frontiers in Genetics

METHODS
published: 09 June 2020
doi: 10.3389/fgene.2020.00515

Using Cellular Automata to Simulate Domain Evolution in Proteins

Xuan Xiao[1], Guang-Fu Xue[1], Biljana Stamatovic[2] and Wang-Ren Qiu[1]**

[1] Computer Department, Jing-De-Zhen Ceramic Institute, Jingdezhen, China, [2] Faculty of Information Systems and Technologies, University of Donja Gorica, Podgorica, Montenegro

Proteins play primary roles in important biological processes such as catalysis, physiological functions, and immune system functions. Thus, the research on how proteins evolved has been a nuclear question in the field of evolutionary biology. General models of protein evolution help to determine the baseline expectations for evolution of sequences, and these models have been extensively useful in sequence analysis as well as for the computer simulation of artificial sequence data sets. We have developed a new method of simulating multi-domain protein evolution, including fusions of domains, insertion, and deletion. It has been observed via the simulation test that the success rates achieved by the proposed predictor are remarkably high. For the convenience of the most experimental scientists, a user-friendly web server has been established at http://jci-bioinfo.cn/domainevo, by which users can easily get their desired results without having to go through the detailed mathematics. Through the simulation results of this website, users can predict the evolution trend of the protein domain architecture.

Keywords: simulation, protein evolution, cellular automaton, multi-domain proteins, protein domain architecture

OPEN ACCESS

Edited by:
Juan Caballero, Universidad Autónoma de Querétaro, Mexico

Reviewed by:
Franco Bagnoli, University of Florence, Italy
Hiep Xuan Huynh, Can Tho University, Vietnam

****Correspondence:***
Xuan Xiao
jdzxiaoxuan@163.com
Wang-Ren Qiu
qiuone@163.com

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Genetics

Received: *24 September 2019*
Accepted: *28 April 2020*
Published: *09 June 2020*

Citation:
Xiao X, Xue G-F, Stamatovic B and Qiu W-R (2020) Using Cellular Automata to Simulate Domain Evolution in Proteins. Front. Genet. 11:515. doi: 10.3389/fgene.2020.00515

INTRODUCTION

Proteins are the biological macromolecular entities most close-knitly related to organismal functions. Evolution in the sequences of proteins results in the way these proteins function, and protein evolution is a critical component of organismal evolution and a valuable method for generating useful molecules in the laboratory (Leconte et al., 2013). Therefore, the research on protein evolution plays an elementary and central role in computational proteomics. Experimental efforts to understand protein evolution have largely depended on the reconstruction of hypothetic evolutionary intermediates or on experimental evolution over modest numbers of rounds of evolution (Weinreich et al., 2006; Gumulya et al., 2012). Long evolutionary trajectory experiments have met challenges in studying proteins but have been successfully executed only for whole organisms and RNA. Directed evolution has been a powerful technique for generating tailor-made enzymes for a wide range of biocatalytic applications (Zeymer and Hilvert, 2018), but it is both time- and money-consuming to study protein evolution by conducting experiments alone. With the rapid development of computational power, hidden Markov model based on statistics, phylogeny model based on Bayesian statistics, and better prediction method of protein structure, the ability to model evolutionary processes in proteins has improved.

Protein evolution is modeled firstly by considering the amino acid substitution process. Dayhoff et al. (1978) proposed the most influential amino acid substitution model. This simple model supposes that all sites in the protein sequence are independent of each other during protein evolution, and that each site mutation depends on an amino acid replacement matrix. Since then, many protein evolution models based on amino acid substitution matrices have been proposed,

such as the JTT model (Jones et al., 1992), the mtREV model (Adachi and Hasegawa, 1996), and the WAG model (Whelan and Goldman, 2001). However, in most cases, the assumption that "the proteins are independent of each other during evolution" is not consistent with the fact that any amino acid residue within the protein interacts with its neighboring amino acids. Yang (1993) has designed an ingenious method that allows variant sites in the amino acid sequence to have variant rates of evolution. This method basically classifies amino acids according to their physicochemical properties, making amino acids with similar properties more likely to be replaced (Yang, 1993). Protein evolution is driven by the sum of variant physiochemical and genetic processes that usually results in strong purifying selection to maintain biochemical functions. However, proteins that are part of systems under arms race dynamics often evolve at unparalleled rates that can produce atypical biochemical properties (Wilburn et al., 2018).

Phylogenetic methods have been widely used to analyze the evolutionary history of protein sequences. The simulation of sequences is one means of investigating phylogenetic hypotheses (Tuffery, 2002). There are many more powerful bioinformatics tools for such simulations (Bakan et al., 2014). Sirakoulis et al. (2003) used a cellular automaton (CA) model for the study of DNA sequence evolution where DNA is modeled as a one-dimensional (1D) CA with four states per cell which correspond to four DNA bases. Moreover, they have developed genetic algorithms in order to determine the rules of CA evolution that simulate the DNA evolution course. Simulation models for protein evolution based on CA have lagged far behind models of DNA evolution because proteins are composed of 20 amino acids, while DNA is composed of only four nucleotides. Many authors developed different simulations of protein evolution, but few of them are operated by non-expert users. They are either very specific for certain needs or distributed as non-interactive command-line programs or require a complex preparation of the input data. This precludes these techniques being used by most molecular biologists.

Proteins are composed of domains, recurrent protein fragments with distinct structure and function, and proteins can be classed as single-domain proteins or multi-domain proteins (Chothia, 1992; Riley and Labedan, 1997). The structural domain databases SCOP and CATH were gathered based on identifying recurring elements in experimentally determined protein three-dimensional (3D) structures (Dawson et al., 2016; Chandonia et al., 2017). In Pfam databases, conserved regains are identified from sequence analysis and background knowledge to make multiple sequence alignments (El-Gebali et al., 2019). Domain definitions form different databased only partially overlap; however, the choice of database appears to have little effect on modeling the evolution of protein domain architectures (Apic et al., 2001). Domain architecture generally refers to the domains in a protein and their order, reported in N- to C-terminal direction along the amino acid chain. The mechanisms for domain architecture change can be classed into new domain, fission, and fusion (Fong et al., 2007). The multi-domain architectures usually evolve from existing architectures because few multi-domain architectures contain all new domain. Fusion class would be partitioned into three sub-cases as fusion of new domains, fusion of parent architectures, and fusion of parent architecture and new domain. Snel et al. (2000) summarized that domain fusions are more common that domain fissions, and the result was subsequently supported by a larger study by Kummerfeld and Teichmann (2005). Buljan and Bateman observed that domain architecture changes primarily take place at the protein termini and it can be explained from that terminal changes to the architecture are less likely to disturb overall protein structure (Buljan and Bateman, 2009), and similar results have been found in several other studies (Buljan et al., 2010). Zhang et al. (2012) and Sharma and Pandey (2016) studied the role of gene duplication in plants protein domain architecture evolution. More recently, Wiedenhoeft et al. (2011) used a network construct named as plexus to reconstruct domain architecture history. Stolzer et al. (2015) present another method for domain architecture history inference, made available through the Notung software.

Multi-domain proteins have evolved by insertions or deletions of distinct protein domains. We have a general understanding of the mechanisms of protein domain architecture evolution based on the aforementioned models. Here we introduce a new protein evolution simulation model to simulate the evolution of protein domains by 1D CA. In the model, the HMMER (Prakash et al., 2017) and Pfam databases are united in the process for annotating the protein domains, and it can be easily to simulate the evolution of the domain architecture in the multi-domain protein family. Furthermore, the model may obtain new domain architecture which may be the potential protein evolution.

