5 (P_5')	P_6 (P_6')	T_4 (T_4') is defined to determine the transcription process is finished, i.e., all bases on the DNA strand are already contrasted and formed a complete mRNA strand. When T_4 (T_4') is fired, it will output the mRNA to P_6 (P_6'). The guard function of T_3 evaluates to true if the token color in P_1 is *Null* and there is no token stored in P_2 and P_3.
T_5 (T_5')	P_5 (P_5')	P_5 (P_5')	T_5 (T_5') is used to combine the bases into a mRNA strand. T_5 (T_5') will be fired when two colored tokens appeared in P_5 (P_5'). With the firing of T_5 (T_5'), it will combine the colors of the two tokens by placing the second color following the original one, and then output a colored token to P_5 (P_5').
T_6 (T_6')	P_6 (P_6')	P_7 (P_7')	The function of T_6 (T_6') is to distinguish the start codon in the mRNA strand. If the first three bases can form the start codon, T_6 (T_6') will be fired and output the mRNA strand to P_7 (P_7'); otherwise, T_6 (T_6') cannot be fired. The guard function of T_6 (T_6') evaluates to true if the following two conditions are satisfied: 1) there is no token in P_5 and P_5; 2) the first three bases can constitute the start codon.
T_7 (T_7')	P_6 (P_6')	P_6, P_8 (P_6', P_8')	T_7 (T_7') is designed to remove the first nucleobase from the mRNA strand stored in P_6 (P_6'). As T_7 (T_7') is firing, it delivers the first base to P_8 (P_8') via a token. Meanwhile, the remaining bases will return to P_6 (P_6'). Its guard function is evaluated to true if the color of token stored in P_6 (P_6') is not *Null*.
T_8 (T_8')	P_7 (P_7')	P_7, P_9 (P_7', P_9')	T_8 (T_8') has the function to identify any stop codon in the mRNA sequence. T_8 (T_8') will be fired on condition that the codon encoded by the first three bases is not a stop codon, meaning an amino acid in the growing polypeptide chain, Specifically, with the firing of T_8 (T_8'), it outputs the codon $\{X_iX_{i+1}X_{i+2}, j\}$ ($\{X_i'X_{i+1}'X_{i+2}', j\}$) to P_9 (P_9'). Simultaneously, the remaining bases move back to P_7 (P_7'). The guard function of T_8 (T_8') evaluates to true when P_7 (P_7') has the colored token and the first codon cannot constitute the stop codon.
T_9 (T_9')	P_7 (P_7')	P_{10} (P_{10}')	T_9 (T_9') plays the role of separating the stop codon and the following bases from the mRNA sequence stored in P_7 (P_7'). If T_9 (T_9') is fired, it delivers a segment of mRNA to P_{10} (P_{10}') by a colored token. Define the guard function of T_9 (T_9') evaluates to true if P_7 (P_7') has a colored token and the first three bases can compose any stop codon.
T_{10} (T_{10}')	P_9 (P_9')	P_{11}, P_{12} (P_{11}', P_{12}')	T_{10} (T_{10}') is defined to shift the amino acid to an output place and simultaneously generate all the synonymous codons that encode this amino acid. With the firing of T_{10} (T_{10}'), a colored token indicating an amino acid will be transmitted to P_{11} (P_{11}'), while all synonymous codons of the same amino acid will be transferred to P_{12} (P_{12}').
T_{11}	P_{12}, P_{12}'	P_{13}	T_{11} is used for contrasting the two input codons. As T_{11} is firing, it will execute a comparison operation which compares the token's color of *Codon* $\{X_iX_{i+1}X_{i+2}\}$ in P_{12} with the token $\{X_i'X_{i+1}'X_{i+2}'\}$ in P_{12}'. If they are the same amino acid, T_{11} will transfer the judgment result of amino acid by a token with color $\{X_iX_{i+1}X_{i+2}, 0\}$ to P_{13}; otherwise, a token $\{X_i'X_{i+1}'X_{i+2}', 1\}$ will be transmitted to P_{13}. Define the guard function of T_{11} evaluates to true if each input place (P_{12} and P_{12}') has at least one colored token.

3) TRANSITIONS

In the model, we design 20 transitions to represent the state changes or alignment analysis operations involved in the process of gene expression and mutation type determination. The definition and function of all transitions are defined in Table 7.

4) WEIGHT OF DIRECTED ARCS

Similar to the definition stated in the protein synthesis CPN model, we define the weight of two directed arcs as 2, i.e., $W(P_5,T_5) = 2$ and $W(P_5', T_5') = 2$. The weight of the other directed arcs in the model is still defined as 1, signifying when a transition is fired only one token can be transmitted through each input arc or output arc at a time.

5) INITIAL MARKING

We define two tokens stored in all places in the original state of the proposed model. One token is placed in P_1, which denotes the original DNA strand, and the other one is in P_1', representing the mutated DNA strand.

6) PRIORITY OF TRANSITIONS

Based on the transition priority rules in the protein synthesis CPN model, we still define the priority of transitions in this model.

1) For transition T_1 (T_1') and $T_2(T_2')$ used in the stages of initialization and transcription, we define their priority rules as $T_2 > T_1$ and $T_2' > T_1'$.

2) The priority rule of $T_4(T_4')$ and $T_5(T_5')$ are defined by $T_5 > T_4(T_5' > T_4')$, in order to make the bases list sequentially in the process of transcription and form the corresponding mRNA strand.

3) For the stage of identifying the start codon, the priority rules of transitions among T_6, T_7, T_6' and T_7', are that $T_6 > T_7$ and $T_6' > T_7'$.

4) Also in the stage of identifying stop codons, the priority of T_8, T_9, T_8' and T_9', is defined as $T_8 > T_9$ and $T_8' > T_9'$.

7) BASES MUTATION RATE

In genetics, the mutation rate can be expressed and defined in many different ways. Generally, the mutation rate is assumed as a measure that equivalent to the error rate of base changes in the nucleotide sequence of DNA in replication. On the basis of definition and procedure operation of DNA bases alignment in the model, it is possible to get a ratio of how many bases changed between the given normal and mutated DNA sequences by determining the token's color that stored in P_4. We define the bases mutation rates (R) as:

$$R = s/P,$$

where s is the number of the bases that have mutated between the two DNA sequences (i.e., the number of $s_i = 1$ appeared in the base judgment result), and P is the total number of bases along the whole DNA strand. The bases mutation rate is just a reflection of the percentage of bases changed among the contrasting original DNA strand and mutated DNA strand.

8) MUTATION TYPE DETERMINING RULES

Through model operation and alignment analysis of two DNA strands, the following mutation type determining rules are provided to identify and analyze the mutation type via judging the tokens and their colors stored in the final stage.

