CROATICA CHEMICA ACTA CCACAA **73** (4) 1123–1139 (2000)

ISSN-0011-1643
CCA-2707 *Original Scientific Paper*

Universal Metric Properties of the Genetic Code

*Nikola Štambuk**

Rudjer Bošković Institute, P. O. Box 180, HR-10002 Zagreb, Croatia

Received May 21, 2000; June 29, 2000; accepted July 31, 2000

Universal metric properties of the genetic code (*i.e.* RNA, DNA and protein coding) are defined by means of the nucleotide base representation on the square with vertices U or T = 00, C = 01, G = 10 and A = 11. It is shown that this notation defines the Cantor set and Smale horseshoe map representation of the genetic code, the classic table arrangement and Siemion one-step mutation ring of the code. Gray code solutions to the problem of defining codon positions on the [0, 1] interval, and an extension to the octal coding system, based on the linear block triple check code, are given. This result enables short block (word) decoding of the genetic code patterns. The block code is related to the minimization of errors during transcription and translation processes, which implies that the genetic code is error-correcting and not degenerate. Two algorithms for the representation of codons on the [0, 1] interval and the related binary trees are discussed. It is concluded that the ternary Cantor set algorithm is the method of choice for this type of analysis and coding. This procedure enables the analysis of the six dimensional hypercube codon positions by means of a simple time series and/or 'logistic' difference equation. Finally, a unified concept of the genetic code linked to the Cantor set and horseshoe map is introduced in the form of a classic combinatorial 4 colour necklace model with three horizontal frames consisting of 64 coloured pearls (bases) and vertically hanging decorations of triplets (codons). Three horizontal necklace frames define Crick's code without comma, and vertical necklace decorations define the evolutional code. Thus, the type of the code depends on the level or direction of observation. The exact location of the mRNA and complementary DNA coding groups of triplets within a frame is determined. The latter enables decoding of long code block (language) patterns within the genetic code. This

* (E-mail: stambuk@rudjer.irb.hr)

method of genetic code analysis is named Symbolic Cantor Algorithm (SCA). The validity of the method was confirmed by 94% accurate classification of 50 proteins of known secondary structure (25 α-helices and 25 β-sheets) with the C5.0 machine learning system. Nucleotide strings of proteins transcribed by SCA were used for the analysis. Spectral Fourier analysis of Pro-opiomelanocortin and Bone Morphogenetic Protein 6 confirmed that the method might be also applied to the analysis of bioactive hormone and cytokine sequences.

Key words: Cantor set, symbolic dynamics, SCA, Gray code, genetic code, necklace, protein, secondary structure, C5.0, machine learning, spectral analysis.

INTRODUCTION

The protein coding and synthesis in biological systems is, along with all other information of the genome, found in DNA and RNA strings consisting of 4 nucleotide base combinations (U or T, C, A and G).[1–3] Four bases define 64 codon triplets that specify 20 amino acids and 3 stop codons for the protein synthesis.[1–3] The aim of this paper is to define the universal metric properties of the codon and nucleotide base recombination. This will be done by addressing three dimensions of the problem, as follows.

First, we show that the quadratic binary representation of the 4 bases on the unit square maps all codons and amino acids to the Cantor set binary addresses on the unit interval. It is proved that, for the one-dimensional projection, symbolic binary coordinates provide a reflected Gray code solution to the problem of Hamming distance minimization of the clear binary text addresses (representing nucleotide base and amino acid positions on the tree algorithm). The underlying coding system is shown to be based on a linear block triple check code. It is speculated that this ensures accurate transcription and translation of the strings.

Second, we show that the Smale horseshoe map representation of binary blocks with fixed Cantor set codon or amino acid positions defines the classic table of the genetic code. This result indicates that the syntax of nucleotide and protein strings is based on the rich dynamical linguistic structures generated by means of the map that has an invariant set. Orbits of the map are represented by the space of symbols, *i.e.* symbolic dynamics, and are used for the analysis of the system.

Third, we show that a classic combinatorial 4 colour necklace problem,[4] with each colour representing a nucleotide base projection on the unit square, defines the unified concept of the genetic code. Reflected Gray code was used to define proper arrangement of codons within the frames of automa-

ton. Three horizontal frames of the necklace, consisting of 64 coloured pearls (bases), make Crick's comma-less code and vertically hanging decoration triplets (codons) define the evolutional code. Thus, the necklace model defines both concepts, depending on the level of the observation and/or position of the observer.

This method of coding notation and analysis is named Symbolic Cantor Algorithm (SCA). Machine learning classifier C5.0 and Fourier spectral analysis of nucleotide strings transformed by SCA define accurately the protein secondary structure folding types and functional properties of one hormone and growth factor.

RESULTS AND DISCUSSION

Metric of the Unit Interval – First Dimension

The Notation

We introduce the binary representation of 4 nucleotide bases on the square with vertices 00, 01, 10, 11 in the manner defined for the Cantor set by H. Steinhaus in 1917 (when discussing interesting properties of the set noticed by S. Banach).[3,7] The notation U or T = 00, C = 01, G = 10 and A = 11 is presented in Figure 1. It has the following properties:

The combination of 2 digits (0 or 1), denoting primary and secondary characteristics of the nucleotide bases, describes each of the letters according to the group subdivision/discrimination principles. The first digit defines the primary chemical caracteristic as a type of the base ring, *i.e.* pyrimidine is coded by 0 and purine by 1 (Figure 1). The second digit defines the secondary chemical characteristic of the ring according to the keto group (0) or amino group (1) coding. Keto group possessing pyrimidine base U or T = 00 is discriminated from the amino group bearing pyrimidine base C = 01 by the second digit notation. Full complementarity in obtaining amino group purines (A) and keto group purines (G) is achieved by symmetrical $0 \leftrightarrow 1$ ring and group changes (to A = 11, G = 10), or *vice versa*. The patterns of Figure 1 define the Siemion mutation ring and physical-chemical characteristics of the amino acids[3] in a manner analogous to the particular type of deterministic finite automaton (DFN).

