8. Філинюк, М. А. Методи та засоби вимірювання параметрів потенційно-нестійких чотириполюсників / М. А. Філинюк, К. В. Огороднік, Л. Б. Ліщинська. Вінниця : ВНТУ, 2010. – 176 с. – Бібліогр. : с. 173–175. – ISBN 978-966-641-342-3.
9. Богачев, В. М. Транзисторные усилители мощности / В. М. Богачев, В. В. Никифоров – М. : Энергия, 1978. – 344 с.
10. Брежнева, К. М. Транзисторы для аппаратуры широкого применения : справочник / К. М. Брежнева, Е. И. Гантман, Т. И. Давыдова и др. ; под ред. Б. Л. Перельмана. – М. : Радио и связь, 1981. – 656 с.

УДК 612.17

HEART RATE COMPLEXITY DEFINITION USING KOLMOGOROV ALGORITHMIC COMPLEXITY METHOD

O. Kyselova
Senior Lecturer in Computer Science*
Tel.: +38-050-443-02-82
E-mail: olga.mmif@gmail.com

Ie. Nastenko
Dr.Sci.Biol., Head of Informatics and Computational Physiology Department
Amosov National Institute of Cardiovascular Surgery of National Academy of Medical Science of Ukraine
6, Amosov street, 252601 GSP, Kyiv, Ukraine
E -mail: nastenko@inbox.ru

M. Gerasymchuk*
E-mail: maxgerasimchuk@gmail.com
*Department of Therapeutical Diagnostic Complexes
Intercollegiate College of Medical Engineering
National Technical University of Ukraine “Kyiv Polytechnic Institute”
prospect Peremogy, 37, Kyiv, Ukraine, 03056

Стаття присвячена застосуванню метода алгоритмічного стиснення за Колмогоровим для визначення складності поведінки добового серцевого ритму

Ключові слова: алгоритмічна складність за Колмогоровим, динаміка серцевого ритму, нелінійна динаміка, алгоритм стиснення Lempel-Ziv-Welch (LZW)

Статья посвящена применению метода алгоритмического сжатия по Колмогорову для определения сложности поведения суточного сердечного ритма

Ключевые слова: алгоритмическая сложность по Колмогорову, динамика сердечного ритма, нелинейная динамика, алгоритм сжатия Lempel-Ziv-Welch (LZW)

This paper describes the possibility of Kolmogorov algorithmic complexity method using for definition of heart rate complexity

Keywords: Kolmogorov complexity, the heart rate dynamics, nonlinear dynamics, Lempel-Ziv-Welch compression algorithm (LZW)

Introduction

The Kolmogorov algorithmic complexity of an object, such as a piece of text, is a measure of the computational resources needed to specify the object. Kolmogorov complexity of a bit string x is the length of the shortest program that computes x and halts.

One can draw an analogy with the archiving. Thus we understand the complexity of the file as the size of the smallest archive that after unpacking is this file. In other words Kolmogorov complexity is the minimum number of bits into which a string can be compressed without losing information.

The Kolmogorov complexity decline especially when combined with a variability of a numerical sequence reduction is rather serious indication of the body regulatory redundancy reduce. And it can be regarded as indication of increasing probability of sudden cardiac death occurrence. Such situations are often observed in coronary care units in patients with high severity of conditions (e.g. acute myocardial infarction, several old myocardial infarction, etc).

Description

Determining the complexity of the behavior of biological signals by the Kolmogorov complexity allows to estimate the level of algorithmic complexity of the variation of heart rate.

The method is to use linear archivator and to compare the original and the compressed cardio sequences. This method, along with a value of approximate entropy [2], can be used to assess the frequency of cardiointervalogram fragments. From the original cardiointervalogram and/or difference sequence dictionary of regularity is constructed. It is estimated the content, frequency of relapses and the length of regularities, including extrasystoles, rhythm jogging, etc and their temporal location (time of day, when they occur).

Extraction of regularities in temporal sequence allows archiving this sequence. In this case, the compression ratio (the ratio of the archived sequences length to the original sequence length) is a quantitative characteristic of the complexity of HRV (Heart Rate Variability).

In order to identify recurring areas of cardiac rhythm a compression algorithm Lempel-Ziv-Welch (LZW) [3, 4] is used.

To register sequences of RR-intervals that occur in the HR special dictionary is used. Dictionary is a table that contains the sequence code, the sequence itself and its number of occurrences in the input stream. This table has a so-called precedence property. For each sequence wK in the dictionary, which consists of some sequence of time intervals w and interval K, the dictionary contains sequence w also.