Mutation Type Determining Rules: Considering all tokens stored in places P_4, P_{12} and P_{13}in the final state of the model, we determine the mutation type by analyzing the judgment result of amino acid $\{X_iX_{i+1}X_{i+2},\ k\}$ ($k \in \{0, 1\}$) stored in P_{13} associated with the base judgment result $\{s_1, s_2, \ldots, s_i, \ldots, s_P\}(s_i \in \{0, 1\})$ stored in P_4. The determining rules are defined as follows.

1) If there is a base alteration happened in the DNA strand expressed by $s_i = 1$, and the judgment result of its corresponding amino acid is $k = 0$, which means the amino acid is the same one as respected in product polypeptide chain, this mutation type must be synonymous mutation.

2) If the mutation not only brings about the base change but also results in a different amino acid finally, i.e., the judgment result of the amino acid is $k = 1$, the type of this mutation should be missense mutation.

3) If the base judgment result $s_i = 1$ and its corresponding amino acid judgment result $k = 1$ appear continuously, indicating that the base variation caused a number of bases and amino acid are changed subsequently, the mutation is regarded as frameshift mutation.

4) If P_{12} still has colored tokens in the final stage, which signifies the translation process on the mutated DNA strand has been forced to terminate prematurely, so the mutation type is nonsense mutation.

The following procedure of mutation type determination is proposed to complete this process.

Here we would like to state the feasibility of mutation type determining rules, especially in determining the missense mutation and frameshift mutation. As is known, in nature the mutation rarely occurs and the mutation rate is actually very low, though a great number of mutations have been found by researchers. If the base and codon judgment results $s_i = 1$ and $k = 1$ appear separately or discontinuously, we can decide the mutation is missense mutation. Otherwise, if the base judgment result $s_i = 1$ and its related codon judgment result $k = 1$ are emerged continuously in DNA sequence and amino acids chain respectively, implying that the base insertion or deletion causes a great change in the DNA sequence and the reading frame shifts during the translation process, so we can determinate the type should be frameshift mutation. This determinant is based on the fact that several missense mutations occurred continuously at the same time is very rare.

```
Procedure 1 Mutation Type Determination
(1) Input: given normal DNA strand and the mutated one
(2) Output: mutation type of the occurred base changes
(3) Model operation: the input DNA strands are read and then the model is operated step by step until we have the final state of each place of the model.
(4) Mutation type determining: consider the token stored in $P_4$, like $\{s_1, s_2, \ldots, s_i, \ldots, s_P\}(s_i \in \{0, 1\})$, that shows which base has been mutated and may result in a mutation;
    For $s_i = 1, i \in \{1, 2, \ldots, P\}$, do
      Check the tokens stored in place $P_{13}$. Determine the mutation type by judging the token color $\{X_iX_{i+1}X_{i+2}, k\}$ where $k \in \{0, 1\}$, denoting the corresponding amino acid codon $X_iX_{i+1}X_{i+2}$ and the judgment result $k$.
      (4a) If $k = 0$, then it is synonymous mutation;
          End if
      (4b) If $k = 1$, then it is missense mutation;
          End if
    End For
    If $s_i = 1$ and $k = 1$ continuously appear, then the missense mutations identified in (4b) are further determined as frameshift mutation;
    End if
    If $P_{12}$ still has tokens, then it is nonsense mutation;
    End if
End Procedure
```

C. EXAMPLE AND RESULT ANALYSIS

It is an important step in model verification to validate and illustrate the rationality, effectiveness, and accuracy of the proposed CPN model through the actual mutation examples. Therefore, it is necessary to find some representative biological examples of DNA mutations in nature. In order to better test the function of the model, in the paper we select some DNA mutation examples to constitute persuasive and confirmatory test examples.

1) BIOLOGICAL EXAMPLES OF MUTATIONS

The Human Gene Mutation Database (HGMD) builds a comprehensive collection of published germline mutations in nuclear genes, which underlies the close relationship between mutations and human inherited disease. The database stored more than 203,000 different gene mutations identified in over 8,000 genes until March 2017 [40]. Missense mutations account for 45.3% of the total number of mutations and

52.5% of nearly 9,800 human inherited diseases together with disease-associated/functional polymorphisms that are cataloged in the database. For nonsense mutations, the percentage is about 11.0% of total mutations and this type of mutation accounts for 3.4% of human inherited diseases [41].

Sickle cell anemia is one of the most common inherited red blood cell disorders. This disease is caused by a missense mutation in codon 6 of the b-globin gene leading to an amino acid substitution that *Glutamic acid* (*GAG*) is replaced by *Valine* (*GTG*) [42]. As reported in HGMD, the numbers of nonsense mutations resulting in stop codons *TGA* (38.5%) or *TAG* (40.4%) are approximately equal, which is nearly twice of the instances of mutation to *TAA* (21.1%). Statistics show that the substitutions *CGA* (*Arginine*) → *TGA* and *CAG* (*Glutamine*) → *TAG* are two of the most frequent nonsense mutations in the HGMD. *Beta thalassemia* is a blood disorder that reduces the production of hemoglobin, mainly resulting from the introduction of premature termination codons [43]. A nonsense mutation has been first discovered and extensively studied is the mutation occurred at the codon 39 ($CAG \rightarrow TAG$) of the β-globin gene [44].

Frameshift mutations are generally known as one of the most deleterious changes to the coding sequence of a protein. They are extremely likely to cause large-scale changes in the length of the polypeptide and chemical composition, which may produce a non-functional protein that often disrupts the biochemical processes of a cell. *Smith-Magenis syndrome* is a complex disorder affecting facial features, intellectual disability, difficulty sleeping, and other behavioral problems [45]. One patient with this disease was reported to have a frameshift mutation in the *retinoic acid induced-1* (*RAI1*) gene [46], resulting from the deletion of the base C within the codon *CAG* that changes the gene's reading frame.

Synonymous mutations are regarded as the genetic changes that indirectly alter the sequence of amino acids on the encoded protein, but there is evidence that some of them still frequently contribute to human cancer or diseases [47]. *Cervical and vulvar cancer* is reported as one of the human diseases sometimes caused by a synonymous mutation. One inductive reason for this disease is that a codon *CTG* located in gene *Interleukin-2* (*IL2*) is changed to *CTT*, though these two codons stand for the same amino acid *Leucine* [48].