Codon Positions on the Binary Tree

Table I shows the binary notation for all 64 codons and 20 amino acids. To define more precisely the positions of particular codon intervals of the binary tree with respect to the quadratic base mapping we examine the in-

variant Cantor set **C** with the method of symbolic dynamics in a standard manner.[3,8,9] This was performed because the Cantor set possesses two properties related to the binary coding of the Figure 1 notation:[3,8,9]

1. Binary decomposition of the initial segment into 2^n segments is projected on the $(n-1)^{th}$ binary tree level;

2. Partitioning of the observed set **C**, by excluding 1/3 length of its mapping interval at each tree level, may be defined by (0, 1) coin tossing, and set **C** splitting into two halves. Half of the set **C** codons are coded by the left 1/3 of the interval as 0 and the other half by the right 1/3 as 1, provided that the bifurcation of the set takes place at tossing outcome 1 with 1/2 probability. When the outcome is 0, the splitting does not take place.

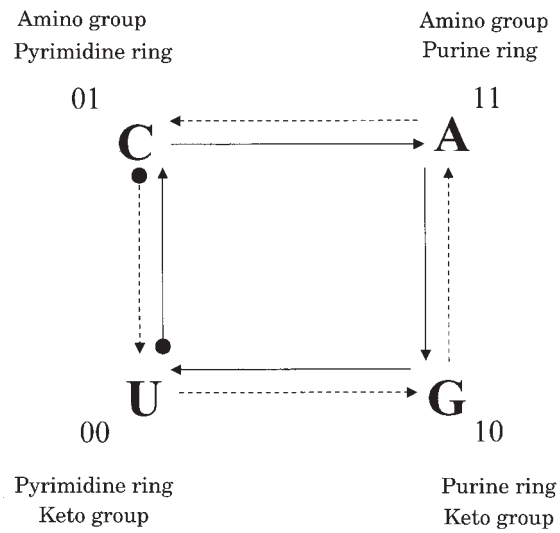

Figure 1. Binary notation of the 4 nucleotide bases based on the purine-pyrimidine ring and amino-keto group coding principles.

This process of bifurcation is determined by two universal parameters of fixed numerical value, discovered by M. Feigenbaum (α = 2.5029... and δ = 4.6692....). α is linked to the clustering of the elements (codons/amino acids) on the binary tree with respect to the bifurcation cycles that partition the total set. δ is the universal measure that defines how these elements (codons/amino acids) of the stable cycles periodically bifurcate from the origin to obtain the partition pattern.[5,6,9,10]

TABLE I

Binary and symbolic notation with the Cantor set ternary addresses of RNA, DNA and amino acids

