Amino Acids (2005) 28: 29–35
DOI 10.1007/s00726-004-0154-9

Amino Acids
Printed in Austria

Using cellular automata to generate image representation for biological sequences

X. Xiao[1,2], **S. Shao**[1], **Y. Ding**[1], **Z. Huang**[1], **X. Chen**[1], and **K.-C. Chou**[1,3,4]

[1] Bio-Informatics Research Center, Donghua University, Shanghai, China
[2] Computer Department, Jing-De-Zhen Ceramic Institute, Jing-De-Zhen, China
[3] Department of Biomedical Engineering, Shanghai Jiaotong University, Shanghai, China
[4] Gordon Life Science Institute, San Diego, California, U.S.A.

Received October 11, 2004
Accepted December 14, 2004
Published online February 10, 2005;

Summary. A novel approach to visualize biological sequences is developed based on cellular automata (Wolfram, S. *Nature* 1984, 311, 419–424), a set of discrete dynamical systems in which space and time are discrete. By transforming the symbolic sequence codes into the digital codes, and using some optimal space-time evolvement rules of cellular automata, a biological sequence can be represented by a unique image, the so-called cellular automata image. Many important features, which are originally hidden in a long and complicated biological sequence, can be clearly revealed thru its cellular automata image. With biological sequences entering into databanks rapidly increasing in the post-genomic era, it is anticipated that the cellular automata image will become a very useful vehicle for investigation into their key features, identification of their function, as well as revelation of their "fingerprint". It is anticipated that by using the concept of the pseudo amino acid composition (Chou, K.C. Proteins: Structure, Function, and Genetics, 2001, 43, 246–255), the cellular automata image approach can also be used to improve the quality of predicting protein attributes, such as structural class and subcellular location.

Keywords: Cellular automata images – Data visualization – Pseudo amino acid composition – Bioinformatics

I Introduction

The success of human genome project has generated deluge of sequence information. Sequence databases, such as GenBank and EMBL, have been growing at an exponential rate (Venter et al., 1996; Chou, 2002; Chou, 2004). In general, gene sequences are stored in the computer database system in the form of long character strings. It would act like a snail's pace for human beings to read these sequences with the naked eyes. Also, it is very hard to extract any key features by directly reading these sequences. However, if they can be converted to some diagrams (see, e.g., Chou and Zhang, 1992; Zhang and Chou, 1994), some important features can automatically manifested and become easily visible.

The question of how to visualize gene sequence is an important topic today (Hu et al., 2003; Kashuk et al., 2002; Liu et al., 2002; Mayor et al., 2000; Nandy, 1996; Randic et al., 2000). Previous effort in biological sequence visualization was focused on single sequence representations. About 20 years ago, the first 3D (dimensional) H curve was proposed to represent a DNA sequence (Hamori, 1985; Hamori and Ruskin, 1983). Subsequently, a graphic representation of DNA sequences was suggested using Barnsley's iterative function (Jeffrey, 1990). Later, a different method was proposed thru the iterative function system (Roman-Roldan et al., 1994; Tino, 1999). By extrapolating the work of Hamori and Jeffrey, a different iterative method called W-curves was presented (Wu et al., 1993). Meanwhile, a diagrammatical approach for codon usage were also proposed (Chou and Zhang, 1992; Zhang and Chou, 1994). Gates (1985) proposed a 2D graphical representation that is simpler than the H curve. However, Gates's graphical representation has high degeneracy. Guo took an ulterior step and proposed a novel 2D graphical representation of DNA sequences of low degeneracy (Guo et al., 2001). In 2003, Yau presented a representation without degeneracy (Yau et al., 2003).

In parallel to the above development, various representations for protein sequences have also been proposed.

Williams et al. (1995) used five vertical spaces to represent each amino acid position, with the spaces filled according to the chemical properties of the residues. This leads to sequences resembling Morse code, with some structural features highlighted by the resulting pattern of dots. The properties of a protein's amino acids may also be visualized in the form of a line graph, for example, protein rhodopsin is showed using the hydropathic scale (Alston et al., 2003). Chou et al. (1997) first introduced the elegant ''wenxiang'' diagram to highlight the typical sequence feature of the amphiphilic helices in proteins.