Although a large number of mutations have been discovered and studied at the nucleotide sequence level, the mutation rate in nature is usually very low. Due to the many harmful effects that can be caused by mutations, biological systems exhibit an extraordinary ability to keep the number of genetic variations at extreme low level. According to whole genome sequencing data, the human genome mutation rate is similarly estimated to be 1.1×10^{-8} per site per generation [49]. Therefore, it is difficult to give a biological example that includes these four kinds of mutations simultaneously in one DNA strand. Without loss of generality, in the paper we assume two concrete examples with above representative mutational diseases in DNA strands. In example 4.1, there are three mutations occurred in a DNA coding strand, containing a missense mutation (*sickle cell anemia* [42], *GAG* mutates to *GTG*), a synonymous mutation (*Cervical and vulvar cancer* [48], *CTG* changes to *CTT*), and a nonsense mutation (*Beta thalassemia* [43], *CAG* to *TAG*). Example 4.2 is one of the causes of *Smith-Magenis syndrome* that has a frameshift mutation in the *RAI1* gene [46] with the deletion of the base C within the codon *CAG*. Therefore, we assume a base C is deleted in the mutated DNA strand which reduces the length of the DNA strand of the *RAI1* gene and also contains the start codon as well as the stop codon to ensure the translation process is complete.

2) EXAMPLE 4.1

Assume two equal-length DNA sequences, including a coding DNA strand and a mutated one. Compared with the coding DNA strand, there are three alterations happened in the mutated DNA strand, where the 6th base mutated from $A$to T, 10th base from G to T, and 11th base from C to T, respectively. Use the proposed model and mutation type determining rules to identify the type of these three mutations. The coding and the mutated DNA strands as well as their produced proteins are shown in Fig. 6. The red arrows mark the changes of three bases. Furthermore, Fig. 6 also displays the gene reading frame and the encoded amino acids with the start and stop codon.

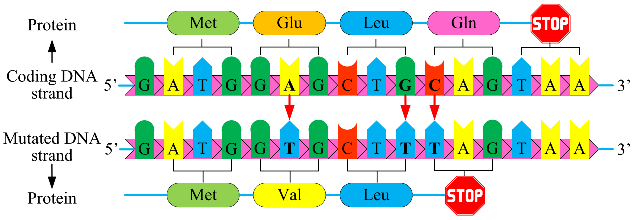


FIGURE 6. DNA coding strand, mutated DNA strand, and corresponding polypeptide chain in Example 4.1.

Table 8 displays the state changes and the token distributions in all places of the CPN model for contrasting the two DNA strands shown in Example 4.1. After 28 steps processing and analysis, we have the following results in the final state of the model. Each place of P_1 and P'_1 stores one colorless token, indicating all bases have been processed in the model. P_4 owns one token with color $\{0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0\}$, which stands for the base contrasting result and its position (where the base judgment result 1 appears) along the two DNA sequences. According to the definition of base mutation rate, it can be calculated by $R = 3/16 = 18.75\%$. There is one colored token stored in P_8 and P'_8, which indicates the bases ahead of the start codon on the coding and mutated DNA strands, respectively. The tokens in P_{10} and P'_{10} stand for the stop codon together with the following bases in the coding and mutated DNA sequences, respectively. In P_{11} and P'_{11}, a group of colored tokens represent the amino acids that are sequenced in the polypeptide chains transformed from the given coding and mutated DNA strands, respectively. Since the 11th base C is

TABLE 8. State changes of the mutation type determination CPN model in Example 4.1.