METHODS AND IMPLEMENTATION

Data Preprocessing

After receiving the protein sequence file P ($P = \{p_1, p_2, p_3, \ldots, p_n\}$, where $p_1, p_2, p_3, \ldots, p_n$ represent the protein sequence in the file) in FASTA format, the system annotates the protein domain of each protein sequence p_i based on the HMMER and Pfam databases, and each protein p_i generates a corresponding annotation file f_i. By analyzing the annotation file f_i, the domain information of protein p_i can be screened out and expressed as a multi-domain sequence:

$$p_i = \{d_{i,1}, d_{i,2}, d_{i,3}, \ldots, d_{i,k}\} \quad (1)$$

where $d_{i,1}, d_{i,2}, d_{i,3}, \ldots, d_{i,k}$ represent the homologous domain of the protein p_i, k is the number of domains in protein p_i. According to the ACC (the average posterior probability of the aligned target sequence residues) value of each domain, we determine the position of these domains in the protein sequence or the order of domains in the sequence. The protein p_{i} is expressed as an ordered multi-domain sequence $\mathrm{p_i}'$ based on the context of these domains:

$$p_i' = d_{i,1}', d_{i,2}', d_{i,3}', \ldots, d_{i,k}' \quad (2)$$

The protein sequence file P is expressed as a set of multi-domain sequences as file P′:

$$\mathrm{P}' = \begin{pmatrix} p_1{}' \\ p_2{}' \\ \vdots \\ p_n{}' \end{pmatrix} = \begin{pmatrix} d_{1,1}{}', d_{1,2}{}', \ldots, d_{1,w}{}' \\ d_{2,1}{}', d_{2,2}{}', \ldots, d_{2,u}{}' \\ \vdots \\ d_{n,1}{}', d_{n,2}{}', \ldots, d_{n,l}{}' \end{pmatrix} \quad (3)$$

where w is the number of domains in protein $\mathrm{p}_1{}'$, u is the number of domains in protein $\mathrm{p}_2{}'$, and so forth. Therefore, a considerate protein sequence file P in FASTA format is processed by the above methods to form the training data set P' for the proposed evolutionary simulation model.

Inspired by incorporating the dipeptide position-specific propensity into the general pseudo nucleotide composition (Xiao et al., 2016), here we develop a new method to simulate the protein evolution process by domain position-specific propensity.

If g domain classes appeared in the training data set P', we added two termination symbols "X-start" and "X-end" in the front and the back of the $p_i{}'$, there are ($g^2 + g \times 2$) couple: X-startD_1, X-startD_2,…, D_1D_1 (where domain D_1 and domain D_1 are connected together), D_1D_2, D_1D_3,…, D_gD_g, D_1X. -ending,…, D_gX-ending. Thus, for the training data set P', its profile (or detailed information) of the domain position-specific propensity can be summarized by the following two matrices:

$$F2B = \begin{bmatrix} X\text{–start } D_1 & D_1D_1 & D_1D_2 & \cdots & D_1D_g & D_1\,X\text{–end} \\ X\text{–start } D_2 & D_2D_1 & D_2D_2 & \cdots & D_2D_g & D_2\,X\text{–end} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ X\text{–start } D_g & D_gD_1 & D_gD_2 & \cdots & D_gD_g & D_g\,X\text{–end} \end{bmatrix} \quad (4)$$

$$B2F = \begin{bmatrix} X\text{–start}D_1{}' & D_1{}'D_1{}' & D_1{}'D_2{}' & \cdots & D_1{}'D_g{}' & D_1{}'X\text{–end} \\ X\text{–start}D_2{}' & D_2{}'D_1{}' & D_2{}'D_2{}' & \cdots & D_2{}'D_g{}' & D_2{}'X\text{–end} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ X\text{–start}D_g{}' & D_g{}'D_1{}' & D_g{}'D_2{}' & \cdots & D_g{}'D_g{}' & D_g{}'X\text{–end} \end{bmatrix} \quad (5)$$

where $F2B$ (fromfronttoback) is matrix of evolution prior probability from front to back and $B2F$(from back to front) is the matrix of evolution prior probability from back to front. In the matrix $F2B$, D_iD_j ($1 \le i \le g$, $1 \le j \le g$) is the occurrence frequency of domain D_i attached to D_j, and D_j behind D_i. In the matrix $B2F$, $D_i{}'D_j{}'$ is the occurrence frequency of domain D_i attached to D_j, and D_i is followed by D_j.

Simulation Model of Domain Evolution in Proteins Based on Cellular Automaton

Let us give a brief introduction to CA. A CA is a dynamical system in which space, time, and the states are discrete. Each cell, defined by a point in a regular spatial lattice, can be any one of a finite number of states that are updated according to a local rule (Schwartz et al., 1967; Chopard and Droz, 1998). In 1D CA, the lattice consists of identical cells, i-m, …, i-3, i-2, i-1, i, i+1, i+2, i+3, …, i+m, and the corresponding states of these cells are $C_{i-m}, \cdots, C_{i-2}, C_{i-1}, C_i, \ C_{i+1}, C_{i+2}, \cdots, C_{i+m}$. The symbol i is the center of initial sequence with length equals to $2m + 1$. The state of the ith cell takes value from a predefined discrete set: $C_i \in \{c_1, c_2, \cdots, c_Q\}$, where $c_1, c_2, \cdots, c_Q$ are the elements of the set. The CA evolves in discrete time steps, and its evolution is manifested by the change of its cell states with time. The state of each cell is affected by the states of its neighboring cells. The neighborhood is defined as $\mathrm{N}(i, \mathrm{r}) = \{C_{i-r}, \cdots, C_{i-1}, C_i, C_{i+1}, \cdots, C_{i+r}\}$, where r is the size of the neighborhood. If $r = 1$, the neighborhood of the ith cell consists of the same cell and its left and right immediate neighbors $\mathrm{N}(i, 1) = \{C_{i-1}, C_i, C_{i+1}\}$. The state of the ith cell at time step t+1 is affected by the states of its neighbors at the previous time step t, $C_i^{t+1} = F\left(C_{i-r}^t, \cdots, C_{i-1}^t, C_i^t, C_{i+1}^t, \cdots, C_{i+r}^t\right)$. F is the CA evolution rule (Sirakoulis et al., 2003). **Figure 1** shows the evolution of a 1D CA. the horizontal axis is space, and the vertical axis is time. Each column represents the state of cell at various time steps.

In this study, the non-uniform 1D CA was used to simulate the domain evolution in proteins. The square arrays are a very basic data structure in computers, and it was rational to use a square lattice in our model. If protein $\mathrm{p_q}$ in the training data set P′ has the most domains, and the number of domains is m, then the spatial dimension of the 1D CA in the model is $1 \times (2m+1)$ and the time step in CA evolution is set as m. The model was a (g+3)-state model in which each cell in the lattice was one of the following (g+3) states: (1) g domain classes appeared in the training data set P′; (2) evolution termination symbols "X-start"; (3) evolution termination symbols "X-end"; (4) an empty state "$\emptyset$". The state of the cell at time $t$can be expressed as C_j^t. The upper index in the state symbol denotes the time step, and the lower index denotes the cell j. When the CA is initialized ($t = 0$), the state of the cell C_i^0 in the middle of the CA is set to the ancestral domain Y, and the state of the other cells is set to the empty state $\emptyset$, as shown in **Figure 2**. The state C_j^{t+1} of the cell at time $t + 1$ is determined by the state C_i^t of the cell at time t and the state $C_{j-1}^t$and C_{j+1}^t of its neighbor cells at time t.

The evolution rules of the proposed CA model can be expressed as follows (**Figure 3**):

Rule A: ***Inheritance***. If the state of cell C_j^t is domain $E(E \neq \emptyset)$, it means that the cell has evolved into domain E, so this cell

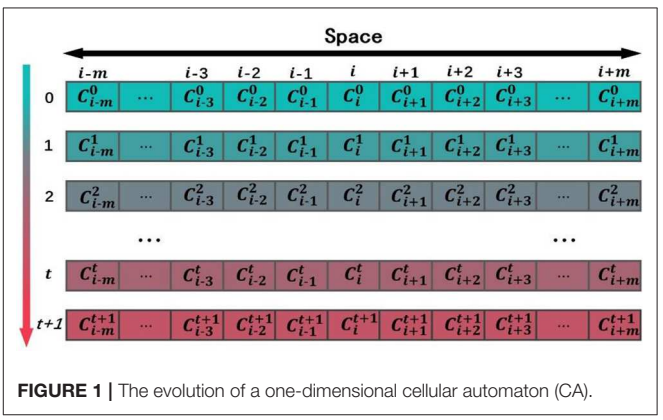

FIGURE 1 | The evolution of a one-dimensional cellular automaton (CA).