There is a common characteristic in the aforementioned visual methods for the gene representation, i.e., the point of the special curve corresponding to a certain nucleic acid is colligated only with the base prior to it, while the effects of all the bases behind it are totally ignored. This is inconsistent with the fact that all the bases in a gene are coupled with each other as an entity in nature. In view of this, here a completely new and different method will be introduced to image the gene sequences. The novel method is based on Cellular Automata, as will be illustrated below.

II Methods

Cellular automata

Cellular automata are discrete dynamical systems whose behavior is completely specified in terms of a local relation. A cellular automaton can be thought of as a stylised universe consisting of a regular grid of cells, each of which can be in one of a finite number of k possible states, updated synchronously in discrete time steps according to a local, identical interaction rule (Wolfram, 1986). Cellular automata provide us an access to model complex dynamical phenomena by reformulating the macroscopic behavior into microscopic and mesoscopic rules that are discrete in space and time. A set of rules specifies the time and space evolution of the system, which is discrete in both variables. These systems have attracted a great deal of interest in recent years because even with very simple rules cellular automata can show very complex evolution patterns. It is recognized that repeated applications of simple rules can lead to extremely complex behavior that can emulate physical, social and biological systems.

A one-dimensional cellular automata consists a collection of time-dependent variables S_t^i, namely the local states, arrayed on a lattice of N sites (or cells), $i = 0, 1, 2, \ldots, N-1$. We take each of these to be a Boolean variable: $S_t^i = \{0, 1\}$. As visualization is considered in a two-state automaton, each of the cells can be either black or white. The collection of all local states is called the configuration: $S_t = S_t^0 S_t^1 \cdots S_t^{N-1}$, where S_0 denotes an initial configuration. The rule F of cellular automata can be expressed as a lookup table that lists, for each local neighborhood, the state that is taken on by the neighborhood's central cell at the next step. A neighborhood comprises a cell and its r neighbors on either side, where r is called the cellular automata radius. The course of state evolving can be represented as: $S_{t+1}^i = F(S_t^{i-r} \cdots S_t^i \cdots S_t^{i+r})$. If the r is 1, each cell can be either black or white, then this will allows $2^3 = 8$ possible color combinations along the top three cells. Because each of these combinations will cause a cell to be either black or white and there are eight possible upper color combinations then there will be $2^8 = 256$ possibilities in total. In general, if there are K states and if each cell is taken to have N neighbors (including itself), then there are K^N rules. We can easily utilize a binary byte to encode these rule sets into decimal numbers between the numbers 0 and 255. For example, rule number 184 would correspond to Fig. 1. The global equation of motion ϕ maps a configuration at one time step to the next; i.e., $S_{t+1} = F(S_t)$, where the local function ϕ is applied simultaneously to all lattice sites.

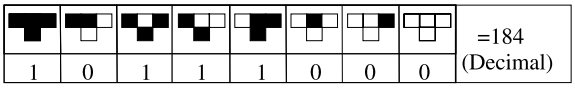

Fig. 1. Rule number 184. The string of eight zeros and ones create one binary byte, which can represent a decimal number between 0 and 255

Digital coding for amino acid and ribonucleic acid

Molecular biologists seek to determine the genes in the cells of organisms, the function of the proteins that these genes encode, and how these proteins are related evolutionarily across organisms. Genes, composed of RNA, is represented by sequences of nucleic acids, also called bases. The 4 nucleic acids are adenine(A), cytosine(C), guanine(G), uracil(U). To deal with it in a computer, a nucleotide sequence is coded as follows:

$$\mathrm{A} = 00, \quad \mathrm{C} = 01, \quad \mathrm{G} = 10, \quad \mathrm{U} = 11 \tag{1}$$

Proteins are represented by sequences of amino acids, also called residues. There are 20 native amino acids. By means of the similarity rule, complementarity rule, molecular recognition theory and information theory, a set of digital codes are formulated to represent amino acids, as shown in Table 1. The representation can better reflect the chemical physical properties of amino acids, as well as their structure and degeneracy (Xiao et al., 2004).