Step	Distribution of tokens	Transition Enabled	Fired (in sequence)
1	P_1:{*GATGGAGCTGCAGTAA*}; P_1':{*GATGGTGCTTTAGTAA*}	T_1,T_1'	T_1,T_1'
2	P_1:{*ATGGAGCTGCAGTAA*}; P_1':{*ATGGTGCTTTAGTAA*}; $P_2=P_2'$:{*G*,1};	T_1,T_2,T_1',T_2'	T_2,T_2',T_1,T_1'
3	P_1:{*TGGAGCTGCAGTAA*}; P_1':{*TGGTGCTTTAGTAA*}; $P_2=P_2'$:{*A*,2}; $P_3=P_3'$:{*G*,1}	T_1,T_2,T_1',T_2',T_3	T_2,T_2',T_1,T_1',T_3
4	P_1:{*GGAGCTGCAGTAA*}; P_1':{*GGTGCTTTAGTAA*}; $P_2=P_2'$:{*T*,3}; $P_3=P_3'$:{*A*,2}; P_4:{0}; $P_5=P_5'$:{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	T_2,T_2',T_1,T_1',T_3
5	P_1:{*GAGCTGCAGTAA*}; P_1':{*GTGCTTTAGTAA*}; $P_2=P_2'$:{*G*,4}; $P_3=P_3'$:{*U*,3}; P_4:{0,0}; $P_5=P_5'$:{*G*}{*A*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
6	P_1:{*AGCTGCAGTAA*}; P_1':{*TGCTTTAGTAA*}; $P_2=P_2'$:{*G*,5}; $P_3=P_3'$:{*G*,4}; P_4:{0,0,0}; $P_5=P_5'$:{*GA*}{*U*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
7	P_1:{*GCTGCAGTAA*}; P_1':{*GCTTTAGTAA*}; P_2:{*A*,6}; P_2':{*T*,6}; $P_3=P_3'$:{*G*,5}; P_4:{0,0,0,0}; $P_5=P_5'$:{*GAU*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
8	P_1:{*CTGCAGTAA*}; P_1':{*CTTTAGTAA*}; $P_2=P_2'$:{*G*,7}; P_3:{*A*,6}; P_3':{*U*,6}; P_4:{0,0,0,0,0}; $P_5=P_5'$:{*GAUG*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
9	P_1:{*TGCAGTAA*}; P_1':{*TTTAGTAA*}; $P_2=P_2'$:{*C*,8}; $P_3=P_3'$:{*G*,7}; P_4:{0,0,0,0,0,1}; P_5:{*GAUGG*}{*A*}; P_5':{*GAUGG*}{*U*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
10	P_1:{*GCAGTAA*}; P_1':{*TTAGTAA*}; $P_2=P_2'$:{*T*,9}; $P_3=P_3'$:{*C*,8}; P_4:{0,0,0,0,0,1,0}; P_5:{*GAUGGA*}{*G*}; P_5':{*GAUGGU*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
11	P_1:{*CAGTAA*}; P_1':{*TAGTAA*}; P_2:{*G*,10}; P_2':{*T*,10}; $P_3=P_3'$:{*U*,9}; P_4:{0,0,0,0,0,1,0,0}; P_5:{*GAUGGAG*}{*C*}; P_5':{*GAUGGUG*}{*C*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
12	P_1:{*AGTAA*}; P_1':{*AGTAA*}; P_2:{*C*,11}; P_2':{*T*,11}; P_3:{*G*,10}; P_3':{*A*,10}; P_4:{0,0,0,0,0,1,0,0,0}; P_5:{*GAUGGAGC*}{*U*}; P_5':{*GAUGGUGC*}{*U*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
13	P_1:{*GTAA*}; P_1':{*GTAA*}; $P_2=P_2'$:{*A*,12}; P_3:{*C*,11}; P_3':{*U*,11}; P_4:{0,0,0,0,0,1,0,0,0,1}; P_5:{*GAUGGAGCU*}{*G*}; P_5':{*GAUGGUGCU*}{*U*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
14	P_1:{*TAA*}; P_1':{*TAA*}; $P_2=P_2'$:{*G*,13}; P_3:{*A*,12}; P_3':{*A*,12}; P_4:{0,0,0,0,0,1,0,0,0,1,1}; P_5:{*GAUGGAGCUG*}{*C*}; P_5':{*GAUGGUGCUU*}{*U*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
15	P_1:{*AA*}; P_1':{*AA*}; $P_2=P_2'$:{*T*,14}; $P_3=P_3'$:{*G*,13}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0}; P_5:{*GAUGGAGCUGC*}{*A*}; P_5':{*GAUGGUGCUUU*}{*A*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
16	P_1:{*A*}; P_1':{*A*}; $P_2=P_2'$:{*A*,15}; $P_3=P_3'$:{*U*,14}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0}; P_5:{*GAUGGAGCUGCA*}{*G*}; P_5':{*GAUGGUGCUUUA*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
17	P_1:{}; P_1':{}; $P_2=P_2'$:{*A*,16}; $P_3=P_3'$:{*A*,15}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0,0}; P_5:{*GAUGGAGCUGCAG*}{*U*}; P_5':{*GAUGGUGCUUUAG*}{*U*}	$T_2,T_2',T_3,T_4,T_5,T_4',T_5'$	T_2,T_2',T_5,T_5',T_3
18	$P_1=P_1'$:{}; $P_3=P_3'$:{*A*,16}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0,0,0}; P_5:{*GAUGGAGCUGCAGU*}{*A*}; P_5':{*GAUGGUGCUUUAGU*}{*A*}	T_3,T_4,T_5,T_4',T_5'	T_5,T_5',T_3
19	$P_1=P_1'$:{}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0}; P_5:{*GAUGGAGCUGCAGUA*}{*A*}; P_5':{*GAUGGUGCUUUAGUA*}{*A*}	T_4,T_5,T_4',T_5'	T_5,T_5'
20	$P_1=P_1'$:{}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0}; P_5:{*AUGGAGCUGCAGUAA*}; P_5':{*AUGGUGCUUUAGUAA*}	T_4,T_4'	T_4,T_4'
21	$P_1=P_1'$:{}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0}; P_6:{*AUGGAGCUGCAGUAA*}; P_6':{*AUGGUGCUUUAGUAA*}	T_6,T_7,T_6',T_7'	T_7,T_7'
22	$P_1=P_1'$:{}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0}; P_6:{*AUGGAGCUGCAGUAA*}; P_6':{*AUGGUGCUUUAGUAA*}; $P_8=P_8'$:{*G*}	T_6,T_7,T_6',T_7'	T_6,T_6'
23	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*AUGGAGCUGCAGUAA*}; P_7':{*AUGGUGCUUUAGUAA*}	T_8,T_9,T_8',T_9'	T_8,T_8'
24	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*GAGCUGCAGUAA*}; P_7':{*GUGCUUUAGUAA*}; $P_9=P_9'$:{*AUG*,1}	$T_8,T_9,T_8',T_9',T_{10},T_{10}'$	T_{10},T_{10}',T_8,T_8'
25	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*CUGCAGUAA*}; P_7':{*CUUUAGUAA*}; P_9:{*GAG*,2}; P_9':{*GUG*,2}; $P_{11}=P_{11}'$:{*AUG*}; $P_{12}=P_{12}'$:{*AUG*}	$T_8,T_9,T_8',T_9',T_{10},T_{10}',$ T_{11}	$T_{11},T_{10},T_{10}',T_8,T_8'$
26	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*CAGUAA*}; P_7':{*UAGUAA*}; P_9:{*CUG*,3}; P_9':{*CUU*,3}; P_{11}:{*AUG*}{*GAG*}; P_{11}':{*AUG*}{*GUG*}; P_{12}:{*GAG*,*GAA*}; P_{12}':{*GUG*,*GUA*,*GUC*,*GUU*}; P_{13}:{*AUG*,0}	$T_8,T_9,T_8',T_9',T_{10},T_{10}',$ T_{11}	$T_{11},T_{10},T_{10}',T_8,T_9'$
27	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*UAA*}; P_9:{*CAG*,4}; P_{10}':{*UAGUAA*}; P_{11}:{*AUG*}{*GAG*}{*CUG*}; P_{11}':{*AUG*}{*GUG*}{*CUU*}; $P_{12}=P_{12}'$:{*CUA*,*CUC*,*CUG*,*CUU*,*UUA*,*UUG*}; P_{13}:{*AUG*,0}{*GUG*,1}	T_8,T_9,T_{10},T_{11}	T_{11},T_{10},T_9
28	$P_1=P_1'$:{}; P_4:{0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0}; $P_8=P_8'$:{*G*}; P_{10}:{*UAA*}; P_{10}':{*UAGUAA*}; P_{11}:{*AUG*}{*GAG*}{*CUG*}{*CAG*}; P_{11}':{*AUG*}{*GUG*}{*CUU*}; P_{12}:{*CAG*}; P_{13}:{*AUG*,0}{*GUG*,1}{*CUU*,0}		

mutated to *T* that constitutes a stop codon *TAG*, this substitution causes the translation process in the mutated DNA strand to terminate in advance. It is for this reason that P_{12} still has one token {*CAG*}, implying that in the mutated polypeptide chain the corresponding amino acids are missing after translation. Three colored tokens are orderly stored in P_{13}, containing $\{AUG, 0\}$, $\{GUG, 1\}$, and $\{CUU, 0\}$. These tokens not only stand for the amino acids chain that transformed from the mutated DNA strand but also show the codon judgment results via contrasting the codons along the two produced amino acids chain.

Finally, we can recognize the mutation type of each base change ocurred according to the procedure of mutation type determining. The mutations that happened on the 6th and 10th bases are missense mutation and synonymous mutation, respectively. As a result of P_{12} having one token, indicating that the translation process in the mutated DNA strand is terminated in advance, so the type of mutation occurred in the 11th base is nonsense mutation.

3) EXAMPLE 4.2

As shown in Fig. 7, assume there are two DNA sequences and both of them have 16 bases. Compared to the coding DNA strand, the 5th base *C* is removed from the mutated DNA strand. It can be easily found that both the bases on the mutated DNA strand and the amino acid sequence along the produced polypeptide chains are greatly changed due to the base deletion.

TABLE 9. State changes of the mutation type determination CPN model in Example 4.2.