aa ↓	codon ↓	Cantor address	binary notation	*symbolic notation (reflected Gray code)*		aa ↑	codon ↑	Cantor address	binary notation	*symbolic notation (reflected Gray code)*	
F	UUU	0.0000	00 00 00	***F***	***UUU 00 00 00****	K	AAA	0.9986	11 11 11	*V*	*GUU 10 00 00*
F	UUC	0.0027	00 00 01	***F***	***UUC 00 00 01****	K	AAG	0.9959	11 11 10	*V*	*GUC 10 00 01*
L	UUG	0.0082	00 00 10	***L***	***UUA 00 00 11***	N	AAC	0.9904	11 11 01	*V*	*GUA 10 00 11*
L	UUA	0.0110	00 00 11	***L***	***UUG 00 00 10***	N	AAU	0.9877	11 11 00	*V*	*GUG 10 00 10*
S	UCU	0.0247	00 01 00	***S***	***UCG 00 01 10***	R	AGA	0.9739	11 10 11	*A*	*GCG 10 01 10*
S	UCC	0.0274	00 01 01	***S***	***UCA 00 01 11****	R	AGG	0.9711	11 10 10	*A*	*GCA 10 01 11*
S	UCG	0.0329	00 01 10	***S***	***UCC 00 01 01****	S	AGC	0.9657	11 10 01	*A*	*GCC 10 01 01*
S	UCA	0.0357	00 01 11	***S***	***UCU 00 01 00***	S	AGU	0.9630	11 10 00	*A*	*GCU 10 01 00*
C	UGU	0.0741	00 10 00	***Y***	***UAU 00 11 00****	T	ACA	0.9246	11 01 11	*D*	*GAU 10 11 00*
C	UGC	0.0768	00 10 01	***Y***	***UAC 00 11 01****	T	ACG	0.9218	11 01 10	*D*	*GAC 10 11 01*
W	UGG	0.0823	00 10 10	***ochre***	***UAA 00 11 11***	T	ACC	0.9163	11 01 01	*E*	*GAA 10 11 11*
opal	UGA	0.0850	00 10 11	***amber***	***UAG 00 11 10***	T	ACU	0.9136	11 01 00	*E*	*GAG 10 11 10*
Y	UAU	0.0988	00 11 00	***W***	***UGG 00 10 10***	I	AUA	0.8999	11 00 11	*G*	*GGG 10 10 10*
Y	UAC	0.1015	00 11 01	***opal***	***UGA 00 10 11****	M	AUG	0.8971	11 00 10	*G*	*GGA 10 10 11*
amber	UAG	0.1070	00 11 10	***C***	***UGC 00 10 01****	I	AUC	0.8916	11 00 01	*G*	*GGC 10 10 01*
ochre	UAA	0.1097	00 11 11	***C***	***UGU 00 10 00***	I	AUU	0.8888	11 00 00	*G*	*GGU 10 10 00*
L	CUU	0.2222	01 00 00	***R***	***CGU 01 10 00****	E	GAA	0.7764	10 11 11	*S*	*AGU 11 10 00*
L	CUC	0.2250	01 00 01	***R***	***CGC 01 10 01****	E	GAG	0.7737	10 11 10	*S*	*AGC 11 10 01*
L	CUG	0.2305	01 00 10	***R***	***CGA 01 10 11***	D	GAC	0.7682	10 11 01	*R*	*AGA 11 10 11*
L	CUA	0.2332	01 00 11	***R***	***CGG 01 10 10***	D	GAU	0.7654	10 11 00	*R*	*AGG 11 10 10*
P	CCU	0.2469	01 01 00	***Q***	***CAG 01 11 10***	G	GGA	0.7517	10 10 11	*K*	*AAG 11 11 10*
P	CCC	0.2497	01 01 01	***Q***	***CAA 01 11 11****	G	GGG	0.7490	10 10 10	***K***	***AAA 11 11 11***
P	CCG	0.2551	01 01 10	***H***	***CAC 01 11 01****	G	GGC	0.7435	10 10 01	***N***	***AAC 11 11 01****
P	CCA	0.2579	01 01 11	***H***	***CAU 01 11 00***	G	GGU	0.7407	10 10 00	***N***	***AAU 11 11 00****
R	CGU	0.2963	01 10 00	***P***	***CCU 01 01 00****	A	GCA	0.7023	10 01 11	***T***	***ACU 11 01 00***
R	CGC	0.2990	01 10 01	***P***	***CCC 01 01 01****	A	GCG	0.6996	10 01 10	***T***	***ACC 11 01 01****
R	CGG	0.3045	01 10 10	***P***	***CCA 01 01 11***	A	GCC	0.6941	10 01 01	***T***	***ACA 11 01 11****
R	CGA	0.3073	01 10 11	***P***	***CCG 01 01 10***	A	GCU	0.6914	10 01 00	***T***	***ACG 11 01 10***
H	CAU	0.3210	01 11 00	***L***	***CUG 01 00 10***	V	GUA	0.6776	10 00 11	***M***	***AUG 11 00 10***
H	CAC	0.3237	01 11 01	***L***	***CUA 01 00 11****	V	GUG	0.6749	10 00 10	***I***	***AUA 11 00 11***
Q	CAG	0.3292	01 11 10	***L***	***CUC 01 00 01****	V	GUC	0.6694	10 00 01	***I***	***AUC 11 00 01****
Q	CAA	0.3320	01 11 11	***L***	***CUU 01 00 00***	V	GUU	0.6666	10 00 00	***I***	***AUU 11 00 00****

aa = amino acids; U = T
Bold italics denote the Gray code solution, asterisk (*) for 2 digit moves.

The relative location of different coding intervals and their orientation are additionally specified in Table I by the nodes of alternating binary tree and their symbolic coordinates (names).[3,8–10] Briefly, the left half of the unit interval is labelled 0 and the right one 1. For $x < 1/2$ and its derivative $f'_\lambda(x) > 0$, with quadratic map $f_\lambda(x) = \lambda x(1–x)$, $\lambda > 4$, the pairs of the initial binary tree preserve orientation and for $x > 1/2$, $f'_\lambda(x) < 0$ they reverse orientation in the alternating binary tree.[3,8–10]

Algorithms and Metric

The metric of the symbol space on the unit interval defines each number $\boldsymbol{c} \in \mathbf{C}$ in the ternary expansion[10,11] $\boldsymbol{c} = \Sigma j_n/3^n$, with $j_n = 0$ for coin tossing outcome 0 and $j_n = 2$ for outcome 1, $n = 1, 2, 3\ldots\infty$. The number $\boldsymbol{c}$ of each binary address is defined on the middle-third Cantor set of the [0, 1] interval for points r_n and s_n, as discussed by Milnor and Robinson.[10–12] The total length of the interval $\boldsymbol{P_c} = \Sigma \boldsymbol{p_c} \to 1$ for $n = 1, 2, 3\ldots\infty$ and $\boldsymbol{p_c} = \Sigma \mid s_n - t_n \mid /3^n$ with $j_n = \mid s_n - t_n \mid$ defines the maximum precision of the algorithm at each of n tree levels.[10,12] This algorithm is based on the 3^{-n} metric that makes the so called cylinder sets into balls.[10] The metric distance[10,11] is $d\ (r,s) = \Sigma \mid r_n - s_n \mid /3^n$, $n = 0, 1, 2\ldots\infty$. We denote this algorithm as a Symbolic Cantor Algorithm (SCA).[3,9,13]

As shown in Table II, the binary algorithm based on the 2^{-n} metric ($\boldsymbol{c} = \Sigma\ j_n/2^n, j_n = 0$ or 1) converges more slowly to the maximum probability $\boldsymbol{P_c} = \Sigma \boldsymbol{p_c} \to 1$, sufficient to describe the system of 2^n hypercube vertices with acceptable accuracy.[9] The latter algorithm is more often applied in the algorithmic information theory.[14] It is related to baker map[15] and the 7 digit Hamming's code,[16] since at digit $n = 6$ it does not satisfy the informational coverage of >99% of the [0, 1] interval needed for the accurate (hypercube) system description (Table II). Contrary to the binary, the Cantor set based algorithm covers, by means of the 6 digit words, a sufficient proportion of the interval to obtain >99% accuracy (Table II). Therefore, this metric enables data analysis by means of linear block triple-check code.[16] Six digits

TABLE II

Efficacy of two algorithms that definine the information of hypercube address mapping on the [0, 1] interval

Algorithmic defining of [0, 1] interval	digit no. 1	digit no. 2	digit no. 3	digit no. 4	digit no. 5	digit no. 6
Cantor (SCA) address	0.666	0.888	0.963	0.988	0.996	0.999
Binary address	0.500	0.750	0.875	0.938	0.969	0.984

are also more appropriate for describing economically the two digit specified base triplets that code for the amino acids and stop codons (Figure 1, Table I).