Space-time evolution of gene sequence

A gene sequence is always a 1D string regardless it is denoted by bases or by binary digits. It is very difficult to find its characteristic vector particularly when it is very long. To cope with this situation, we resort to the images derived from the 1D sequence thru the space-time evolution of cellular automata. The cellular automata we adopt here is a simple two-state, one-dimensional cellular automata, consisting of a line of cells with the value of 0 or 1. The rule is simply implemented as that the nearest cells around the one we focus will decide its next state. Because many genes are circular, we adopt the circulating boundary condition with the iterative formula given by:

$$D(i,j) = F(D(i-1,j-1), D(i-1,j), D(i-1,j+1)) \quad (1 \le i < n, 1 \le j < M \times N - 1) \tag{2}$$

$$D(i,0) = F(D(i-1, M\times N-1), D(i-1,0), D(i-1,1)) \quad (1 \le i < n) \tag{3}$$

$$D(i, M\times N-1) = F(D(i-1, M\times N-2), D(i-1, M\times N-1), D(i-1,0)) \quad (1 \le i < n) \tag{4}$$

where, $D(i,j)$ is an element of 2D array to present the gene sequence image, F the iterative rule, n the iterative time, and N the length of the gene sequence. If the sequence is composed of RNA, the $M = 2$; if the sequence composed of amino acids, the $M = 5$. For example, Rule 84 can be illustrated by Fig. 2.

Table 1. Binary notation of amino acid coding language

codon	amino acid	binary notation	codon	amino acid	binary notation
ccu ccc cca ccg	P	00001	cuu cuc cua cug uua uug	L	00011
caa cag	Q	00100	cau cac	H	00101
cgu cgc cga cgg aga agg	R	00110	ucu ucc uca ucg agu agg	S	01001
uau uac	Y	01100	uuu uuc	F	01011
ugg	W	01110	ugu ugc	C	01111
acu acc aca acg	T	10000	auu auc aua	I	10010
aug	M	10011	aaa aag	K	10100
aau aac	N	10101	gcu gcc gca gcg	A	11001
guu guc gua gug	V	11010	gau gac	D	11100
gaa gag	E	11101	ggu ggc gga ggg	G	11110
uaa uag uga	end	11111			

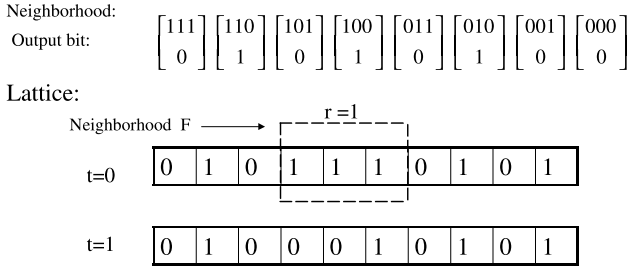

Fig. 2. Illustration of a one-dimensional, binary-state, nearest-neighbor (r = 1) cellular automata with N = 10. Both the lattice and the rule table F for updating the lattice are illustrated. The lattice configuration is shown at two successive time steps. The cellular automaton has spatially periodic boundary conditions: the lattice is viewed as a circle, with the leftmost cell being the right neighbor of the rightmost cell, and vice versa

Image generation

When transforming the 2D array (matrix) into a binary image with visualization techniques, the basic bitmap format is chosen because its property is easily handled. In this way, if the matrix element was zero, the color of the counterpart pixel bit will be black; otherwise, white.

Image compression

The total size thus obtained are too large for some long sequences, the compression of the image is needed that is actually to highlight the characteristic of the image concerned the following mathematical mapping:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{5}$$

where (x_0, y_0) denote the coordinates of the pixel in the original image, while (x_1, y_1) the corresponding coordinates for the transformed image, f_x is the scaling along the horizontal axis, and f_y the scaling along the vertical axis. The inverse transformation is given by:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1/f_x & 0 \\ 0 & 1/f_y \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{6}$$