Step	Distribution of tokens	Transition Enabled	Fired (in sequence)
1	P_1:{*CATGCAGGGACAGTAA*}; P_1':{*CATGAGGGACAGTAAG*}	T_1,T_1'	T_1,T_1'
2	P_1:{*ATGCAGGGACAGTAA*}; P_1':{*ATGAGGGACAGTAAG*}; $P_2=P_2'$:{*C*,1};	T_1,T_2,T_1',T_2'	T_2,T_2',T_1,T_1'
3	P_1:{*TGCAGGGACAGTAA*}; P_1':{*TGAGGGACAGTAAG*}; $P_2=P_2'$:{*A*,2}; $P_3=P_3'$:{*C*,1}	T_1,T_2,T_1',T_2',T_3	T_2,T_2',T_1,T_1',T_3
4	P_1:{*GCAGGGACAGTAA*}; P_1':{*GAGGGACAGTAAG*}; $P_2=P_2'$:{*T*,3}; $P_3=P_3'$:{*A*,2}; P_4:{0}; $P_5=P_5'$:{*C*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	T_2,T_2',T_1,T_1',T_3
5	P_1:{*CAGGGACAGTAA*}; P_1':{*AGGGACAGTAAG*}; $P_2=P_2'$:{*G*,4}; $P_3=P_3'$:{*U*,3}; P_4:{0,0}; $P_5=P_5'$:{*C*}{*A*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
6	P_1:{*AGGGACAGTAA*}; P_1':{*GGGACAGTAAG*}; P_2:{*C*,5}; P_2':{*A*,5}; $P_3=P_3'$:{*G*,4}; P_4:{0,0,0}; $P_5=P_5'$:{*CA*}{*U*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
7	P_1:{*GGGACAGTAA*}; P_1':{*GGACAGTAAG*}; P_2:{*A*,6}; P_2':{*G*,6}; P_3:{*C*,5}; P_3':{*A*,5}; P_4:{0,0,0,0}; $P_5=P_5'$:{*CAU*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
8	P_1:{*GGACAGTAA*}; P_1':{*GACAGTAAG*}; $P_2=P_2'$:{*G*,7}; P_3:{*A*,6}; P_3':{*G*,6}; P_4:{0,0,0,0,1}; P_5:{*CAUG*}{*C*}; P_5':{*CAUG*}{*A*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
9	P_1:{*GACAGTAA*}; P_1':{*ACAGGTAAG*}; $P_2=P_2'$:{*G*,8}; $P_3=P_3'$:{*G*,7}; P_4:{0,0,0,0,1,1}; P_5:{*CAUGC*}{*A*}; P_5':{*CAUGA*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
10	P_1:{*ACAGTAA*}; P_1':{*CAGTAAG*}; P_2:{*G*,9}; P_2':{*A*,9}; $P_3=P_3'$:{*G*,8}; P_4:{0,0,0,0,1,1,0}; P_5:{*CAUGCA*}{*G*}; P_5':{*CAUGAG*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
11	P_1:{*CAGTAA*}; P_1':{*AGTAAG*}; P_2:{*A*,10}; P_2':{*C*,10}; P_3:{*G*,9}; P_3':{*A*,9}; P_4:{0,0,0,0,1,1,0,0}; P_5:{*CAUGCAG*}{*G*}; P_5':{*CAUGAGG*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
12	P_1:{*AGTAA*}; P_1':{*GTAAG*}; P_2:{*C*,11}; P_2':{*A*,11}; P_3:{*A*,10}; P_3':{*C*,10}; P_4:{0,0,0,0,1,1,0,0,1}; P_5:{*CAUGCAGG*}{*G*}; P_5':{*CAUGAGGG*}{*A*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
13	P_1:{*GTAA*}; P_1':{*TAAG*}; P_2:{*A*,12}; P_2':{*G*,12}; P_3{*C*,11}; P_3':{*A*,11}; P_4:{0,0,0,0,1,1,0,0,1,1}; P_5:{*CAUGCAGGG*}{*A*}; P_5':{*CAUGAGGGA*}{*C*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
14	P_1:{*TAA*}; P_1':{*AAG*}; P_2:{*G*,13}; P_2':{*T*,13}; P_3:{*A*,12}; P_3':{*G*,12}; P_4:{0,0,0,0,1,1,0,0,1,1,1}; P_5:{*CAUGCAGGGA*}{*C*}; P_5':{*CAUGAGGGAC*}{*A*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
15	P_1:{*AA*}; P_1':{*AG*}; P_2:{*T*,14}; P_2':{*A*,14}; P_3:{*G*,13}; P_3':{*U*,13}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1}; P_5:{*CAUGCAGGGAC*}{*A*}; P_5':{*CAUGAGGGACA*}{*G*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
16	P_1:{*A*}; P_1':{*G*}; $P_2=P_2'$:{*A*,15}; P_3:{*U*,14}; P_3':{*A*,14}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1,1}; P_5:{*CAUGCAGGGACA*}{*G*}; P_5':{*CAUGAGGGACAG*}{*U*}	$T_1,T_2,T_1',T_2',T_3,T_4,T_5,$ T_4',T_5'	$T_2,T_2',T_1,T_1',T_5,T_5',T_3$
17	P_1:{}; P_1':{}; P_2:{*A*,16}; P_2':{*G*,16}; $P_3=P_3'$:{*A*,15}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1,1,1}; P_5:{*CAUGCAGGGACAG*}{*U*}; P_5':{*CAUGAGGGACAGU*}{*A*}	$T_2,T_2',T_3,T_4,T_5,T_4',T_5'$	T_2,T_2',T_5,T_5',T_3
18	$P_1=P_1'$:{}; P_3:{*A*,16}; P_3':{*G*,16}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1,1,1,0}; P_5:{*CAUGCAGGGACAGU*}{*A*}; P_5':{*CAUGAGGGACAGUA*}{*A*}	T_3,T_4,T_5,T_4',T_5'	T_5,T_5',T_3
19	$P_1=P_1'$:{}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1,1,1,0,1}; P_5:{*CAUGCAGGGACAGUA*}{*A*}; P_5':{*CAUGAGGGACAGUAA*}{*G*}	T_4,T_5,T_4',T_5'	T_5,T_5'
20	$P_1=P_1'$:{}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1,1,1,0,1}; P_5:{*CAUGCAGGGACAGUAA*}; P_5':{*CAUGAGGGACAGUAAG*}	T_4,T_4'	T_4,T_4'
21	$P_1=P_1'$:{}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1,1,1,0,1}; P_6:{*CAUGCAGGGACAGUAA*}; P_6':{*CAUGAGGGACAGUAAG*}	T_6,T_7,T_6',T_7'	T_7,T_7'
22	P_1, P_1', P_4 as above; P_6:{*AUGCAGGGACAGUAA*}; P_6':{*AUGAGGGACAGUAAG*}; $P_8=P_8'$:{*C*}	T_6,T_7,T_6',T_7'	T_6,T_6'
23	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*AUGCAGGGACAGUAA*}; P_7':{*AUGAGGGACAGUAAG*}	T_8,T_9,T_8',T_9'	T_8,T_8'
24	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*CAGGGACAGUAA*}; P_7':{*AGGGACAGUAAG*}; $P_9=P_9'$:{*AUG*,1}	$T_8,T_9,T_8',T_9',T_{10},T_{10}'$	T_{10},T_{10}',T_8,T_8'
25	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*GGACAGUAA*}; P_7':{*GACAGUAAG*}; P_9:{*CAG*,2}; P_9':{*AGG*,2}; $P_{11}=P_{11}'$:{*AUG*}; $P_{12}=P_{12}'$:{*AUG*}	$T_8,T_9,T_8',T_9',T_{10},T_{10}',$ T_{11}	$T_{11},T_{10},T_{10}',T_8,T_8'$
26	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*CAGUAA*}; P_7':{*AGUAAG*}; P_9:{*GGA*,3}; P_9':{*GAC*,3}; P_{11}:{*AUG*}{*CAG*}; P_{11}':{*AUG*}{*AGG*}; P_{12}:{*CAG*,*GAA*}; P_{12}':{*AGG*,*AGA*,*CGA*,*CGC*,*CGG*,*CGU*}; P_{13}:{*AUG*,0}	$T_8,T_9,T_8',T_9',T_{10},T_{10}',$ T_{11}	$T_{11},T_{10},T_{10}',T_8,T_8'$
27	P_1, P_1', P_4, P_8, P_8' as above; P_7:{*UAA*}; P_7':{*AAG*}; P_9:{*UGG*,4}; P_9':{*GGU*,4}; P_{11}:{*AUG*}{*CAG*}{*GGA*}; P_{11}':{*AUG*}{*AGG*}{*GAC*}; P_{12}:{*GGA*,*GGC*,*GGG*,*GGU*}; P_{12}':{*GAC*,*GAU*}; P_{13}:{*AUG*,0}{*GGC*,1}	$T_8,T_9,T_8',T_9',T_{10},T_{10}',$ T_{11}	$T_{11},T_{10},T_{10}',T_9,T_8'$
28	P_1, P_1', P_4, P_8, P_8' as above; P_{10}:{*UAA*}; P_9':{*AAG*,5}; P_{11}:{*AUG*}{*CAG*}{*GGA*}{*CAG*}; P_{11}':{*AUG*}{*AGG*}{*GAC*}{*AGU*}; P_{12}:{*CAG*,*CAA*}; P_{12}':{*AGU*,*AGC*,*UCA*,*UCC*,*UCG*,*UCU*}; P_{13}:{*AUG*,0}{*AGG*,1}{*GAC*,1}	T_{10}',T_{11}	T_{11},T_{10}'
29	$P_1=P_1'$:{}; P_4:{0,0,0,0,1,1,0,0,1,1,1,1,1,1,0,1}; $P_8=P_8'$:{*C*}; P_{10}:{*UAA*}; P_{11}:{*AUG*}{*CAG*}{*GGA*}{*CAG*}; P_{11}':{*AUG*}{*AGG*}{*GAC*}{*AGU*}{*AAG*}; P_{12}':{*AAG*,*AAA*}; P_{13}:{*AUG*,0}{*AGG*,1}{*GAC*,1}{*AGU*,1}		