SCA Defines the Triple Check Code

RNA and DNA strings represent the message divided into code words of fixed digit length $n = 6$ due to the fact that two binary digits define each base of the codon word or block of fixed length $m = 3$. As shown in Table II, the previously discussed Cantor set based algorithm (SCA) ensures >95% accuracy in the informational coverage of the message for the first three bits, which indicates that the three remaining bits may be applied for error correction. The code that corresponds to this condition is a triple check linear block code. It has a the block length 6 (n), rank 3 (m) and rate 1/2 (m/n).[16]

The code is constructed as follows. The message is divided into blocks of 3, say *'abc'*, where each of a, b and c is 0 or 1. Three check bits *'xyz'*, also 0 or 1, are added. Three conditions are satisfied for the word *'abcxyz'*:

1. The number of 1*s* in *abx* is even,
2. The number of 1*s* in *acy* is even,
3. The number of 1*s* in *bcz* is even.

So, if $abc = 110$, then $x = 0$, $y = 1$, $z = 1$ and the code word is 110011.

The standard array of the code is given in Table III. The top row of the 8 × 8 table is constructed from 8 possible *'abc'* combinations,[16] and weights are sorted according to the SCA.[3,9,13] Row or coset leaders are chosen to be of the smallest possible letter weight changes according to the SCA.[3,9,13]

TABLE III

Standard array for the triple check code reconstructs the genetic code table and Siemion mutation ring of the code[3,31] by means of the algorithm presented in Figure 1. Detailed analysis of the Siemion mutation ring transformations is found in Štambuk.[3]

000000	**001011**	**010101**	**011110**	**100110**	**101101**	**110011**	**111000**
000001	001010	010100	011111	100111	101100	110010	111001
000011	001000	010110	011101	100101	101110	110000	111011
000010	001001	010111	011100	100100	101111	110001	111010
000100	001111	010001	011010	100010	101001	110111	111100
000101	001110	010000	011011	100011	101000	110110	111101
000111	001100	010010	011001	100001	101010	110100	111111
000110	001101	010011	011000	100000	101011	110101	111110

Once the heads of each column and row leaders have been chosen, the rest of the words is determined by adding the code word at the head of each column to the row leader. The adding for each digit is performed as follows: 1 + 0 = 1, 0 + 1 = 1, 0 + 0 = 0 and 1 + 1 = 0. The error correction within the standard array of the code is achieved by replacing any received word by the code word at the head of each column. The linear triple check code has 8 code words and it is quite a good error correcting code.[16] It has minimum distance $d = 3$, the Hamming bound gives the maximal possible size for such a code as $2^6/|D_1| = 2^6/7$ and the Gilbert-Varshamov bound says that a code of size $2^6/|D_2| = 2^6/22$ exists.

The genetic code table is reconstructed by the standard array of the triple check code, which confirms that this type of code is the most appropriate one for the analysis of DNA and RNA strings. Octal, *i.e.* 8×8 codon structuring within the code table is also confirmed by the horseshoe mapping in Table IV and the necklace model of the genetic code.[9,13]

TABLE IV

Horseshoe map representations of 6 digit Cantor set addresses by means of 2 binary triplets, or 2 octal numbers. Classic genetic code patterns[3] are extracted and the related codon mapppings are defined by means of the unit square transformations (Figure 1).

Base position	1st/2nd	1st/2nd	1st/2nd	1st/2nd	1st/2nd	1st/2nd	1st/2nd	1st/2nd	Base position
3rd/2nd†	000.	100.	110.	010.	011.	111.	101.	001.	†2nd/3rd
↑↓	U→	G→	A→	C→	←C	←A	←G	←U	↑↓
U C .100	**S**	A	**T**	**P**	**H**	**N**	D	**Y**	*A U*
C C .101	**S**	A	**T**	**P**	**H**	**N**	D	**Y**	*A C*
A C .111	**S**	A	**T**	**P**	**Q**	**K**	E	**ochre***	*A A*
G C .110	**S**	A	**T**	**P**	**Q**	K	E	**amber***	*A G*
G U .010	**L**	V	**M****	**L**	**R**	R	G	**W**	*G G*
A U .011	**L**	V	**I**	**L**	**R**	R	G	**opal***	*G A*
C U .001	**F**	V	**I**	**L**	**R**	S	G	**C**	*G C*
U U .000	**F**	V	**I**	**L**	**R**	S	G	**C**	*G U*

* Stop codons, **start.
† Follows 1st/2nd base to obtain codon addresses of Table I (Gray code solution is bold).