i.e.,

$$\begin{cases} x_0 = x_1/f_x \\ y_0 = y_1/f_y \end{cases} \tag{7}$$

III Results and discussion

The images of real and simulated gene data will be presented as examples to show how these cellular automata images provide useful information. The aforementioned gene sequences are all downloaded from Genbank: http://www.ncbi.nlm.nih.gov. To the same sequence, if the evolving rules are different, the images are different. That is to say, 256 different images can be created for a same sequence based on cellular automata. These images can fall into 4 classes. The first class is named balanced, the states of cells been quickly resolved into boring configurations, e.g., all 0 or all 1. The second class is periodic. The third class is of chaos. The fourth class is not disordered, but complex and sometimes long-lived. The evolution rule of the formulation image that we need must generate the features that can be easily used to distinguish whether the gene concerned are homologous to each

other. By this way, the bases in a gene or residues in a protein must be coupled with each other as an entity. During the process of producing the gene image, the state of cell corresponding to a certain nucleic acid is colligated with both the base prior to it and bases behind it. Because of above-mentioned characteristics, the gene image can reveal some implicit sequence features, and these features are difficult to be displayed by other gene visualizations. We have found that among the 256 evolving rules some is better than the others in building gene image for a given gene. For example, Rule 184 is most suitable for coronavirus, while Rule 84 is the best for building the image of amino acid sequences.

If the rule and time for the evolution are all changeless, the gene sequence and image thus produced will be one-to-one correspondence. Because digital coding for amino acid and nucleotide are degeneracy, the images will appear in different cells for the first row at least. Figure 3 shows the comparative image between mouse TGFA gene (P01134) and its recombine gene. The recombine gene only has one difference to P01134 in the 61th amino acid, phenylalanine to lysine. The method of generating comparative image is for comparing the corresponding bit between the previously generated two pieces of images: if the color is same, the corresponding pixel point on the comparative image will be drawn in the original color; otherwise, the counterpart in the comparative image will be drawn as a red point.

Different rules have been applied to analyze the 90 coronavirus, but only when Rule 184 is used, are the images of SARS-CoVs different most distinctively from those of other coronavirus (Wang et al., 2005). The images obtained directly by the aforementioned procedures are generally too large for analysis. After the images are zoomed out with the compression ratio 14:2 as showed in Fig. 4, the images of SARS-CoVs are mainly with the V-shaped cross-lines pattern, whereas those of non-SARS

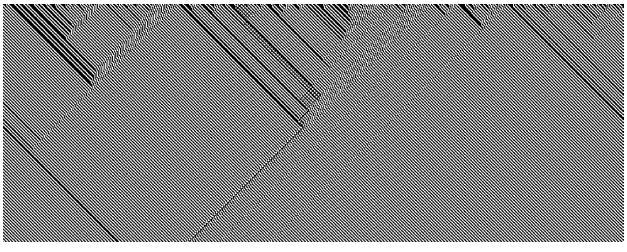

Fig. 3. Comparative image between mouse TGFA gene (P01134) and its recombine gene. The recombine gene only has one different to P01134 in 61th amino acid, phenylalanine to lysine. The Rule 84 was used for the evolutive

Fig. 4. Sample images obtained by applying the Rule 184 on the SARS coronal virus and non-SARS coronavirus: (**a**) BJ01(AY278488), and (**b**) AF208066_Murine. The time of evolving was 2400, the compression ratio is 14:2. the SARS image is with a V-shaped cross-lines pattern, a token for SARS coronal viruses; and the non-SARS coronavirus image is with a parallel slash-lines pattern, a remarkable distinction with the SARS coronal virus