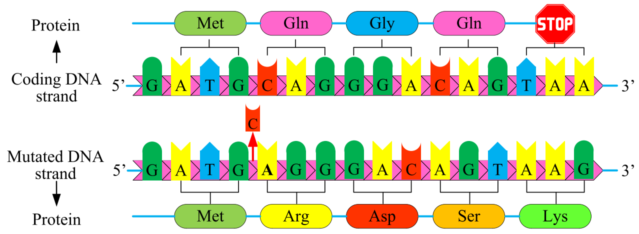


FIGURE 7. DNA coding strand, mutated DNA strand, and corresponding polypeptide chain in Example 4.2.

The state changes and alignment analysis operations of the model in Example 4.2 are demonstrated in Table 9. Besides, the distribution of tokens in all places and the transitions that could be enabled or fired in 29 steps are shown in the table. In the final state, both P_1 and P'_1 have one token with the color *Null*. Place P_4 stores one token with the color $\{0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1\}$, where 1 stands for the base is altered in that position, and 0 denotes the base keeps the same. We can obtain that there are 9 bases altered on the mutated DNA sequence, and calculate the base mutation rate by $R = 9/16 = 56.25\%$, denoting that these two DNA sequences have many base changes. The meaning and explanation of the tokens stored in P_8 (P'_8), P_{10} (P'_{10}) and P_{11}(P'_{11}) are similar to the analysis result of the previous Example 4.1.

As the 5th base *A* is deleted from the mutated DNA sequence, this mutation alters the gene read frame during the translation process, resulting in produce a group of different amino acids. There is a token with color $\{AAG, AAA\}$ stored in P'_{12}, indicating that translation is still going on because of no stop codon showing up. P_{13} owns four colored tokens, including $\{AUG, 0\}$, $\{AGG, 1\}$, $\{GAC, 1\}$, and $\{AGU, 1\}$. The base judgment result $s_i = 1$ and codon alignment result $k = 1$ appears continuously, indicating that this type of mutation gives rise to a group of amino acids (Met, Arg, Asp, Ser, and Lys) different from the normal ones (Met, Gln, Gly, and Gln). On the basis of above analysis, we can determinate the type of mutation is frameshift mutation by the mutation type determining rules.

V. CONCLUSION AND DISCUSSION

Protein synthesis is a fundamental biological process that characterizes the flow of genetic information from DNA to protein. To more intuitively display the inherent interactions among the various components, such as nucleobases, amino acids, and polypeptide chain, a CPN-based model was designed to describe this complex biological process. The modeling method not only provides an intuitive and vivid description of the detailed state changes in the process of protein synthesis, but also establishes a theoretical foundation for intuitively modeling and analyzing other essential biological processes.

Varying alternations in DNA sequences cause different types of mutations that lead to the genetic changes and the potential to cause the diseases. Identifying the type of the mutation is beneficial for analyzing whether it affects the structure or function of the produced protein. Based on the proposed model of protein synthesis, we designed a CPN model to judge the mutation's type by aligning and analyzing the differences between the normal and the mutated DNA strand together with the changes of the amino acids in the synthesized polypeptide chain. Some biological mutations examples illustrated the effectiveness and accuracy of the proposed model.