Gray Code Solution to the Metric Problem

Symbolic coordinates of codon and amino acid locations on the Cantor set in Table I represent the reflected Gray code solution to the $n = 6$ digit binary notation for $2^n = 64$ codons. This result was published in 1972 by M. Gardner for the Chinese ring puzzle solution,[17] but the solution to our problem of coding is identical.[9,13] Each one of the $n = 6$ rings that have to be freed from the double bar in a minimal number of moves represents a digit.[17] Gardner's Gray numbers that solve the puzzle in 42 moves for $n = 6$ digits/rings are symbolic addresses of different codons in Table I (bold italic letters). If we assume that for each move two rings or digits are moved simultaneously at both ends of the bar, the puzzle is solved in 31 moves (denoted by asterisks in Table I). The Cantor set solution to this problem represents codon projection to the [0, 1] interval according to their addresses on an invariant set $\mathbf{C}$.[9,13]

The addresses of the closest $\boldsymbol{c} \in \mathbf{C}$ are obtained by means of the Hamming distance minimization of the hypercube Hamiltonian paths,[17] mapped by means of the SCA to the [0, 1] interval.[3,9,13] The unit interval Cantor mapping in Table I solves the complementary coding problem *via* binary tree codon projection, and the Gray code solution requires at least 32 binary numbers from the first part of Table I. Complementary addresses for the second half of the table are symmetrically arranged at opposite binary Cantor positions and obtained by $0 \leftrightarrow 1$ digit switch.

Our result represents the optimization of R. Swanson's Gray code notation.[3,9,13,18] According to Swanson,[18] coding addresses are obtained by simple summation of the square Gray code positions: U or T = 00, C = 01, A = 10, G = 11. For the reflected Gray code, which is the most economic one, the addresses are obtained from the binary notation U or T = 00, C = 01, G = 10, A = 11 by the following transformation. Start with the digit at the right and consider each digit in turn. If the next digit to the left is even (0), let the former digit stand, and if it is odd (1), change the former digit.[9,13,17,19] It is assumed that the digit at the extreme left has 0 at its left and therefore remains unchanged (Table I).

We investigated the secondary protein structure by means of the Quinlain C5.0 classifier, which is the outgrowth of the classic C4.5 machine learning system.[20,21] The nucleotide sequences of 25 α-helix and 25 β-sheet proteins were retrieved from Barton's JPred database[22] according to their alphabetical appearance. The Cantor set symbolic addresses listed in Table I were assigned to each protein. Table V shows that SCA enables the decision rules based prediction of protein secondary structure with 94% accuracy, from the descriptive statistics codon parameters. The accuracy of the procedure rises to 100% with the 10 boosting trial option. An almost identi-

cal result is obtained if the triple check code octal notation from Table III is applied. This precision of SCA is due to the fact that stretching and folding of the quadratic map with symbolic dynamics on the unit interval[3,8–13] keeps track and information of the hypercube codon (amino acid) representations of the string by means of the reflected Gray code.[9,13,17,19] Two dimensional representation is defined *via* the horseshoe map.[9–13]

TABLE V

Decision tree and rules for defining α and β protein folding types by means of Quinlan's C5.0 machine learning classifier

```
Read 50 cases A = α-helix, B = β-sheet

Decision tree:

Skewness > -0.2087285: A (8)
Skewness <= -0.2087285:
:...Notriplets > 50: B (9)
     Notriplets <= 50:
     :...Range <= 0.9877: B (2)
          Range > 0.9877:
          :...Minimum > 0:
               :...StdErr <= 0.03485898: B (6/1)
               :    StdErr > 0.03485898: A (4)
               Minimum <= 0:
               :...Notriplets > 45: A (5)
                    Notriplets <= 45:
                    :...Notriplets > 38: B (8/1)
                         Notriplets <= 38:
                         :...Range <= 0.9959: B (3/1)
                              Range > 0.9959: A (5)


Evaluation on training data (50 cases):

          Decision Tree           Rules
          ─────────────           ─────────────
          Size     Errors         No      Errors
            9      3(6.0%)         9     3(6.0%)  <<


           (a)     (b)      <-classified as
           ──      ──
           22       3       (a): class A
                   25       (b): class B


Rule utility summary:

          Rules             Errors
          ─────             ──────
            1-2             16(32.0%)
            1-4             10(20.0%)
            1-5             10(20.0%)
            1-7              6(12.0%)

Time: 0.1 secs
```

Horseshoe Map – Second Dimension

Smale Horseshoe Map

The Smale horseshoe map is an example of diffeomorphism $f : S^2 \to S^2$, or from $\mathbf{R}^2$ to itself, that has an invariant set which is a Cantor set.[8–13] The map is closely related to the map $f_\lambda(x) = \lambda x(1–x)$ on $\mathbf{R}$ for $\lambda > 4$.[8–13] It is one of the important examples with complicated and chaotic behaviour. The horseshoe map often behaves like a skeleton on which chaotic and periodic orbits of the system are organized.[8,9] The horseshoe is the mapping of the unit square in Figure 1, which contracts the horizontal directions, expands in the vertical direction, and then folds.[8–11] The mapping is only defined on the unit square while points that leave the square are ignored.[8,10] Forward and backward iterations of the horseshoe map generate the locations of the periodic points.[8–13,15]

Amino Acid and Codon Horseshoe Mapping

By iterating the map, we specifiy the locations of periodic orbits of the codons and amino acids within the homoclinic tangle of the horseshoe. Table IV gives the labelling scheme for horizontal and vertical branches from a pair of alternating binary trees. The projections of 2 binary triplets (or 2 octal numbers) according to the horseshoe pattern extract the standard table of the genetic code, which proves that this map defines the patterns of the codon recombination buried in the code. Patterns of the first, second and third base changes also satisfy and confirm the standard square notation with 4 binary addresses presented in Figure 1, typical of the horseshoe map. The algorithm in Figure 1 is therefore confirmed for the genetic code, and the horseshoe map in Table IV represents its proper labelling scheme for the codon and amino acid positions. The octal horseshoe map in Table IV is confirmed by the column leader positions of the triple check code in Table III.

Since the invariant horseshoe set is a product of two Cantor sets intersections in horizontal and in vertical directions,[8–12] the Cantor set projection of the genetic code is also proved for a two-dimensional case.