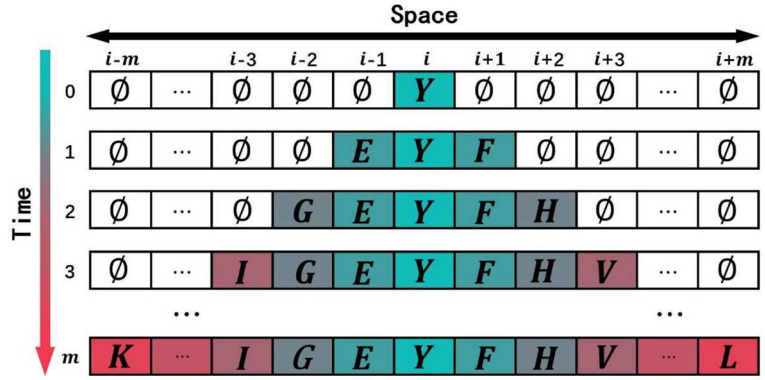

FIGURE 2 | Schematic drawing to show the initial state of one-dimensional cellular automaton (CA) and the course of evolution. Each site of this lattice is called cell. The value of domain is the state of the cell.

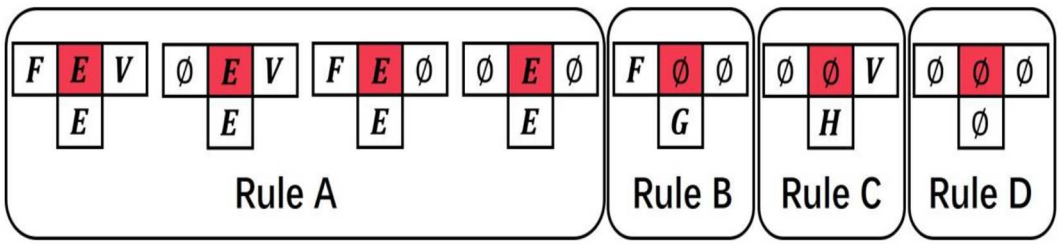

FIGURE 3 | The evolution of cellular automaton (CA). The state of each cell is affected by the states of its neighboring cells.

C_j^t will inherit the domain at the next time, which means that the state of C_j^{t+1} is E.

Rule B: ***Rules of evolution from front to back***. If the state of cells C_j^t and C_{j+1}^t is the empty domain $\emptyset$, and the state of cell C_{j-1}^t is domain $F(F \neq \emptyset)$, the state C_j^{t+1} will be obtained by the Roulette wheel selection algorithm. According to the matrix *F2B* (from front to back evolution prior probability matrix), we know that the probability of all domains to appear behind domain F. It is not true that the domain of the biggest probability certainly appears behind domain F in natural evolution course. Hence, we determined the state C_j^{t+1} into one domain form all domains that appear behind domain F probability is not zero based on Roulette wheel selection algorithm. The probability of selecting a domain G is the same as the probability of domain F appearing after domain G in matrix *F2B*.

Rule C: ***Rules of evolution from back to front***. If the state of cells C_{j-1}^t and C_j^t is the empty state $\emptyset$, and the state of cell C_{j+1}^t is domain $V(V \neq \emptyset)$, then the state C_j^{t+1} will be obtained by the Roulette wheel selection algorithm. The selected course is similar to Rule B except that the matrix *B2F* (from back to front evolution prior probability matrix) was used instead of the matrix *F2B*. The probability of selecting a domain H being selected is the same as the probability of domain H appearing before domain V in matrix B2F.

Rule D: ***Return inanimateness***. If the states of cells C_{j-1}^t, C_j^t, and C_{j+1}^t are all equal to the empty state $\emptyset$, the state C_j^{t+1} will remain the empty state $\emptyset$.

Proteins in a family descend from a common ancestor and have similar 3D structures, functions, and sequence similarity. Thus, the model assumes that all of the evolved proteins contain ancestral domains $Y(Y \in \{D_1, D_2, D_3, \ldots, D_g, X\text{-}start, X\text{-}end\})$, and where the common domains in the P′ are considered to be the hypothetical ancestral domain. By running the model once, a new protein will be simulated, and the evolved protein is represented by an ordered sequence of multi-domains.

As shown in **Figure 4**, when the CA is initialized ($t = 0$), the state of the cell C_i^0 in the middle of the CA is set to the ancestral domain Y, and the state of the other cells is set to the empty state $\emptyset$.

When $t = 1$, the state of C_i^1 is determined by the state of C_{i-1}^0, C_i^0 and C_{i+1}^0. Since the state of C_i^0 is the domain $Y(Y \neq \emptyset)$, then C_i^1 will be domain Y according to rule A inherit the state of C_i^0; the state of C_{i-1}^1 is determined by the state of C_{i-2}^0, C_{i-1}^0 and C_i^0. Because the state of C_{i-1}^0 is the empty domain $\emptyset$ and the state of C_i^0 is the domain Y, the state of C_{i-1}^1 should be selected by the roulette method according to rule C and the matrix *B2F* to obtain the domain E. The state of C_{i+1}^1 is determined by the state of C_i^0, C_{i+1}^0 and C_{i+2}^0 with the reason that the state of C_i^0 is the domain Y and the state of C_{i+1}^0 is the domain $\emptyset$. According to rule B, the state of C_{i+1}^1 should be selected by the roulette method

on the basis of the matrix *F2B* to obtain the domain *F* after the domain *Y*. Keep the rule *D* in mind, the state of the other cells is the empty domain Ø because the state of the other cells at the previous moment and the states of their neighbors are the empty domain Ø.

When $t = 2$, the states of C^2_{i-1}, C^2_i, and C^2_{i+1} are determined by the states of $C^1_{i-1}, C^1_i, C^1_{i+1}$ and their neighbors with the reason that the states of C^1_{i-1}, C^1_i and C^1_{i+1} are not domain Ø. According to the rule A, C^2_{i-1}, C^2_i, and C^2_{i+1} will inherit the states, respectively, C^1_{i-1}, C^1_i, and C^1_{i+1} with domains *E*, *Y* and domain *F*; the state of C^2_{i-2} is determined by the state of C^1_{i-3}, C^1_{i-2}, and C^1_{i-1} at the previous time. Because the states of C^1_{i-3} and C^1_{i-2} are the domain Ø and the state of C^1_{i-1} is the domain *E*, according to rule C, the state of C^2_{i-2} will be selected by the roulette method according to the matrix *B2F* to obtain the domain *G* in front of the domain *E*; the state of C^2_{i+2} is determined by the state of C^1_{i+1}, C^1_{i+2}, and C^1_{i+3}. Because the states of C^1_{i+2} and C^1_{i+3} are the domain Ø and the state of C^1_{i+1} is the domain *F*, the state of C^2_{i+2} will be selected by the roulette method according to rule B and the matrix *F2B* to obtain the domain *H* after the domain *F*. According to rule D, the state of the other cells is the domain Ø on the basis of that the state of the other cells and the states of their neighbors are the domain Ø.