This work provides a useful model for the study of whether gene mutation changes the structure and function of the protein synthesis. An interesting work to be considered in the future is to extend the functions of the proposed model by adding some analysis and control operations. If some simple control approaches can be added to the model to make it as a controllable systematic model, a multi-step induction control model of gene mutation can be further established. It will provide new research ideas and model basis for studying the genetic stability and induction control strategies for gene mutations from the perspective of system science.

REFERENCES

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 6th ed. New York, NY, USA: Garland Science, 2014, pp. 1–10.

[2] C. Mulhardt, *Molecular Biology and Genomics*, 1st ed. San Diego, CA, USA: Academic, 2006.

[3] A. Isaev, *Introduction to Mathematical Methods in Bioinformatics*, 1st ed. Berlin, Germany: Springer-Verlag, 2006.

[4] M. Zhang, M. X. Cheng, and T. Tarn, "A mathematical formulation of DNA computation," *IEEE Trans. Nanobiosci.*, vol. 5, no. 1, pp. 32–40, Mar. 2006.

[5] R. Gao, J. Yu, M. Zhang, T.-J. Tarn, and J.-S. Li, "Systems theoretic analysis of the central dogma of molecular biology: Some recent results," *IEEE Trans. Nanobiosci.*, vol. 9, no. 1, pp. 59–70, Mar. 2010.

[6] R. Gao, W. Hu, and T.-J. Tarn, "The application of finite state machine in modeling and control of gene mutation process," *IEEE Trans. Nanobiosci.*, vol. 12, no. 4, pp. 265–274, Dec. 2013.

[7] B.-J. Yoon, "Hidden Markov models and their applications in biological sequence analysis," *Current Genomics*, vol. 10, no. 6, pp. 402–415, 2009.

[8] A. Löytynoja and M. C. Milinkovitch, "A hidden Markov model for progressive multiple alignment," *Bioinformatics*, vol. 19, no. 12, pp. 1505–1513, 2003.

[9] K.-J. Won, T. Hamelryck, A. Prügel-Bennett, and A. Krogh, "An evolutionary method for learning HMM structure: Prediction of protein secondary structure," *BMC Bioinform.*, vol. 8, p. 357, Sep. 2007.

[10] T. F. Oliver, B. Schmidt, Y. Jakop, and D. L. Maskell, "High speed biological sequence analysis with hidden Markov models on reconfigurable platforms," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 5, pp. 740–746, Sep. 2009.

[11] Y.-B. Zhao and J. Krishnan, "mRNA translation and protein synthesis: An analysis of different modelling methodologies and a new PBN based approach," *BMC Syst. Biol.*, vol. 8, p. 25, Feb. 2014.

[12] T. von der Haar, "Mathematical and computational modelling of ribosomal movement and protein synthesis: An overview," *Comput. Struct. Biotechnol. J.*, vol. 1, no. 1, p. e201204002, 2012.

[13] A. Ay and D. N. Arnosti, "Mathematical modeling of gene expression: A guide for the perplexed biologist," *Critical Rev. Biochem. Mol. Biol.*, vol. 46, no. 2, pp. 137–151, 2011.

[14] C. A. Petri, "Kommunikation mit automaten," Bonn: Institut für Instrumentelle Mathematik, Schriften des Instituts fur Instrumentelle Mathematik, New York, NY, USA, Griffiss Air Force Base, Tech. Rep. RADC-TR-65-377, 1966, vol. 1.

[15] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[16] J. Barjis and I. Barjis, "Formalization of the protein production by means of Petri nets," in *Proc. Int. Conf. Inf. Intell. Syst.*, Washington, DC, USA, Oct./Nov. 1999, pp. 4–9.

[17] J. Nummela and B. A. Julstrom, "Evolving Petri nets to represent metabolic pathways," in *Proc. 7th Annu. Conf. Genet. Evol. Comput.*, Washington, DC, USA, 2005, pp. 2133–2139.

[18] S. Grunwald, A. Speer, J. Ackermann, and I. Koch, "Petri net modelling of gene regulation of the Duchenne muscular dystrophy," *BioSystems*, vol. 92, no. 2, pp. 189–205, 2008.

[19] K. Jensen, "Coloured petri nets and the invariant-method," *Theor. Comput. Sci.*, vol. 14, no. 3, pp. 317–336, 1981.

[20] K. Jensen, "An introduction to the theoretical aspects of coloured Petri nets," in *A Decade of Concurrency Reflections and Perspectives* (Lecture Notes in Computer Science), vol. 803, J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, Eds. Heidelberg, Germany: Springer-Verlag, 1994, pp. 230–272.

[21] K. Jensen and L. M. Kristensen, *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Berlin, Germany: Springer, Jul. 2009.

[22] W. M.P. van der Aalst and C. Stahl, *Modeling Business Processes: A Petri Net-Oriented Approach*. Cambridge, MA, USA: MIT Press, May 2011.

[23] H. Genrich, R. Küffner, and K. Voss, "Executable Petri net models for the analysis of metabolic pathways," *Int. J. Softw. Tools Technol. Transf.*, vol. 3, no. 4, pp. 394–404, 2001.

[24] K. Voss, M. Heiner, and I. Koch, "Steady state analysis of metabolic pathways using Petri nets," *Silico Biol.*, vol. 3, no. 3, pp. 367–387, 2003.

[25] J.-P. Comet, H. Klaudel, and S. Liauzu, "Modeling multi-valued genetic regulatory networks using high-level Petri nets," in *Proc. 26th Int. Conf. Appl. Theory Petri Nets*, 2005, pp. 208–227.

[26] M. Heiner and D. Gilbert, "BioModel engineering for multiscale systems biology," *Prog. Biophys. Mol. Biol.*, vol. 111, nos. 2–3, pp. 119–128, 2013.

[27] F. Liu, M.-A. Blätke, M. Heiner, and M. Yang, "Modelling and simulating reaction–diffusion systems using coloured Petri nets," *Comput. Biol. Med.*, vol. 53, pp. 297–308, Oct. 2014.

[28] R. V. Carvalho, J. van den Heuvel, J. Kleijn, and F. J. Verbeek, "Coupling of Petri net models of the mycobacterial infection process and innate immune response," *Computation*, vol. 3, no. 2, pp. 150–176, 2015.

[29] M. Pennisi, S. Cavalieri, S. Motta, and F. Pappalardo, "A methodological approach for using high-level Petri nets to model the immune system response," *BMC Bioinform.*, vol. 17, p. 498, Dec. 2016.