Map orbits in a space of symbols may be analyzed with respect to bifurcation, stability and resonance.[23] Table VI shows the results of the molecular resonant analysis of the string spectra by means of Discrete Fourier Transformation (DFT).[24] Human Pro-opiomelanocortin (POMOC) and Bone Morphogenetic Protein6 precursor (BMP-6) sequences were retrieved from the NR and SWISS-PROT databases. Their Gray code spectra obtained by SCA were analyzed with the software STATISTICA® (version 5.0). The resonant peaks of the spectral analysis of POMOC in Table VI predicted all bioactive peptides and hormones that are cleaved from the precursor mole-

TABLE VI

Bioactive sequences of human Pro-opiomelanocortin and Bone morphogenetic protein 6 precursor (BMP-6) determined experimentally and by a spectral (single series) Fourier analysis

	BIOACTIVE REGIONS (amino acids no.)			
	Pro-opiomelanocortin (POMC)		BMP-6 (BPC consensus)	
	Experimental	Periodogram values*	Experimental	Periodogram values*
γ-MSH	77–87	86	16–18	14
ACTH	138–176	138, 154, 156, 170, 174	24–29	22
α-MSH	138–150	138	86–88	82
Lipotropin-γ	179–234	182	122–128	128
β-endorphin	237–261	240	141–146	142
Met-enkephalin	237–241	240	339–348	348

* Same for the triple check code (Table III).

cule. The resonant analysis of BMP-6 precursor extracted the bioactive parts of this molecule that correspond to the consensus BPC-157 gut peptide.[25] This confirms the previously reported data regarding the structure of both peptides and explains the similarity in their protective effects on different tissues.[25–27] Resonant peaks obtained by means of the 8 column leaders of the triple check code (Table III) do not differ from the resonant analysis performed with all 64 codons (Table I), which confirms the octal nature of the code. Resonant Recognition Method and Molecular Recognition Theory might enable, with respect to our notation, extraction of bioactive protein parts and their complementary receptors from DNA and RNA sequences.[3,9,27]

Necklace Model – Third Dimension

Circular Code Arithmetic and Necklace Coding

We defined the genetic and protein circular code by means of a combinatorial necklace model.[13] This structure consists of 64 beads of 4 different colours representing 4 nucleotide bases (U or T, C, A, G). The coloured beads make decorations that consist of vertically hanging chains of $x = 3$ beads, which represent each of the codons. Consequently, there are $y = 4^3$ distinct vertical chains that can be made (*i.e.* the number of words of length $x = 3$

with the alphabet of size $y = 4$). The total number of possible vertical decorations containing at least two colours each is $y^x - y = 60$, and $y = 4$ decorations contain beads of the same colour. The arrangement of the codons in three frames according to their projection on the Cantor set transforms each frame in such way that if one letter shift is performed, the next frame is automatically retrieved.[13] This result was obtained for DNA and tRNA.[13] Gray code arrangement of the complementary frames that code for mRNA and complementary DNA, beginning from the start AUG, *i.e.* Methionine, codon is presented in Table VII. $0 \leftrightarrow 1$ digit switches, and *vice versa*, define the arrangements of the complementary DNA and RNA sequences.[13]

Necklace model of the genetic code extracts three frames of the automaton that prints the protein according to the Gray code based error minimization procedure. It remains an open question whether or not the decoding procedure of protein coding, *i.e.* Gray code protein printing, regions might enable the location of the corresponding DNA and RNA programming language structures within non-coding genome regions.[9,13]

CONCLUDING REMARKS

Presented results indicate that the concepts of the code without comma and of the evolutionary code, based on different premises, strongly depend on the level of the observation (analysis). In the necklace model, Crick's code without comma[1,2,13,28] represents three horizontal frames that define necklace chains, while Dounce's evolutionary code[1,2,13,29] makes vertically hanging beds (codon triplets). Therefore, the circular coding necklace algorithm represents a unifying concept for the genetic code.[13] Its structure has a striking resemblance to the Enigma coding device.[30] Knowledge of the binary-Gray code relations and codon positions within three automaton frames opens new possibilities for the genome software analysis.

Symbolic Cantor Algorithm enables the genetic code and protein analysis *via* the number theory arithmetic for codes in several dimensions, depending on the code type. Two dimensional Cantor set projection of the binary (square) notation of the Smale horseshoe map reconstructs the classic table of the genetic code, which proves our result and opens the possibility of the gene and protein analyses as chaotic dynamical systems. The genetic code table is also contained in the one dimensional Cantor set projection of the six dimensional hypercube vertices.[3,9,13] It is worth mentioning that the one-dimensional SCA based mapping enables the analysis of any six-dimensional hypercube system as a time series, providing that the proper coding of elements is performed.

TABLE VII

Messenger RNA and complementary DNA frames of the combinatorial necklace model of the genetic code previously defined by Štambuk for DNA and tRNA coding structures.[13] In horizontal directions, we observe circular coding patterns of 3 necklace frames that make Crick's comma-less code, while vertical directions define 64 hanging codons of the evolutional code (a-d). Arrangements of codons in frames according to their projection on the Cantor set and Gray code (symbolic dynamics) transform each frame in such a way that when a one letter shift is performed the next frame is automatically retrieved (a-d). Amino acids in the second and third frames (m_2, m_3) may be also generated from the first one (m_1) due to the fact that endpoints of frame 1 enable automatic one letter shifts when the end of the frame with regard to codon triplets is reached (e).

a)