virus RNA sequences are mainly with the parallel slash-lines pattern. By analyzing the different parts of the full-length RNA sequence visualized images, a remarkable fingerprint for the SARS-CoV has been found. It is in some regions of the SARS-CoV sequences near 5′-terminal (Chou et al., 1996; Zhang and Chou, 1996) that the occurrence frequencies of repeated character 'A' (i.e., 'AA', 'AAA', and 'AAAA') are obviously greater than those of repeated character 'U' (i.e., 'UU', 'UUU', and 'UUUU'), respectively. However, for all other coronaviruses, the situation is just opposite in the same region; i.e., the occurrence frequencies of 'AA', 'AAA', and 'AAAA' are obviously less than those of 'UU', 'UUU', and 'UUUU'. Therefore, such a unique feature of SARS-CoV can be defined as its fingerprint. Actually, it was found that the number of individual 'A' in the V-shape region of some SARS gene sequences is approximately equal to the number of individual 'U' according to the statistic result. These segments are from 3232 to 5624 nt, 5703 to 7195 nt, 12128 to 14470 nt, 16444 to 19231 nt, and 17928 to 21803 nt in the SARS-CoV sequence near 5-terminal. There is no such a feature in non-SARS coronaviruses, as will be elaborated elsewhere.

Besides, the gene cellular automata image also has the following features as illustrated below. Shown in Fig. 5 is the cellular automata image for a *C* gene of Hepatitis B virus (HBV) built by the Rule 84. From the figure we can see that the image of HBV *C* gene has its particular pattern and character. Because the circulating boundary condition was used, the image can be a circle when the right

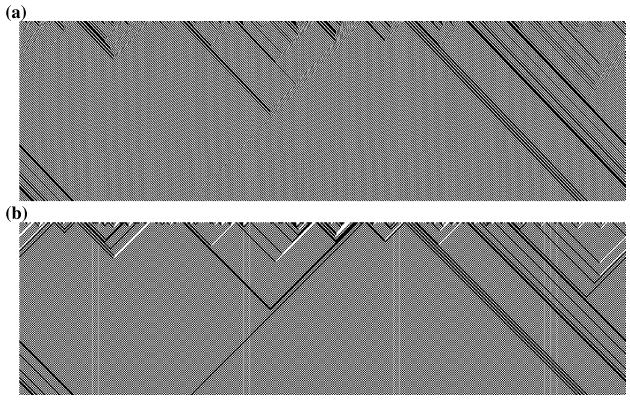

Fig. 5. The cellular automota images of Hepatitis B virus *C* gene are generated by cellular automata Rule 84: the time of evolving is 300, and the sequence is obtained from NCBI GenBank (ab059661). (**a**) The original image, and (**b**) the compressed image from (a). The compression ratio is 2:2

and left edges are connected with each other. There are two big triangular areas and three small triangular areas in the images of the figure. A lot of small triangles are nested into big triangle, and these triangles are all inverted. Therefore, the current method provides a much more intuitive and easier-to-be-identified feature for the complicated gene sequence than the original symbolic sequential expression.

Furthermore, it follows by analyzing the Rule 84 that

$$D(i,j) = \begin{cases} 0, & D(i-1,j-1)D(i-1,j) = 00 \\ \bar{x}, & D(i-1,j-1)D(i-1,j) \neq 00, D(i-1,j+1) = x \end{cases} \tag{8}$$

where $x = \{0,1\}$, and $\bar{x}$ is the inversion of x. Thus, according to Rule 84 we can derive the image for the WIAD gene (Fig. 6).

Different types of the gene sequences from the same organism were used to test the method. The TGFA and beta-globin major genes are different in their functions. Figures 7 and 9 show the two mouse genes, respectively.

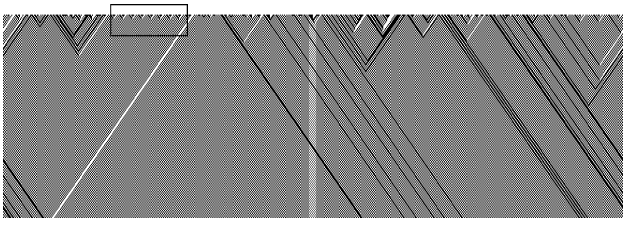

Fig. 6. The cellular automota image of WIAD gene with some periodic sections: the time of evolving is 300, and the evolving rule is the Rule 84. The compression radio is 2:2