We propose some terms used in this document:

1) «String» - a number of continuous RR-intervals, anywhere from one to very many intervals in length. An arbitrary string is specified as «[...]K»;

2) «Prefix» - almost the same as a string, but with the implication that a prefix immediately precedes an interval, and a prefix can have a length of zero. So, a prefix and an interval make up a string. An arbitrary prefix is specified as «[...]»;

3) «Current prefix» is specified as «[.C.]». This prefix is to be compared with. Initially, the current prefix is empty;

4) «Current string» is specified as «[.C.]K», where K is some interval. This string is the current prefix plus the next RR-interval in the input stream.

Dictionary construction algorithm (by LZW compression algorithm):

1) Initialize the dictionary with the values of RR-intervals (single element strings) in the range from minimum value of the input data to a maximum (step is 1ms);

2) Read the first value (RR-interval) from the input file (specified as P), [. C.] P becomes the current string;

3) Search through the dictionary string to see if [.c.]P appears in it. Of course, it does on the first iteration, because our dictionary is initialized to have all values of RR-intervals that appear in input data. Now the current prefix is [.C.] P;

4) Go to the next RR-interval in the input stream. Call it Q. Add it to the current prefix to form [.c.]Q, the current string.

5) Search through the dictionary to see if [.c.]Q appears in it. During the first iteration, of course, it does not appear. Add [.C.]Q to the dictionary. The current prefix is Q;

6) Keep adding RR-intervals to [.c.] to form [.c.]K, until [.c.]K is couldn't be found in the string table. Add [.C.]K to the string dictionary.

In pseudo-code, the algorithm looks like this [4]:

```
[1] Initialize dictionary;
[2] [.c.] <- empty;
[3] K <- next interval in input stream;
[4] Is [.c.]K in dictionary?
(yes: [.c.] <- [.c.]K;
go to [3];
)
(no: add [.c.]K to the dictionary;
[.c.] <- K;
go to [3];
)
```

When you get to step [3] and there are now more RR-intervals left, the formation of the dictionary is over.

Table 1

Step by step the dictionary is constructed (by LZW compression algorithm) for input sequence of numbers "700 800 900"

Element (K)	Current string [.C.]K	Elements, added to the dictionary (string - position)
700	700	
800	800	700 800 (4)
700	800 700	800 700 (5)
800	700 800	
900	700 800 900	700 800 900 (6)
800	900 800	900 800 (7)
700	800 700	
800	800 700 800	800 700 800 (8)
700	800 700	
800	800 700 800	
700	800 700 800 700	800 700 800 700 (9)
700	700 700	700 700 (10)
700	700 700	
700	700 700 700	700 700 700 (11)
700	700 700	
700	700 700 700	
700	700 700 700 700	700 700 700 700 (12)

Table 2

Output dictionary

Strings, added to the dictionary during initialization	Code of the string in the dictionary
700	1
800	2
900	3
Strings, added to the dictionary during compress	Code of the string in the dictionary
700 800	4
800 700	5
700 800 900	6
900 800	7
800 700 800	8
800 700 800 700	9
700 700	10
700 700 700	11
700 700 700 700	12

The procedure of filling the dictionary is performed several times for a more complete dictionary and to identify new chains. At the same time the dictionary that was generated in the previous step is used, instead of initialization (which is performed only at the first step).

Then for each chain of the dictionary search is being performed in the input data. In the process of searching the dictionary is updated and data about the number of occurrences of each chain in the initial cardiointervalogram is recorded.

The dictionary, that was obtained in the n-th iteration of the compression algorithm, contains all received sequences of time intervals.

Due to the high degree of variation of input values, dynamic series require ”averaging”. This occurs as follows:

a) Percentage of ”averaging” is defined (option ”Desensitize”);

b) The minimum (min_value) and maximum (max_value) values of the time interval in the original data is searched;

c) A series of values, starting with min_value and ending max_value with step “step” is formed. The value of step is:

step = min_value * percentage of ”averaging”; (1)

d) The output series of time intervals is reduced to the resulting series of values. Thus, each element of the series differs from the corresponding element of the original series in not more than a specified percentage.

Conclusions about the variability of patient’s heart rate can draw based on the analysis of the number and length of regularities, contained in the dictionary, as well as on the evaluating compression ratio (the ratio of the length of compressed series to the length of the original series).