$m_1 \rightarrow$ *M* AUG	*M* AUG	*K* AAG	*E* GAG	*D* GAC	*V* GUC	*L* CUC	*L* CUU	*H* CAU	*Q* CAA	*Q* CAG	*R* CGG	*W* UGG	*stop* UAG	*stop* UAA	*L* UUA	*L* UUG	*L* CUG	*L* CUA	*P* CCA	*P* CCG	C $\rightarrow m_1$ $\downarrow m_3$
$m_2 \rightarrow$ *M* AUG	*stop* UGA	*R* AGA	*R* AGG	*T* ACG	*S* UCG	*S* UCC	*F* UUC	*I* AUC	*N* AAC	*S* AGC	*G* GGC	*G* GGU	*S* AGU	*N* AAU	*Y* UAU	*C* UGU	*C* UGC	*Y* UAC	*H* CAC	*R* CGC	C $\rightarrow m_2$ $\downarrow m_3$
$m_3 \rightarrow$ *M* AUG	*D* GAU	*E* GAA	*G* GGA	*R* CGA	*R* CGU	*P* CCU	*S* UCU	*S* UCA	*T* ACA	*A* GCA	*A* GCG	*V* GUG	*V* GUA	*I* AUA	*I* AUU	*V* GUU	*A* GCU	*T* ACU	*T* ACC	*A* GCC	C $\rightarrow m_3$ $\downarrow m_1$

b)

$\Leftarrow$ SHIFT 1 $m_1 = m_2$	*stop* UGA	*R** AGG	*R* AGG	T ACG	*S** UCC	*S* UCC	*F* UUC	*I* AUC	*N* AAC	*S* AGC	*G** GGU	*G* GGU	*S* AGU	*N* AAU	*Y* UAU	*C** UGC	*C* UGC	*Y* UAC	*H* CAC	*R* CGC	
m_2 *M* AUG	*stop* UGA	*R* AGA	*R* AGG	*T* ACG	*S* UCG	*S* UCC	*F* UUC	*I* AUC	*N* AAC	*S* AGC	*G* GGC	*G* GGU	*S* AGU	*N* AAU	*Y* UAU	*C* UGU	*C* UGC	*Y* UAC	*H* CAC	*R* CGC	C

c)

⇐ SHIFT 1 $m_2 = m_3$	*E*** GAA	*E* GAA	*G* GGA	*R** CGU	*R* CGU	*P* CCU	*S** UCA	*S* UCA	*T* ACA	*A** GCG	*A* GCG	*V** GUA	*V* GUA	*I** AUU	*I* AUU	*V* GUU	*A* GCU	*T** ACC	*T* ACC	*A* GCC	
m_3 *M* AUG	*D* GAU	*E* GAA	*G* GGA	*R* CGA	*R* CGU	*P* CCU	*S* UCU	*S* UCA	*T* ACA	*A* GCA	*A* GCG	*V* GUG	*V* GUA	*I* AUA	*I* AUU	*V* GUU	*A* GCU	*T* ACU	*T* ACC	*A* GCC	*C*

d)

⇐ SHIFT 1 $m_3 = m_1$	*M* AUG	*K* AAG	*D*** GAC	*D* GAC	*V* GUC	*L** CUU	*L* CUU	*Q*** CAA	*Q** CAG	*Q* CAG	*R* CGG	*W* UGG	*stop** UAA	*stop* UAA	*L** UUG	*L* UUG	*L** CUA	*L* CUA	*P** CCG	*P** CCC	
$m_1 \rightarrow M$ AUG	*M* AUG	*K* AAG	*E* GAG	*D* GAC	*V* GUC	*L* CUC	*L* CUU	*H* CAU	*Q* CAA	*Q* CAG	*R* CGG	*W* UGG	*stop* UAG	*stop* UAA	*L* UUA	*L* UUG	*L* CUG	*L* CUA	*P* CCA	*P* CCG	C

* Coding for the same amino acid.
** Coding for the closest amino acid (by N-end rule and Siemion mutation ring).[3, 31, 32]

e)

$m_1 \rightarrow M$ AUG	*M* AUG	*K* AAG	*E* GAG	*D* GAC	*V* GUC	*L* CUC	*L* CUU	*H* CAU	*Q* CAA	*Q* CAG	*R* CGG	*W* UGG	*stop* UAG	*stop* UAA	*L* UUA	*L* UUG	*L* CUG	*L* CUA	*P* CCA	*P* CCG	$\rightarrow m_1$ C $\downarrow m_3$
$m_3 \rightarrow H$ CAU	*D* GAU	*E* GAA	*G* GGA	*G* GGA	*R* CGU	*P* CCU	*P* CCU	*S* UCA	*S* UCA	*T* ACA	*A* GCG	*V* GUG	*V* GUA	*V* GUA	*I* AUU	*I* AUU	*A* GCU	*A* GCU	*T* ACC	*T* ACC	$\rightarrow m_3$ $\downarrow m_2$
$m_2 \rightarrow A$ GCA	*stop* UGA	*stop* UGA	*R* AGG	*R* AGG	*T* ACG	*S* UCC	*S* UCC	*F* UUC	*I* AUC	*N* AAC	*S* AGC	*G* GGU	*G* GGU	*S* AGU	*N* AAU	*Y* UAU	*C* UGC	*C* UGC	*Y* UAC	*H* CAC	*R* $\rightarrow m_2$ CGC $\downarrow m_1$