It can be seen by comparing the two images that both images are quite different and there is no significant similarity at all. In molecular biology, there are many similarities in their functions and appearances among homology sequences. The sequences of Transforming Growth Factor-Alpha (TGFA) genes are examined. They include homo sapiens (AAA61157, AAH05308, AAH05309, CAA49806), Capreolus (AAF73229), Danio rerio (CAE30382), Sheep (P98135), Rhesus monkey (P55244), Mus musculus (AAB50554), Rabbit (P98138), Chicken (NP_001001614), Norway rat (NP_036803), and Canis familiaris (AAR21186). As shown in Figs. 7, 8, two images of human and mouse are very similar although they are from three different kinds of organisms. In other words, they do have some common features in these two sequences, which are hard to be identified from their

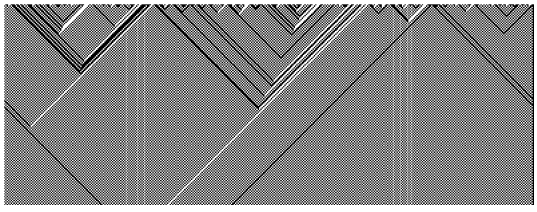

Fig. 7. Compressed image of the mouse TGFA gene. The sequence is obtained from NCBI GenBank (P01134), its length is 159 amino acids, the compression ratio is 2:2, and the time of evolving is 300

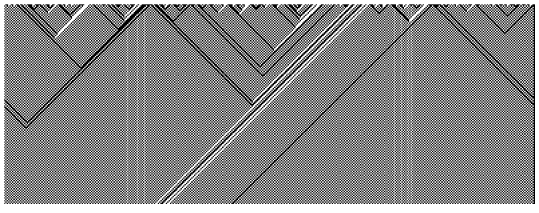

Fig. 8. Compressed image of the human TGFA gene. The sequence was obtained from NCBI GenBank (AAH05308), its length is 159 amino acids, the compression ratio is 2:2, and the time of evolving is 300

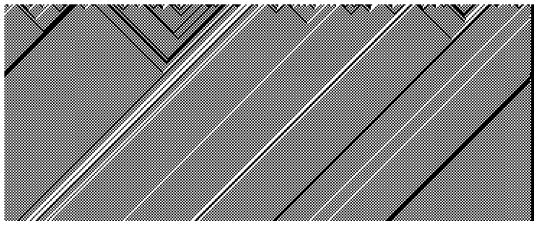

Fig. 9. Compressed image of the mouse beta-globin major gene. The sequence was obtained from NCBI GenBank (J00413), the compression ratio is 2:2, and the time of evolving is 300

symbolic sequences. These compelling results indicate that the current cellular automat approach is indeed very useful in distinguishing a special gene sequences by providing an inductive image.

Finally, it has not escaped our notice that, with the concept of the pseudo amino acid composition as originally introduced by Chou (Chou, 2001), the current cellular automata image approach can also be used to improve protein structural class prediction [see, e.g., (Chou and Zhang, 1993; Chou, 1993; Chou, 1995; Chou, 2000; Chou and Cai, 2004a; Chou and Maggiora, 1998; Chou and Zhang, 1994; Chou, 1989; Luo et al., 2002; Nakashima et al., 1986; Zhou, 1998)], protein subcellular location prediction [see, e.g., Chou and Cai, 2002; Chou and Cai, 2004b; Chou and Elrod, 1999b; Pan et al., 2003; Zhou and Doctor, 2003)], and membrane protein type prediction [see, e,g., (Cai et al., 2003; Chou and Elrod, 1999a; Wang et al., 2004a, b)], as demonstrated elsewhere (Xiao et al., 2004).

IV Conclusions

It is demonstrated thru this study that the novel method developed on the basis of cellular automata is very useful for investigating complicated biological sequences.

Acknowledgments

This work was supported in part by Doctoral Foundation from National Education Committee (20030255009), China.