When $t = k (k < m)$, the cells with the state of termination symbols *X*-start and *X*-*end* are evolved in the cell space, and the simulated evolution of the protein comes to an end. The state of the cell space at time *k* is taken as an ordered sequence of multi-domain, removing the cells with the empty state Ø. The simulated multi-domain sequence of protein A is expressed as:

$$\begin{aligned} &\{ X\text{–}start\,,\ \ldots\,,\ I\,,\ G\,,E\,,Y\,,\ F\,,\ H\,,\ \ldots\,,\ X\text{–}end\,\} \\ &(\,E\,,\ F\,,\ G\,,\ H\,,\ I\,,\ Y\ \in \{D_1, D_2, D_3, \ldots, D_g\}) \end{aligned} \tag{6}$$

As shown in **Figure 5**, when $t = m$, if termination symbols *X*-start and *X*-*end* have not evolved in the cell space, the simulated evolution of the protein also comes to an end. The state of the

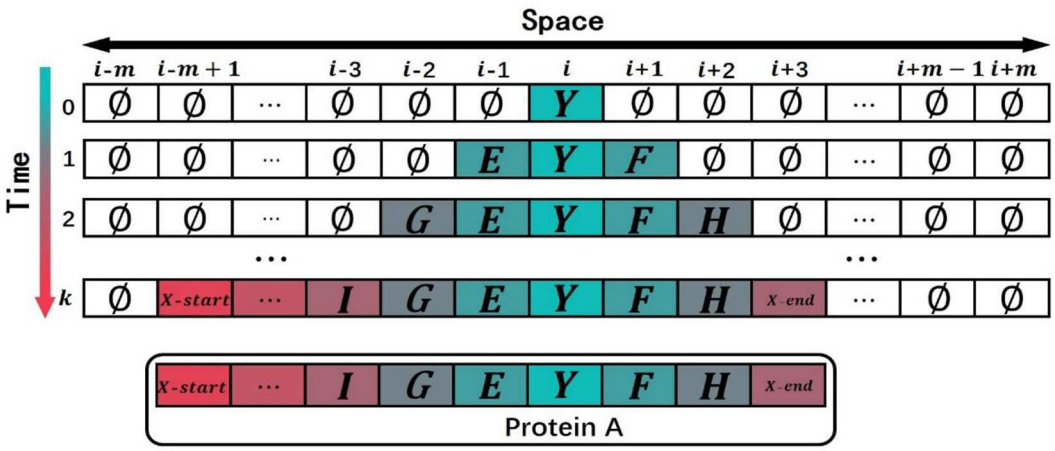

FIGURE 4 | An example to show how the model simulated the domain evolution. The model evolved the termination symbols *X*-start and *X*-end in advance at time *k*, so the protein stopped the evolution at time *k*.

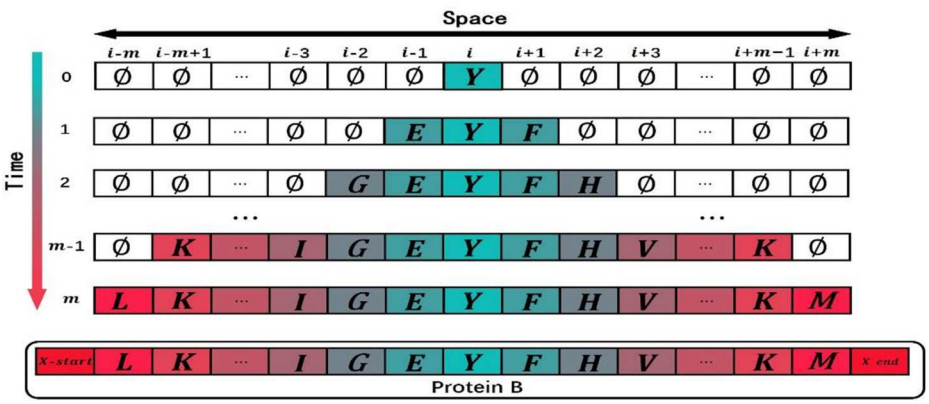

FIGURE 5 | An example to show how to put a termination to domain evolution. When the model is at time *m*, the termination symbols *X*-start and *X*-end have not yet evolved. Since the maximum time step of the cellular automaton is *m*, the protein stops evolution at time *m*.

cell space at time m is taken as an ordered sequence of multi-domain, removing the cells with the empty state $\emptyset$ and adding two termination symbols to the left and right ends of the cell space. The simulated multi-domain sequence of protein B is expressed as:

$$\{X\text{–}start, L, K, ..., I, G, E, Y, F, H, V, ..., K, M, X\text{–}end\}$$
$$(E, F, G, H, I, J, K, L, M, V, Y \in \{D_1, D_2, D_3, \ldots, D_g\}) \quad (7)$$

USER INTERFACE

In order to facilitate the use of researchers, we have developed a web server, where users can directly submit protein sequence files and select various parameters for protein evolution simulation. The system will send the results to the user's e-mail address. The graphical user interface of the website is shown in **Figure 6**.

Input File

The user uploads a protein sequence file in FASTA format by clicking the button "Submit Sequence."

Ancestral Domain

The ancestral domains are the initial state of the cell in the middle of the CA space. The model uses this parameter as a common ancestor of the protein. Once this parameter is filled in, all evolved proteins will contain this domain.

E-Value

The E-value is a parameter used in the HMMER software; it is the expected number of false positives (non-homologous sequences) that scored this well or better. The E-value is a measure of statistical significance. The lower the E-value, the more significant the hit. Changing the value of E-value will cause the same protein to compare different domain architectures, and the default value is "1e-37."

Evo-Num

The number of times of model simulation.

Top-Num

The model analyzes automatically the evolved proteins and sends the top top-num frequency of the domain architectures to users.

After submitting the protein sequence file and parameters, the system will compare the uploaded protein sequences based on HMMER. By setting the value of the E-value, each protein will generate a homology domain information file. The domain of each protein is then extracted and sorted according to the position of the domain in the protein sequence, such that each protein can be represented as a multi-domain sequence.

Next, the sliding window processing is performed on each multi-domain sequence (the sliding window size is 2), the matrix of probability from back to front and the matrix of probability from front to back are obtained by counting the frequency of the domain in pairs.

In the evolution process of the CA, the next time state of the cell is obtained according to the CA evolution rule. When evolution is terminated, the system removes the domain $\emptyset$ and generates a complete multi-domain protein. The user can control the number of proteins that the system simulates by adjusting the size of Evo-num.

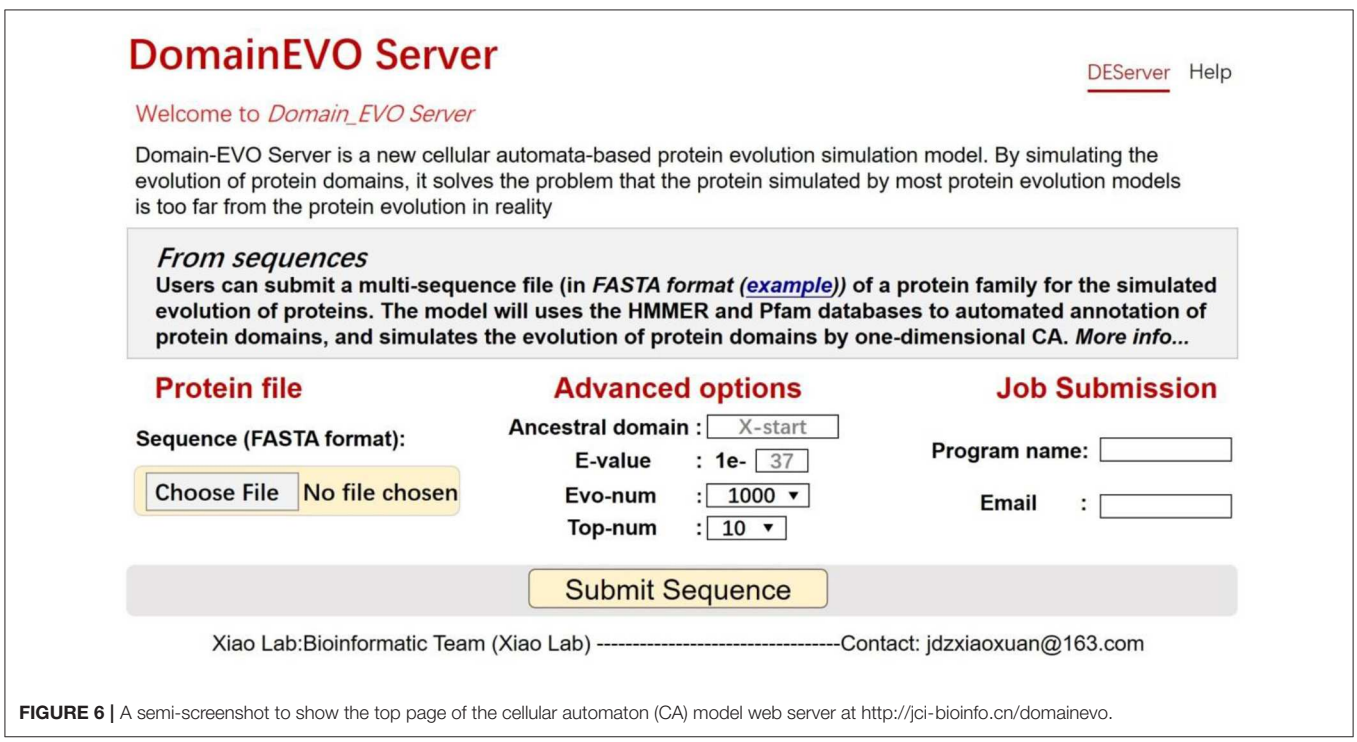

FIGURE 6 | A semi-screenshot to show the top page of the cellular automaton (CA) model web server at http://jci-bioinfo.cn/domainevo.