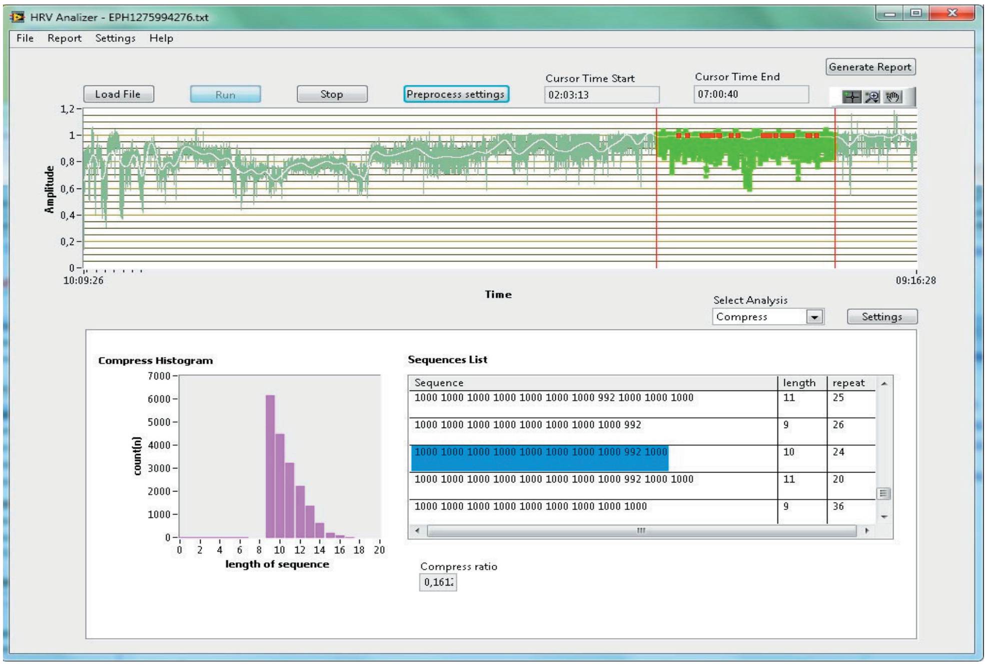
Implementation

The program for cardiac rhythm analysis (Fig. 1) was developed in NI LabVIEW 9.0. The results of heart rate complexity estimating is described in Fig. 1, where (a) is the main graphic user interface, (b, c, d) is heart rate with repeatable lines in different zoom which is shown in (e).

As a result of the analysis we obtain a quantitative measure of compression degree of selected area - compress ratio.

“Compress Histogram” (Fig. 1 (e)) shows the dependence of the number of regularities on their length.

“Sequences List” contains the chain of intervals, matching the search criteria. Their length and number of repetitions in the original cardiointervalogram are also indicated. Double-click on any sequence in the list leads to display all its repetitions in the original data in the chart as red spots (Fig. 1 (a, b)). Thus we can estimate the time of appearance of specific regularity in heart rhythm.



Cursor Time Start: 02:03:13

Cursor Time End: 07:00:40

Select Analysis: Compress

Sequences List

Sequence	length	repeat
1000 1000 1000 1000 1000 1000 1000 992 1000 1000 1000	11	25
1000 1000 1000 1000 1000 1000 1000 1000 992	9	26
1000 1000 1000 1000 1000 1000 1000 1000 992 1000	10	24
1000 1000 1000 1000 1000 1000 1000 1000 992 1000 1000	11	20
1000 1000 1000 1000 1000 1000 1000 1000 1000	9	36

Compress ratio: 0,161:

Fig.1 (a). The main Interface of the software (results of the Kolmogorov algorithmic complexity method)

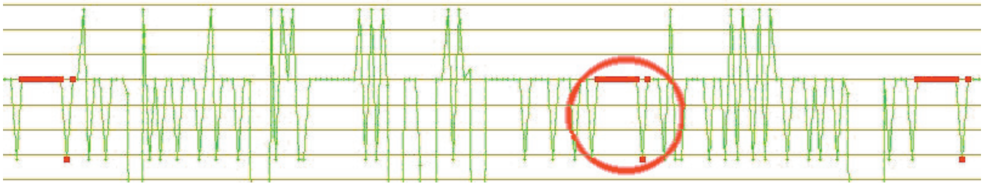
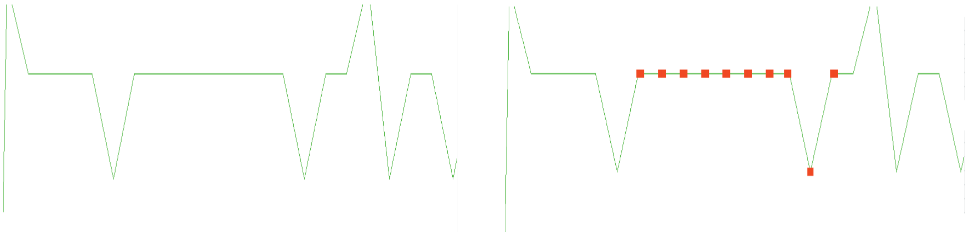
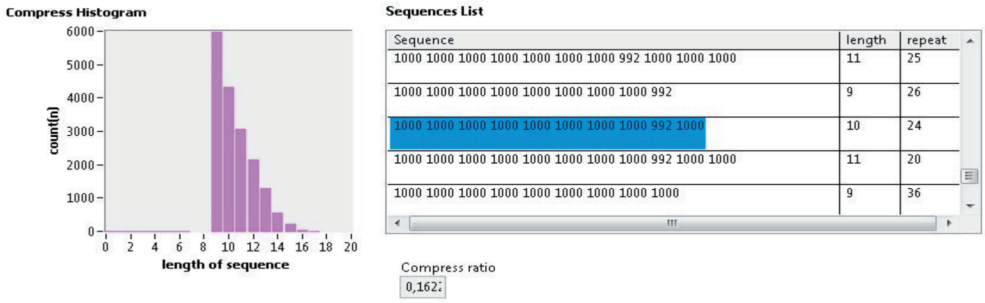