References

Alston M, Johnson CG, Robinson G (2003) Colour merging for the visualization of biomolecular sequence data. Seventh International Conference on Information Visualization (IV'03), July 16–18, London, England, pp 169–175

Cai YD, Zhou GP, Chou KC (2003) Support vector machines for predicting membrane protein types by using functional domain composition. Biophys J 84: 3257–3263

Chou JJ, Zhang CT (1993) A joint prediction of the folding types of 1490 human proteins from their genetic codons. J Theor Biol 161: 251–262

Chou KC (1993) Mini review: Prediction of protein folding types from amino acid composition by correlation angles. Amino Acids 6: 231–246

Chou KC (1995) A novel approach to predicting protein structural classes in a (20-1)-D amino acid composition space. Proteins: Structure, Function and Genetics 21: 319–344

Chou KC (2000) Review: Prediction of protein structural classes and subcellular locations. Current Protein and Peptide Science 1: 171–208

Chou KC (2001) Prediction of protein cellular attributes using pseudo-amino-acid-composition. PROTEINS: Structure, Function, and Genetics (Erratum: ibid. (2001) 44: 60) 43: 246–255

Chou KC (2002) A new branch of proteomics: prediction of protein cellular attributes. In: Weinrer PW, Lu Q (eds) Gene cloning and expression technologies, chapter 4, Eaton Publishing, Westborough, MA, pp 57–70

Chou KC (2004) Review: Structural bioinformatics and its impact to biomedical science. Curr Med Chem 11: 2105–2134

Chou KC, Cai YD (2002) Using functional domain composition and support vector machines for prediction of protein subcellular location. J Biol Chem 277: 45765–45769

Chou KC, Cai YD (2004a) Predicting protein structural class by functional domain composition. Biochem Biophys Res Comm 321: 1007–1009

Chou KC, Cai YD (2004b) Prediction of protein subcellular locations by GO-FunD-PseAA predicor. Biochem Biophys Res Commun 320: 1236–1239

Chou KC, Elrod DW (1999a) Prediction of membrane protein types and subcellular locations. Proteins: Structure, Function, and Genetics 34: 137–153

Chou KC, Elrod DW (1999b) Protein subcellular location prediction. Protein Engineering 12: 107–118

Chou KC, Maggiora GM (1998) Domain structural class prediction. Protein Engineering 11: 523–538

Chou KC, Zhang CT (1992) Diagrammatization of codon usage in 339 HIV proteins and its biological implication. AIDS Res Hum Retroviruses 8: 1967–1976

Chou KC, Zhang CT (1994) Predicting protein folding types by distance functions that make allowances for amino acid interactions. J Biol Chem 269: 22014–22020

Chou KC, Zhang CT, Elrod DW (1996) Do antisense proteins exist? J Protein Chem 15: 59–61

Chou KC, Zhang CT, Maggiora GM (1997) Disposition of amphiphilic helices in heteropolar environments. Proteins: Structure, Function, and Genetics 28: 99–108

Chou KC, Wei DQ, Zhong WZ (2003) Binding mechanism of coronavirus main proteinase with ligands and its implication to drug design against SARS. (Erratum: ibid. (2003) 310: 675). Biochem Biophys Res Comm 308: 148–151

Chou PY (1989) Prediction of protein structural classes from amino acid composition. In: Fasman GD (ed) Prediction of protein structure and the principles of protein conformation, Plenum Press, New York, pp 549–586

Gates MA (1985) Simpler DNA sequence representations. Nature 316: 219

Guo XF, Randic M, Basak SC (2001) A novel 2-D graphical representation of DNA sequences of low degeneracy. Chemical Physics Letters 350(1–2): 106–112

Hamori E (1985) Novel DNA sequence representations. Nature 314: 585–586

Hamori E, Ruskin J (1983) H curves, a novel method of representation of nucleotide series especially suited for long DNA sequences. J Biol Chem 258: 1318–1327

Hu ZJ, Frith M, Niu TH, Weng ZP (2003) SeqVSTA: a graphical tool for sequence feature visualization and comparison. BMC Bioinformatics 4: 1–8

Jeffrey J (1990) Chaos game representation of gene structure. Nucleic Acids Res 18: 2163–2170

Kashuk C, SenGupta S, Eichler E, Chakravarti A (2002) ViewGene: a graphical tool for polymorphism visualization and characterization. Genome Res 12: 333–338

Luo RY, Feng ZP, Liu JK (2002) Prediction of protein strctural class by amino acid and polypeptide composition. Eur J Biochem 269: 4219–4225