After the evolution of the protein, the system calculates the frequency of each domain architecture based on the multi-domain protein of the original file and the multi-domain protein obtained by the proposed simulation model, sorts the frequency from large to small, and saves it as two files in comma-separated values (CSV) format. Since there are many protein domain architectures obtained by simulation, users can adjust the value of Top-num to preserve the domain architectures with high frequency.

Finally, the system will send the results to the user's e-mail address. The contents of the e-mail are the parameters selected by the user, and in the attachment, there are two data files in CSV format. The CA model can be described by the flowchart in **Figure 7**.

SIMULATING THE EVOLUTION OF RHOGEF DOMAIN IN *HOMO SAPIENS* PROTEINS

Materials

To evaluate the performance of the model, we used a number of multi-domain proteins associated with the conserved protein family "RhoGEF" to validate the validity of the model. The keyword "RhoGEF" was used to find the protein of the conserved protein family RhoGEF from the NCBI database. We selected the species *Homo sapiens* to download a multi-sequence file containing 1,597 proteins and used it as a protein sequence file P. In addition to the above methods, researchers can also build data sets by the following methods. The Conservative Domain Database (CDD, https://www.ncbi.nlm.nih.gov/cdd/) collects a large number of protein domains and domain families (Marchler-Bauer et al., 2016). Researchers can search for a conserved protein domain family in the CDD website and then click the "related protein" on the domain family description page to link to the NCBI website to download the related proteins.

Model Performance Evaluation Under Various Parameters

After submitting the protein sequence file, we tested various parameters that may affect the evolution of protein simulation. The protein sequence file P is processed into a training data set P′ of the evolutionary simulation model:

$$\mathrm{P}' = \begin{pmatrix} p_1{}' \\ p_2{}' \\ \vdots \\ p_{1597}{}' \end{pmatrix} = \begin{pmatrix} d_{1,1}{}' & d_{1,2}{}' & \dots & d_{1,w}{}' \\ d_{2,1}{}' & d_{2,2}{}' & \dots & d_{2,u}{}' \\ \vdots & \vdots & \vdots & \vdots \\ d_{1597,1}{}' & d_{1597,2}{}' & \dots & d_{1597,l}{}' \end{pmatrix} \tag{8}$$

P^{evo} represents a multi-domain sequence file from the simulation of protein evolution in this model.

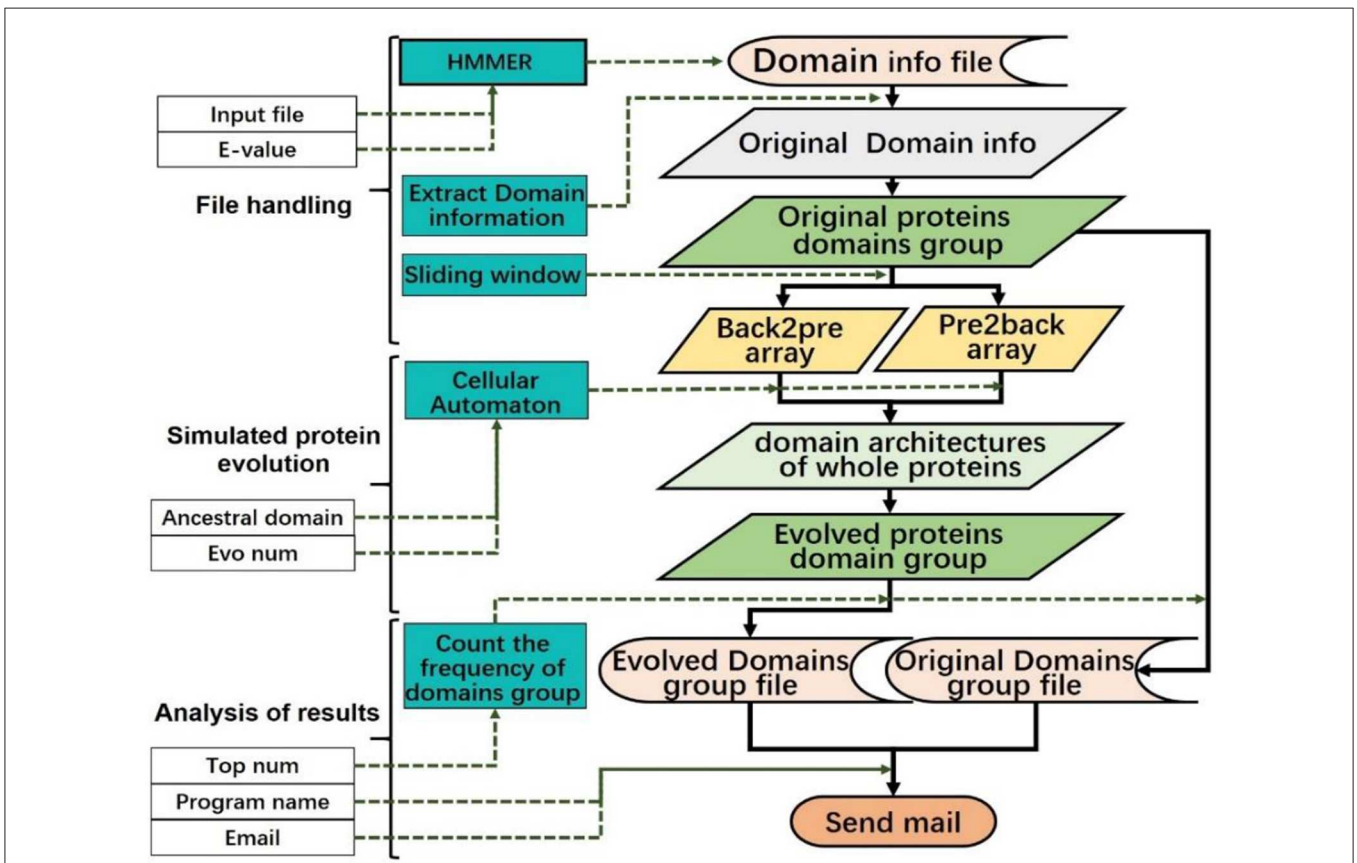

FIGURE 7 | Data flow diagram of the model. Different types of data are represented by different background colors.

$$P^{evo} = \begin{pmatrix} D_1^{evo} \\ D_2^{evo} \\ \cdots \\ D_{Evo\_num}^{evo} \end{pmatrix} = \begin{pmatrix} d_{1,1}^{evo} & d_{1,2}^{evo} & \cdots & d_{1,x}^{evo} \\ d_{2,1}^{evo} & d_{2,2}^{evo} & \cdots & d_{2,y}^{evo} \\ \vdots & \vdots & \vdots & \vdots \\ d_{Evo\_num,1}^{evo} & d_{Evo\_num,2}^{evo} & \cdots & d_{Evo\_num,z}^{evo} \end{pmatrix} \tag{9}$$

where $D_n^{evo}(1 \leq n \leq Evo\_num)$ represents the simulation of the evolution of domain architectures of whole proteins (DAWPs), x is the number of domain in simulation protein D_1^{evo}, y is the number of domain in simulation protein D_2^{evo}, and so forth. Protein domains are represented by $d_{j,k}^{evo}$.