[30] D.-E. Gratie and C. Gratie, "Composition colored Petri nets for the refinement of reaction-based models," *Electron. Notes Theor. Comput. Sci.*, vol. 326, pp. 51–72, Oct. 2016.
[31] F. Liu, M. Heiner, and D. Gilbert, "Coloured Petri nets for multilevel, multiscale and multidimensional modelling of biological systems," *Briefings Bioinform.*, pp. 1–10, Nov. 2017.
[32] J. Yang, R. Gao, M. Q.-H. Meng, and T.-J. Tarn, "Colored petri nets to model gene mutation and amino acids classification," *J. Theor. Biol.*, vol. 300, pp. 183–192, May 2012.
[33] J. Yang, J. Lian, H. Pu, and R. Gao, "Modeling the genetic information transmission based on colored Petri nets," in *Proc. IEEE Int. Conf. Inf. Autom.*, Hailar, China, Jul. 2014, pp. 1025–1029.
[34] J. Yang, H. Pu, J. Lian, and G. Ren, "Modeling the classification of amino acids with colored Petri nets," in *Proc. IEEE Int. Conf. Inf. Autom.*, Lijiang, China, Aug. 2015, pp. 1763–1768.
[35] J. Yang, J. Lian, H. Pu, and J. Gu, "Modeling and analysis of DNA mutation type based on colored Petri net," in *Proc. IEEE Int. Conf. Inf. Autom.*, Macau, China, Jul. 2017, pp. 864–869.
[36] M. Heiner and I. Koch, "Petri net based model validation in systems biology," in *Proc. 25th Int. Conf. Appl. Theory Petri Nets*, Bologna, Italy, 2004, pp. 216–237.
[37] (Sep. 2017). *Protein Synthesis Steps*. [Online]. Available: http://www.proteinsynthesis.org/protein-synthesis-steps/
[38] S. Kanaya and R. J. Crouch, "DNA sequence of the gene coding for Escherichia coli ribonuclease H," *J. Biological Chem.*, vol. 258, pp. 1276–1281, Jan. 1983.
[39] (Oct. 2017). *Gene Home Reference: Mutations and Health*. [Online]. Available: https://ghr.nlm.nih.gov/
[40] P. D. Stenson *et al.*, "The human gene mutation database: Towards a comprehensive repository of inherited mutation data for medical research, genetic diagnosis and next-generation sequencing studies," *Hum. Genet.*, vol. 136, no. 6, pp. 665–677, 2017.
[41] M. Mort, D. Ivanov, D. N. Cooper, and N. A. Chuzhanova, "A meta-analysis of nonsense mutations causing human genetic disease," *Hum. Mutation*, vol. 29, no. 8, pp. 1037–1047, 2008.
[42] M. H. Steinberg and A. H. Adewoye, "Modifier genes and sickle cell anemia," *Current Opinion Hematol.*, vol. 13, no. 3, pp. 131–136, 2006.
[43] S. L. Thein, "The molecular basis of β-thalassemia," *Cold Spring Harbor Perspect. Med.*, vol. 3, no. 5, p. a011700, 2013.
[44] R. K. Humphries, T. J. Ley, N. P. Anagnou, A. W. Baur, and A. W. Nienhuis, "Beta O-39 thalassemia gene: A premature termination codon causes beta-mRNA deficiency without affecting cytoplasmic beta-mRNA stability," *Blood*, vol. 64, no. 1, pp. 23–32, 1984.
[45] H. T. Truong, T. Dudding, C. L. Blanchard, and S. H. Elsea, "Frameshift mutation hotspot identified in Smith-Magenis syndrome: Case report and review of literature," *BMC Med. Genet.*, vol. 11, p. 142, Oct. 2010.
[46] F. Supek, B. Miñana, J. Valcárcel, T. Gabaldón, and B. Lehner, "Synonymous mutations frequently act as driver mutations in human cancers," *Cell*, vol. 156, no. 6, pp. 1324–1335, 2014.
[47] Z. E. Sauna and C. Kimchi-Sarfaty, "Understanding the contribution of synonymous mutations to human disease," *Nature Rev. Genet.*, vol. 12, pp. 683–691, Aug. 2011.
[48] J. C. Roach *et al.*, "Analysis of genetic inheritance in a family quartet by whole-genome sequencing," *Science*, vol. 328, no. 5978, pp. 636–639, 2010.
[49] M. Ueno *et al.*, "Synonymous mutations in stem-loop III of Rev responsive elements enhance HIV-1 replication impaired by primary mutations for resistance to enfuvirtide," *Antiviral Res.*, vol. 82, no. 1, pp. 67–72, 2009.

JINLIANG YANG (M'13) received the Ph.D. degree in pattern recognition and intelligent systems from Shandong University, China, in 2013. He was a Visiting Student with the Department of Mechanical, Aerospace and Biomedical Engineering, The University of Tennessee, Knoxville, TN, USA, from 2011 to 2012. He is currently a Lecturer with the Department of Electrical Engineering and Information Technology, Shandong University of Science and Technology, China. He is also a Visiting Scholar with the Department of Electrical and Computer Engineering, Dalhousie University, Canada. His current research interests include the mathematical modeling and control of molecular biology, bioinformatics, and pattern recognition in bio-system.

HAITAO PU received the M.S. and Ph.D. degrees in control theory and control engineering from the Shandong University of Science and Technology, China, in 2007 and 2011, respectively. He is currently an Associate Professor with the Department of Electrical Engineering and Information Technology, Shandong University of Science and Technology. His current research interests include bioinformatics, Internet of Things, and wireless sensor networks.

JIAN LIAN received the M.S. degree from the Shandong University of Science and Technology, China, in 2007. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Shandong Normal University, China. He is also a Lecturer with the Department of Electrical Engineering and Information Technology, Shandong University of Science and Technology. His current research interests include medical image processing and bio-control.

JASON GU received the bachelor's degree in electrical engineering and information science from the University of Science and Technology of China in 1992, the master's degree in biomedical engineering from Shanghai Jiao Tong University in 1995, and the Ph.D. degree from the University of Alberta, Canada, in 2001. He has over 20 years of research and teaching experience. He is currently a Full Professor of electrical and computer engineering with Dalhousie University, Canada. He is also a Cross-Appointed Professor with the School of Biomedical Engineering for his multidisciplinary research work. He has authored over 240 conference papers and articles. His research interests include robotics, biomedical engineering, rehabilitation engineering, neural networks, and control. He is a fellow of the Engineering Institute of Canada. He has been an Associate Editor of the *Journal of Control and Intelligent Systems*, the *Transactions of the Canadian Society for Mechanical Engineering*, Canada, the IEEE TRANSACTIONS ON MECHATRONICS, the *International Journal of Robotics and Automation*, *Unmanned Systems*, the *Journal of Engineering and Emerging Technologies*, and the IEEE ACCESS.

MINGQU FAN received the Ph.D. degree in ocean information sensing and processing from the Ocean University of China, Qingdao, Shandong, China, in 2008. He is currently a Professor with the Department of Electrical Engineering and Information Technology, Shandong University of Science and Technology, China. His current research interests include information system design, optimal control theory, and nonlinear control in molecular biology.