REFERENCES

1. D. G. Arques and C. J. Michel, *J. Theor. Biol*. **182** (1996) 45–58.
2. D. G. Arques, J. P. Fallot, and C. J. Michel, *J. Theor. Biol*. **1998** (1997) 241–253.
3. N. Štambuk, *Croat. Chem. Acta* **71** (1998) 573–589.
4. E. Baylis, *Error Correcting Codes*, Chapman & Hall, London, 1998, pp. 38–42.
5. M. Schroeder, *Fractals Chaos, Power Laws*, W. H. Freeman, New York, 1991, pp. 334–340.
6. N. Štambuk, *Mathl. Comput. Modelling* **14** (1990) 565–570.
7. H. Steinhaus, *Wektor* (1917) 1–3. (*A New Property of the Cantor Set*, in: *Hugo Steinhaus, Selected Papers.* Polish Academy of Sciences, Institute of Mathematics, Warszava, 1985, pp. 205–207.
8. N. B. Tufillaro, T. Abbott, and J. Reilly, *An Experimental Approach to Nonlinear Dynamics and Chaos,* Addison-Wesley, Redwood City, 1992, pp. 96–231.
9. N. Štambuk, *Period. Biol.* **101** (1999) 355–361.
10. C. Robinson, *Dynamical Systems*, CRC Press, Boca Raton, 1999, pp. 22–63.
11. J. Banks and V. Dragan, *SIAM Review* **36** (1994) 265–271.
12. J. Milnor, *Commun. Math. Phys*. **99** (1985) 177–195.
13. N. Štambuk, *Croat. Chem. Acta* **72** (1999) 999–1008.
14. C. Calude, *Information and Randomness*, Springer-Verlag, New York, 1994, pp. 1–24.
15. K. T. Alligood, T. D. Sauer, and J. A. Yorke, *CHAOS An Introduction to Dynamical Systems*, Springer-Verlag, New York, 1996, pp. 207–227.
16. O. Pretzel, *Error-Correcting Codes and Finite Fields*, Clarendon Press, Oxford, 1996, pp. 3–62.
17. M. Gardner, *Sci. American* **227** (1972) 106–109.
18. R. Swanson, *Bull. Math. Biol*. **46** (1984) 187–203.
19. D. L. Kreher and D. R. Stinson, *Combinatorial Algorithms*, CRC Press, Boca Raton, 1999, pp. 35–42
20. S. Seiwerth, N. Štambuk, P. Konjevoda, N. Mašić, A. Vasilj, M. Bura, I. Klapan, S. Manojlović, and D. Đanić, *J. Chem Inf. Comput. Sci.* **40** (2000) 545–549.
21. I. H. Witten and E. Frank, *Data Mining*, Morgan Kaufmann Publishers, San Francisco, 2000, pp.169–170.
22. J. A. Cuff and G. J. Barton, *PROTEINS: Structure, Functions and Genetics* **34** (1999) 508–519.
23. G. Haller, *Chaos Near Resonance*, Springer-Verlag, New York, 1999, pp. 28–51.
24. I. Ćosić and D. Nesic, *Eur. J. Biochem*. **170** (1987) 247–252.
25. N. Štambuk and P. Konjevoda, *Period. Biol*. **101** (1999) 363–368.
26. J. A. Lipton and A. Catania, *Immunol. Today* **18** (1997) 140–145.
27. N. Štambuk, N. Kopjar, K. Šentija, V. Garaj-Vrhovac, D. Vikić-Topić, B. Marušić-Della Marina, V. Brinar, M. Trbojević-Čepe, N. Žarković, B. Ćurković, Đ. Babić-Naglić, M. Hadžija, N. Zurak, Z. Brzović, R. Martinić, V. Štambuk, P. Konjevoda, N. Ugrinović, I. Pavlić-Renar, Z. Biđin, and B. Pokrić, *Croat. Chem. Acta* **71** (1998) 591–605.
28. F. H. C. Crick, J. S. Griffith, and L. E. Orgel, *Proc. Natl. Acad. Sci. USA* **43** (1957) 416–421.
29. A. L. Dounce, *Enzymologia* **15** (1952) 251–258.
30. F. L. Bauer, *Decrypted Secrets*, Springer-Verlag, Berlin, 1997, pp. 104–120.
31. I. Z. Siemion, *Amino Acids* **8** (1995) 1–13.
32. A. Varshavsky, *Proc. Natl. Acad. Sci. USA* **93** (1996) 12142–12149.

SAŽETAK

Univerzalna metrička svojstva genetičkog koda

Nikola Štambuk

Istražene su opće metričke osobine genetičkog koda te RNA, DNA i proteinskog kodiranja. Pokazano je da binarna notacija nukleotidnih baza zasnovana na Cantorovu skupu, Gray-ovu kodu, simboličkoj dinamici i Smale-ovoj potkovastoj mapi definira standardnu tablicu genetičkog koda. Definirani su i algoritmi koji opisuju spomenuti odnos kodona i aminokiselina na binarnom drvetu. Pokazano je da ternarni Cantorov algoritam predstavlja najpovoljniji način za kodiranje aminokiselina i kodona na osnovi njihovih purinskih i pirimidinskih prstena te amino- i keto-skupina. Metoda je nazvana Simbolički Cantorov Algoritam (SCA). Istaknuto je da spomenuti način kodiranja gena i proteina odgovara linearnom blok-kodu s trostrukom provjerom, što upućuje na to da se ne radi o »degeneriranom kodu« već o kodu koji popravlja pogreške. Dana je i tablica koda. Spomenuti tip koda definira riječi (kodone, aminokiseline) i njihovu transkripciju i translaciju, dok duže leksičke strukture kodira cirkularni kod na principu ogrlice s tri niza. Određeni su nizovi cirkularnog koda koji definiraju mRNA i komplementarne DNA. Strojnim klasifikatorom C5.0 te algoritmom SCA definirana je, iz nukleotidne sekvencije, sekundarna struktura 50 proteina sa 94% točnosti. Spektralna (Fourier-ova) analiza hormona i citokina metodom SCA odredila je bioaktivne dijelove molekula, te ukazala na moguću primjenu metode u praksi.