In order to examine the performance of a predictor in simulating domain evolution of proteins, *Hit-Acc* (DAWP), goodness of fit between the simulation proteins and nature protein in P′, used in this literature based on counting the type and number of the DAWPs. *Hit-Acc* (TAPD), goodness of fit between the triplet domain architectures in P' and P^{evo}, *Hit-Acc* (QAPD), goodness of fit between the quadruple domain architectures in P' and P^{evo} as supplementary.

$$\text{Hit–Acc (DAWP)} = \frac{\sum_{i=1}^{Evo\_num} f^{evo}\left(D_i^{evo}\right)}{Evo\_num} \times 100 \tag{10}$$

$$f^{evo}\left(D_i^{evo}\right) = \begin{cases} 1, \ if\ D_i^{evo}\ appeared\ in\ \mathrm{P}' \\ 0, \ otherwise \end{cases} \tag{11}$$

$$\text{Hit–Acc (TAPD)} = \frac{\sum_{i=1}^{m} \theta^{evo}(t_i)}{\sum_{i=1}^{k} \theta^{evo}\left(t_i^{evo}\right)} \times 100 \tag{12}$$

$$\begin{cases} \mathrm{T}' = \left\{t_1', t_2', \ldots, t_e'\right\} \\ \mathrm{T^{evo}} = \left\{t_1^{evo}, t_2^{evo}, \ldots, t_f^{evo}\right\} \\ T = \mathrm{T}' \cap \mathrm{T^{evo}} = \{t_1, t_2, \ldots, t_r\} \end{cases} \tag{13}$$

where T′ is the set of triplet domain architectures in P′, $\mathrm{T^{evo}}$ is the set of triplet domain architectures in $\mathrm{P^{evo}}$, ∩ represents the symbol for "intersection" in the set theory, $\theta^{evo}(t_i)$ represents the count hits of triplet domain architectures t_i in P^{evo}, and $\theta^{evo}\left(t_i^{evo}\right)$ represents the hits of triplet domain architectures t_i^{evo} in P^{evo}.

$$\text{Hit–Acc (QAPD)} = \frac{\sum_{i=1}^{m} \varphi^{evo}\left(q_i\right)}{\sum_{i=1}^{k} \varphi^{evo}\left(q_i^{evo}\right)} \times 100 \tag{14}$$

$$\begin{cases} \mathrm{Q}' = \left\{q_1', q_2', \ldots, q_h'\right\} \\ \mathrm{Q^{evo}} = \left\{q_1^{evo}, q_2^{evo}, \ldots, q_a^{evo}\right\} \\ Q = \mathrm{Q}' \cap \mathrm{Q^{evo}} = \left\{q_1, q_2, \ldots, q_s\right\} \end{cases} \tag{15}$$

where Q' is the set of quadruple domain architectures in P', Q^{evo} is the set of quadruple domain architectures in P^{evo}, $\varphi^{evo}\left(q_i\right)$ represents the hits of quadruple domain architectures q_i in P^{evo}, and $\varphi^{evo}\left(q_i^{evo}\right)$ represents the hits of quadruple domain architectures q_i^{evo} appearing in P^{evo}.

In the process of testing system performance, we used HMMER to perform multi-domain sequence alignment on *Homo sapiens*' protein family RhoGEF, and set the value of E-values from 1e-1 to 1e-50, so that the protein domain information file corresponding to E-values would be generated, and 50 simulation training data sets would be obtained after processing the file. The smaller the value of E-value, the higher the accuracy of the aligned homology domain. Changing the value of E-value will cause the same protein match different domain architectures when the value of E-value is 1e-1. A total of 228 domains existed in the simulation data set P, and the number of types of protein domain architectures was 326. When the value of E-value is 1e-37, there are 43 domains in the simulation data set P, the number of types of protein domain architecture is 55, and the most complex protein domain architecture is {′*I-set*′, ′*V-set*′, ′*Ig*′, ′*Ig*′, ′*Ig*′, ′*Izumo-Ig*′, ′*Ig*′, ′*Ig*′, ′*Pkinase*′, ′*Pkinase*′}. The 10 most frequent protein domains are shown in **Table 1**.

The model uses RhoGEF, *X*. -start, and *X*-end as ancestral protein domains to simulate each training set, and each training set simulates 50,000 proteins represented by multiple domain sequences. In order to eliminate the impact of individual results on the model evaluation, we repeated the simulation of each parameter combination 50 times and then calculated the average value of Hit-Acc. as the evaluation of the model; the obtned results are shown in **Figures 8–10**.

When the ancestral protein domain is RhoGEF, the Hit-Acc (TAPD). and Hit-Acc (DAWP) reach the maximum; when the E-value is 1e − 43, the Hit-Acc (TAPD) is 90.27%, the Hit-Acc (DAWP) is 89.59%. When the ancestral protein domain is *X*-start or *X*-end, the test results are basically the same as the ancestral protein domain RhoGEF. The Hit-Acc (TAPD) and Hit-Acc (DAWP) reach the maximum when the E-value is 1e-37, the Hit-Acc (TAPD) is 91.01%, and the Hit-Acc (DAWP) is 88.26%. These test results show that the model has good simulation characteristics and robustness. The model can also get better results in most cases by using *X*-start and *X*-end as the ancestral protein domain for simulated evolution. The reason is that this model uses HMMER for automated annotation of

TABLE 1 | The 10 most frequent protein domains when E-values are 1e-1 and 1e-37, respectively.

E-value : 1e-1			E-value : 1e-37		
Domain	**Frequency**	**Probability (%)**	**Domain**	**Frequency**	**Probability (%)**
PH	2,776	20.07	RhoGEF	619	15.10
RhoGEF	1,440	10.41	Ig	243	5.93
SH3	1,091	7.89	PH	172	4.20
IQ	473	3.42	SH3	129	3.15
Ig	379	2.74	Pkinase	115	2.80
PDZ	314	2.27	RhoGEF67	73	1.78
C1	207	1.50	RGS-like	55	1.34
FYVE	204	1.47	RasGEF	52	1.27
EF-hand	190	1.37	I-set	49	1.20
Pkinase	187	1.35	V-set	48	1.17

protein domains. The smaller the E-values, the fewer domains are annotated. Some proteins will not annotate the ancestral domain because the set E-value is too small. For example, if the E-value is 1e-1, 1,440 proteins are annotated with the RhoGEF domain; if the E-value is 1e-37, only 619 proteins are annotated with the RhoGEF domain. In order to solve this problem, the model added *X*-start and *X*-end to the left and right ends of each protein when composing the protein simulation training data set. If the model uses *X*-start or *X*-end as the ancestral protein domain, each protein domain architecture could be simulated, which also allows the system to simulate more protein domain architectures that appear in the original file. This also means that if the user does not know the ancestor domain of the submitted protein sequence file, *X*-start or *X*-end can be used as the ancestral domain for protein simulation. It can be seen from the results that the model can effectively simulate the DAWPs under various parameters. It also simulated existing triplet and quadruplet domain architectures successfully. Analysis on the results shows that the model not only has good stability but also has strong robustness.

Although the proposed model simulating the evolution of protein domain architectures only by fusion operation, the obtained results do contain the results of insertion, deletion, and mutation operations. For example, given that the simulation starts with "RhoGEF," if one result is "RhoGEF-PH-PH-C2" and another is "RhoGEF-PH-C2," then we can assume that the later one is the result of the fore in which the "PH" was deleted. The operations are shown in **Figure 11**.