Fig.1 (b). Chart of repetitions in the original data



c) d)

Fig.1 (c, d). Heart rate with repeatable lines in different zoom

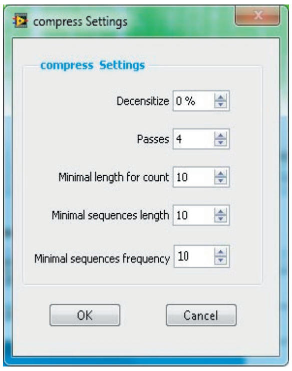


Sequence	length	repeat
1000 1000 1000 1000 1000 1000 1000 992 1000 1000 1000	11	25
1000 1000 1000 1000 1000 1000 1000 1000 992	9	26
1000 1000 1000 1000 1000 1000 1000 1000 992 1000	10	24
1000 1000 1000 1000 1000 1000 1000 1000 992 1000 1000	11	20
1000 1000 1000 1000 1000 1000 1000 1000 1000	9	36

Compress ratio 0,162

Fig. 1 (e). The Histogram of compression heart rate with the repeatable lines with their numbers and lengths

The window with settings for Kolmogorov algorithmic complexity method of analysis is shown in Fig. 2.



Decensitize 0 %

Passes 4

Minimal length for count 10

Minimal sequences length 10

Minimal sequences frequency 10

Fig. 2. Window with compress settings

Where value ”Passes” assigns the number of steps to construct the dictionary. ”Minimal length for count” specifies the minimal length of the sequences, that will be counted. ”Minimal sequences length” assigns the minimal length of cardio sequences that will be displayed in the ”Sequences List”. Velue ”Minimal sequences frequency” determines the minimal number of repetitions of cardio sequences, which will be displayed in the ”Sequences List”.

Conclusions

The use of methods of nonlinear dynamics in the cardial rhythm analysis is becoming more and more popular in recent years [5, 6]. It is due to the fact that normal CR does not demonstrate periodical dynamics but a chaotic one. Besides, normal physiological flexibility of such nonlinear system as heart is its chaotic and complex behavior [7, 8, 9, 10]. One of the assessment methods of the complex behavior of heart rate is the method of algorithmic complexity by Kolmogorov. The principle of its work is in the use of archivator and dictionary to it, which contains repeated strains of signs.

References

1. Nicolis G., Prigogine I. Exploring Complexity. N.Y.: W.H. Freeman, 1989.
2. Pincus SM. Approximate entropy as a measure of system complexity. Proc Natl Acad Sci USA 1991; 88:2297-2301.
3. J. Ziv and A. Lempel. A Universal Algorithm for Sequentia Data Compression, “IEEE Transactions on Information Theory, May 1977.
4. http://www.cis.udel.edu/~amer/CISC651/lzw.and.gif.explained.html.
5. Goldberger A.L. Nonlinear dynamics, fractals, cardiac physiology and sudden death. Temporal disorder in human oscillatory systems. Ed. L. Rensing. N.Y., 1986. P. 118.
6. Pool R. Is it healthy to be chaotic? Science. 1989. V.243. P.604.
7. Babloyantz A., Desstexhe A. Is the normal heart a periodic oscillatory. Biol. Cybern. 1988. V. 58. P. 203.
8. O. Kyselova, Ie. Nastenko, A. Stankus, V. Yatsenko. The use of characteristics of HR behavior complexity for human organism adaptive properties estimation. Proceedings of the International Conference “Biomedical Engineering”, October 28-29, 2010, P. 53 - 57.
9. Kyselova O., Nastenko Ie. Estimation of Heart rate Complexity of Behavior using Different Methods of Nonlinear Dynamics. Proceedings of the Third International Conference “Nonlinear Dynamics - 2010”, September, 21-24, 2010, Kharkov, Ukraine. P. 125 – 128.
10. Albinas Stankus, Olga Kyselova, Gintare Jatautaite, Igor Basov. Paros sirdies ritmo simpatines ir parasimpatines saveikos dinamikos indetifikavimo metodas. Proceeding of International Conference “Virtual Instruments in Biomedicine”, (2009), P. 116-122.