Liu Y, Guo X, Xu J, Pan L, Wang S (2002) Some notes on 2-D graphical representation of DNA sequence. J Chem Inf Comput Sci 42: 529–533

Mayor C, Brudno M, Schwartz JR, Poliakov A, Rubin EM, Frazer KA, Pachter LS, Dubchak I (2000) VISTA: visualizing global DNA

sequence alignments of arbitrary length. Bioinformatics 16: 1046–1047

Nandy A (1996) Two-dimensional graphical representation of DNA sequences and intron-exon discrimination in intron-rich sequences. Comput Apple Biosci 12: 55–62

Nakashima H, Nishikawa K, Ooi T (1986) The folding type of a protein is relevant to the amino acid composition. J Biochem 99: 152–162

Pan YX, Zhang ZZ, Guo ZM, Feng GY, Huang ZD, He L (2003) Application of pseudo amino acid composition for predicting protein subcellular location: stochastic signal processing approach. J Protein Chem 22: 395–402

Randic M, Vracko M, Nandy A, Basak SC (2000) On 3-D graphical representation of DNA primary sequences and their numerical characterization. J Chem Inf Comput Sci 40: 1235–1244

Roman-Roldan R, Bernaola-Galvan P, Oliver JL (1994) Entropic feature for sequence pattern through iteration function systems. Pattern Recognition Letters 15: 567–573

Tino P (1999) Spatial representation of symbolic sequences through iterative function system. IEEE Transaction on Signal Processing 29(4): 386–393

Venter JC, Smith HO, Hood L (1996) A new strategy from genome sequencing. Nature 381: 364–366

Wang M, Yang J, Liu GP, Xu ZJ, Chou KC (2004a) Weighted-support vector machines for predicting membrane protein types based on pseudo amino acid composition. Protein Eng Des Sel 17: 509–516

Wang M, Yang J, Xu ZJ, Chou KC (2004b) SLLE for predicting membrane protein types. J Theoretical Biol 232: 7–15

Wang M, Yao JS, Huang ZD, Xu ZJ, Liu GP, Zhao HY, Wang JS, Yang J, Zhu YS, Chou KC (2005) A new nucleotide-composition based fingerprint of SARS-CoV with visualization analysis. Medicinal Chemistry 1: 39–48

Williams A, Chenault K, Melcher U (1995) Graphic representations of amino acid sequences. In: Pickover CA (ed) Visualizing biological information. World Scientific, River Edge NJ, pp 6–14

Wolfram S (1986) Cellular automation fluid: basic theory. J Stat Phys 45: 471

Wu D, Roberge J, Cork DJ, Nguyen BG, Grace T (1993) Computer visualization of long genomic sequences. IEEE Visualization 93: 308–315

Xiao X, Shao SH, Ding YS, Chen XJ (2004) Digital coding for amino acid based on cellular automata. 2004 IEEE Int. Conf. Systems, Man, and Cybernetics, Oct. 10–13, 2004, the Hague, the Netherlands (in press)

Yau SST, Wang JS, Niknejad A, Lu CX, Jin N, Ho YK (2003) DNA sequence representation with degeneracy. Nucleic Acids Res 31: 3078–3080

Zhang CT, Chou KC (1994) Analysis of codon usage in 1562 E. Coli protein coding sequences. J Mol Biol 238: 1–8

Zhang CT, Chou KC (1996) An analysis of base frequencies in the anti-sense strands corresponding to the 180 human protein coding sequences. Amino Acids 10: 253–262

Zhou GP (1998) An intriguing controversy over protein structural class prediction. J Protein Chem 17: 729–738

Zhou GP, Doctor K (2003) Subcellular location prediction of apoptosis proteins. Proteins: Structure, Function, and Genetics 50: 44–48

Ziv J, Lempel A (1976) On the complexity of finite sequences. IEEE Trans Inf Theory IT-22: 75–81

Authors' address: Prof. Shihuang Shao, Bioinformatics Research Center, Donghua University, Shanghai 200051, China; or Prof. Kuo-Chen Chou, Gordon Life Science Institute, 13784 Torrey Del Mar Drive, San Diego, CA 92130, U.S.A., E-mail: kchou@san.rr.com