Moreover, statistical analysis by the triplet and quadruple domain architectures derived from the evolution of the model shows that only a few domains contain a large number of immediate neighbors, and most of the domains contain only a small number of immediate neighbors, the frequency distribution of each domain neighbor in accordance with a power law (Qian

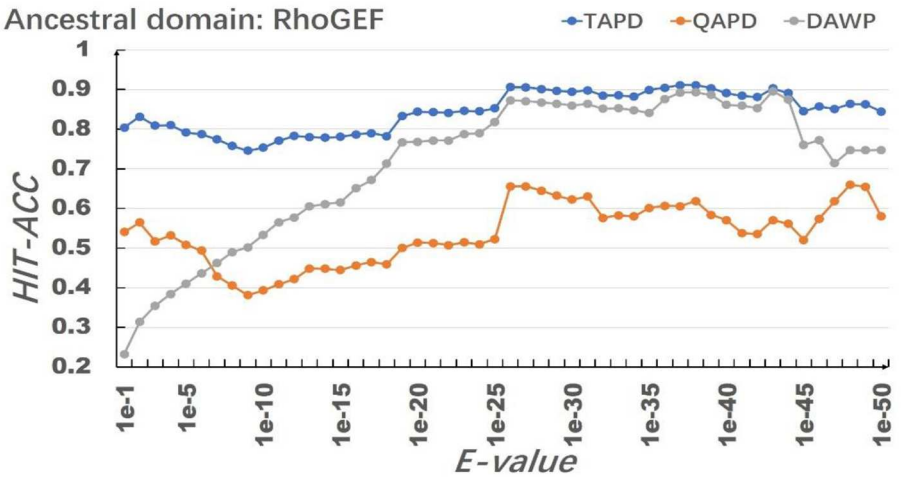

FIGURE 8 | Results of the selection of RhoGEF as an ancestral protein domain.

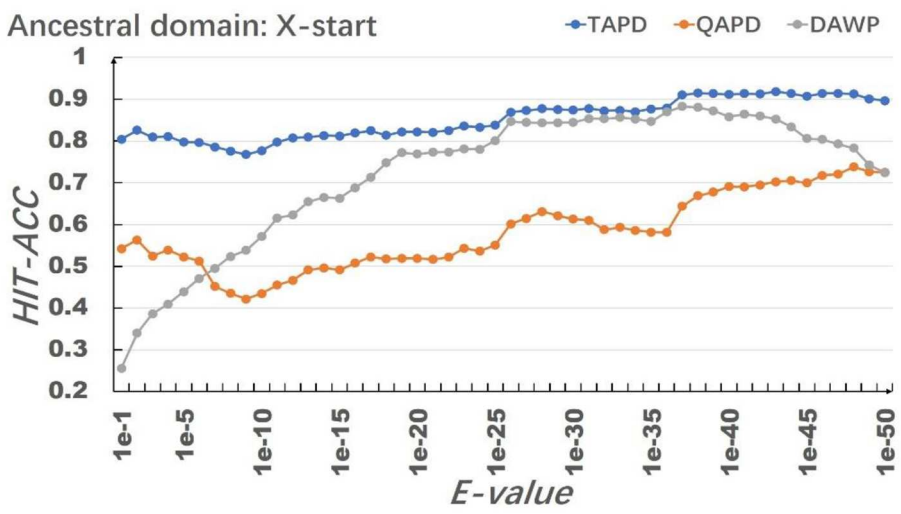

FIGURE 9 | Results of the selection of *X*-start as an ancestral protein domain.

et al., 2001). Although we used domain pair probability matrices to simulate the protein domain architecture, statistical results show that some triplets and quadruped domains are highly repetitive and appear in many sequences. In a statistical sense, these triplet and quadruple protein domain architectures are often over-expressed and highly abundant, which is consistent with the supra-domains concept (Vogel et al., 2004). This further illustrates the effectiveness of this model in simulating the evolution of protein domain architecture.

CONCLUSIONS

The goal of this work was to provide a tool that can reveal how domain architectures have evolved in protein sequences. This tool is very important in analyzing orthologous relationships between proteins in different organisms. CA has been applied in many fundamental issues in biology. Some works have already been devoted to providing a framework for simulation evolution of protein and DNA sequences. In this work, 1D probabilistic CA is used to simulate the evolution of protein domain. This model simulates the fission, fusion, deletion, and insertion of the natural evolution processes by randomly appointing transitional rules. This is the reason that our model is more in line with the natural characteristics of protein evolution. Through this website, users can know the domain architecture distribution of the submitted protein sequence and can also view the evolution results of the model to predict the evolution direction of the protein sequence file with the emergence of new and frequent protein domain architecture.

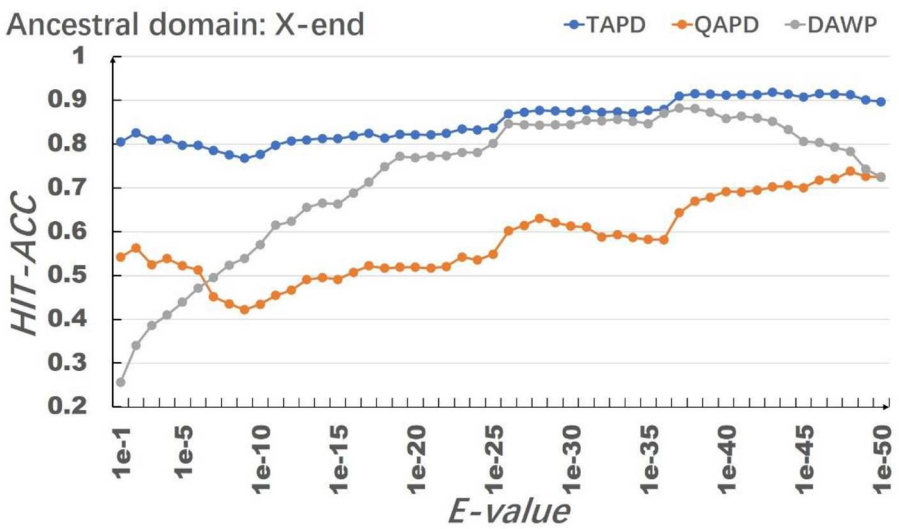

FIGURE 10 | Results of the selection of *X*-end as an ancestral protein domain.

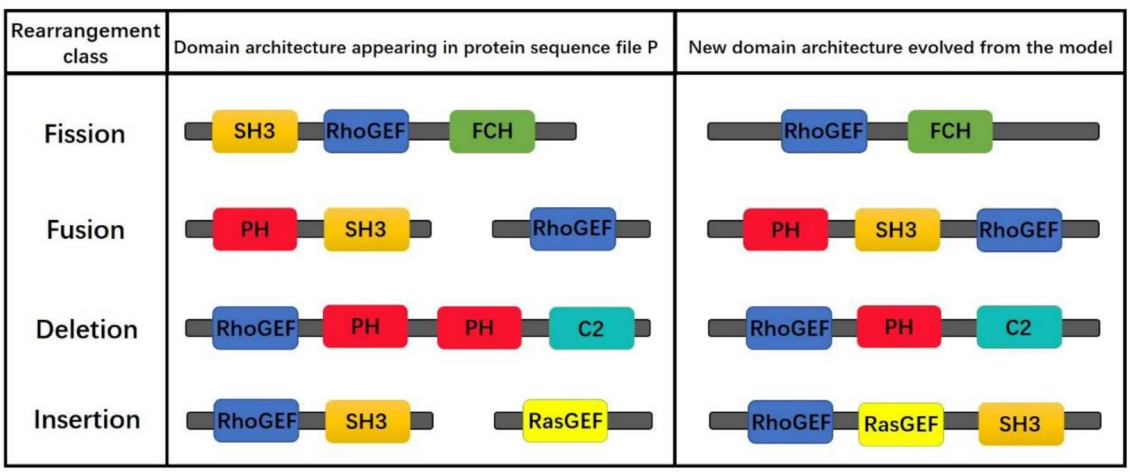

FIGURE 11 | Examples of fission, fusion, deletion, and insertion in the evolution of the model.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: access protein sequence data from NCBI, and searching HMM domains from Pfam with HMMER.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by grants from the National Natural Science Foundation of China (No. 31560316, 31860312, 31760315, 61841104), the Natural Science Foundation of Jiangxi Province, China (No. 20171ACB20023), the Department of Education of Jiangxi Province (GJJ160866, GJJ180703), and the China–Montenegro Intergovernmental S&T Cooperation (No. 2018-3-3).

REFERENCES

Adachi, J., and Hasegawa, M. (1996). Model of amino acid substitution in proteins encoded by mitochondrial DNA. *J Mol Evol*. 42, 459–468. doi: 10.1007/PL00013324

Apic, G., Gough, J., and Teichmann, S. A. (2001). Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *J. Mol. Biol*. 310, 311–325. doi: 10.1006/jmbi.2001.4776

Bakan, A., Dutta, A., Mao, W., Liu, Y., Chennubhotla, C., Lezon, T. R., et al. (2014). Evol and ProDy for bridging protein sequence evolution and structural dynamics. *Bioinformatics* 30, 2681–2683. doi: 10.1093/bioinformatics/btu336

Buljan, M., and Bateman, A. (2009). The evolution of protein domain families. *Biochem. Soc. Trans*. 37, 751–755. doi: 10.1042/BST0370751

Buljan, M., Frankish, A., and Bateman, A. (2010). Quantifying the mechanisms of domain gain in animal proteins. *Genome Biol.* 11:R74. doi: 10.1186/gb-2010-11-7-r74

Chandonia, J.-M., Fox, N. K., and Brenner, S. E. (2017). SCOPe: manual curation and artifact removal in the structural classification of proteins–extended database. *J. Mol. Biol*. 429, 348–355. doi: 10.1016/j.jmb.2016.11.023

Chopard, B., and Droz, M. (1998). *Cellular Automata Modeling of Physical Systems (Collection Alea-Saclay: Monographs and Texts in Statistical Physics)*. (Cambridge: Cambridge University Press), 122–137. doi: 10.1017/CBO9780511549755

Chothia, C. (1992). One thousand families for the molecular biologist. *Nature* 357, 543–544. doi: 10.1038/357543a0

Dawson, N. L., Lewis, T. E., Das, S., Lees, J. G., Lee, D., Ashford, P., et al. (2016). CATH: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Res*. 45, D289–D295. doi: 10.1093/nar/gkw1098

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978). "A model of evolutionary change in proteins. matrices for detecting distant relationships," in *Atlas of Protein Sequence and Structure, Nat. Biomed. Res. Found.*, ed M. O. Dayhoff (Washington DC), 345–358.

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., et al. (2019). The Pfam protein families database in 2019. *Nucleic Acids Res*. 47, D427–D432. doi: 10.1093/nar/gky995

Fong, J. H., Geer, L. Y., Panchenko, A. R., and Bryant, S. H. (2007). Modeling the evolution of protein domain architectures using maximum parsimony. *J. Mol. Biol*. 366, 307–315. doi: 10.1016/j.jmb.2006.11.017

Gumulya, Y., Sanchis, J., and Reetz, M. T. (2012). Many pathways in laboratory evolution can lead to improved enzymes: how to escape from local minima. *Chembiochem* 13, 1060–1066. doi: 10.1002/cbic.201100784

Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci*. 8, 275–282. doi: 10.1093/bioinformatics/8.3.275

Kummerfeld, S. K., and Teichmann, S. A. (2005). Relative rates of gene fusion and fission in multi-domain proteins. *Trends Genet*. 21, 25–30. doi: 10.1016/j.tig.2004.11.007

Leconte, A. M., Dickinson, B. C., Yang, D. D., Chen, I. A., Allen, B., and Liu, D. R. (2013). A population-based experimental model for protein evolution: effects of mutation rate and selection stringency on evolutionary outcomes. *Biochemistry*. 52, 1490–1499. doi: 10.1021/bi3016185

Marchler-Bauer, A., Bo, Y., Han, L., He, J., Lanczycki, C. J., Lu, S., et al. (2016). CDD/SPARCLE: functional classification of proteins via subfamily domain architectures. *Nucleic Acids Res.* 45, D200–D203. doi: 10.1093/nar/gkw1129

Prakash, A., Jeffryes, M., Bateman, A., and Finn, R. D. (2017). The HMMER web server for protein sequence similarity search. *Curr. Protoc. Bioinformatics* 60:3 15 1–3 15 23. doi: 10.1002/cpbi.40

Qian, J., Luscombe, N. M., and Gerstein, M. (2001). Protein family and fold occurrence in genomes: power-law behaviour and evolutionary model. *J. Mol. Biol*. 313, 673–681. doi: 10.1006/jmbi.2001.5079

Riley, M., and Labedan, B. (1997). Protein evolution viewed through *Escherichia coli* protein sequences: introducing the notion of a structural segment of homology, the module. *J. Mol. Biol.* 268, 857–868. doi: 10.1006/jmbi.1997.1003

Schwartz, J. T., Neumann, J. V., and Burks, A. W. (1967). Theory of self-reproducing automata. *Q. Rev. Biol*. 21:745. doi: 10.2307/2005041

Sharma, M., and Pandey, G. K. (2016). Expansion and function of repeat domain proteins during stress and development in plants. *Front. Plant Sci.* 6:1218. doi: 10.3389/fpls.2015.01218

Sirakoulis, G., Karafyllidis, I., Mizas, C., Mardiris, V., Thanailakis, A., and Tsalides, P. (2003). A cellular automaton model for the study of DNA sequence evolution. *Comput. Biol. Med*. 33, 439–453. doi: 10.1016/S0010-4825(03)00017-9

Snel, B., Bork, P., and Huynen, M. (2000). Genome evolution: gene fusion versus gene fission. *Trends Genet*. 16, 9–11. doi: 10.1016/S0168-9525(99)01924-1

Stolzer, M., Siewert, K., Lai, H., Xu, M., and Durand, D. (2015). Event inference in multidomain families with phylogenetic reconciliation. *BMC Bioinformatics* 16:S8. doi: 10.1186/1471-2105-16-S14-S8

Tuffery, P. (2002). CS-PSeq-Gen: simulating the evolution of protein sequence under constraints. *Bioinformatics* 18, 1015–1016. doi: 10.1093/bioinformatics/18.7.1015

Vogel, C., Berzuini, C., Bashton, M., Gough, J., and Teichmann, S. A. (2004). Supra-domains: evolutionary units larger than single protein domains. *J. Mol. Biol*. 336, 809–823. doi: 10.1016/j.jmb.2003.12.026

Weinreich, D. M., Delaney, N. F., DePristo, M. A., and Hartl, D. L. (2006). Darwinian evolution can follow only very few mutational paths to fitter proteins. *Science* 312, 111–114. doi: 10.1126/science.1123539

Whelan, S., and Goldman, N. (2001). A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol. Biol. Evol.* 18, 691–699. doi: 10.1093/oxfordjournals.molbev.a003851

Wiedenhoeft, J., Krause, R., and Eulenstein, O. (2011). The plexus model for the inference of ancestral multidomain proteins. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 8, 890–901. doi: 10.1109/TCBB.2011.22

Wilburn, D. B., Tuttle, L. M., Klevit, R. E., and Swanson, W. J. (2018). Solution structure of sperm lysin yields novel insights into molecular dynamics of rapid protein evolution. *Proc. Natl. Acad. Sci. U.S.A*. 115, 1310–1315. doi: 10.1073/pnas.1709061115

Xiao, X., Ye, H.-X., Liu, Z., Jia, J.-H., and Chou, K.-C. (2016). iROS-gPseKNC: predicting replication origin sites in DNA by incorporating dinucleotide position-specific propensity into general pseudo

nucleotide composition. *Oncotarget* 7:34180. doi: 10.18632/oncotarget.9057

Yang, Z. (1993). Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Mol. Biol. Evol.* 10, 1396–1401.

Zeymer, C., and Hilvert, D. (2018). Directed evolution of protein catalysts. *Annu Rev Biochem*. 87, 131–157. doi: 10.1146/annurev-biochem-062917-012034

Zhang, X.-C., Wang, Z., Zhang, X., Le, M. H., Sun, J., Xu, D., et al. (2012). Evolutionary dynamics of protein domain architecture in plants. *BMC Evol. Biol.* 12:6. doi: 10.1186/1471-2148-12-6

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.