GENOMIN: A SOFTWARE FRAMEWORK FOR READING GENOMIC SIGNALS

PAUL GAGNIUC[1], DĂNUȚ CIMPONERIU[1], CONSTANTIN IONESCU-TÎRGOVIȘTE, CRISTIAN GUJA[2], POMPILIA APOSTOL[1], MONICA STAVARACHI[1] and LUCIAN GAVRILĂ[1]

[1] Human Genome and Molecular Diagnosis Laboratory, Institute of Genetics, University of Bucharest, Romania
[2] "N. C. Paulescu" National Institute for Diabetes Nutrition and Metabolic Diseases, Bucharest, Romania
Corresponding author: Paul Gagniuc, Address: No.1-3, Portocalelor Street, Bucharest, Romania, zip code: 060101, Phone: 004-0727-985-684, E-mail addresses: paulgagniuc@yahoo.com

Received June 17, 2011

Data mining produces models that capture and represent hidden patterns in the DNA structure. Any attempt to develop and test new algorithms for data mining in the field of bioinformatics, must begin with an optimal method by which even the huge FASTA files can be read step by step. The aim of the GENOMIN software is to provide an open source software platform which can work with large files like a whole chromosome or genome sequence. We have created an open source template software, named GENOMIN, for analyzing genetic data of sequences of different sizes downloaded from NCBI servers. Large NCBI FASTA files which store sequences of individual chromosomes come from other processing systems like UNIX. Processing these files on other operating systems is difficult due to different markers which indicate the end of each line. The GENOMIN software, reads the FASTA files by continuous buffer reading, without taking into account the end of line markers. The result of this type of reading is a brute, noisy free DNA sequence of the entire file regardless of its size. We presented three examples to demonstrate how the program can be used in biology: the estimation of GC content, identification of repetitive elements and search for sequences with different biological functions (e.g. duplicated regions or potential binding sites for transcription factors). Development of this open source software is limited only by the researcher programming skills. The results of our tests have been shown that GENOMIN can perform various tests on large sequences files and can work with different algorithms used in biology.

Key words: Genomin, open source, data mining, nucleotide sequence, CpG.

INTRODUCTION

The resulting patrimony of genomic sequence information stepped into a decade of increasingly rich sequence databases. The field of bioinformatics has rapidly developed into an essential asset for modern biology and powerful bioinformatics tools have been developed. We present a new and efficient computational method to extract, analyze and interpret biological data. Genomin is an open source platform publicly available through the World Wide Web. Some software like *TESS, GeneSolve, GENLANG, Sdiscover, Splign* and a variety of online applications from Pasteur Institute, called "*Logiciels pour la biologie*", can be used for predicting transcription factor binding sites in DNA sequences, for analyzing nucleic acid sequence data and to locate genes. *Sdiscover* is a tool for finding motifs in sequences. *Splign* is a utility for computing cDNA-to-Genomic, or spliced sequence alignments.

Public databases provide DNA, RNA and protein sequences in several file formats. FASTA is one of these formats which contains a series of text lines[1]. The first line of a DNA file (*i.e.* sequence header) starts with a ">" symbol. The following lines have a constant length (usually less than 80 characters) and represent the DNA sequence. The end of each line is represented by line feed (LF) or carriage return (CR) characters in different operating systems. A CR is the number 13 whereas a LF is the number 10 in the ASCII table of characters. The end of lines in Microsoft Windows is represented as CRLF or carriage return and line feed, which is a CR followed by a LF.

Researchers prefer FASTA format for the painless effort in representing, handling and manipulating the nucleotide or peptide sequences[2,3] and parse them in different scripting languages like PHP, Perl, Python, VBS or JS.

A frequent aim in bioinformatics is to find certain patterns (*e.g.* in a chromosome or a whole genome sequence). Different methods can be used when short DNA sequences (*e.g.* a gene or a cluster of genes) are analyzed. Processing large sequence files requires many hardware resources and it may be a real problem for some operating systems[4,5].

The aim of the GENOMIN project is to provide an open source software platform which can work with large files like a whole chromosome or genome sequence.

MATERIALS AND METHODS

We begin by writing the "CD" (Character Detection) function used for detecting the operating system on which the FASTA file was generated.

```
Function CD(ByVal s As String) As String
Dim LF() As String
Dim CR() As String

LF() = Split(s, Chr(10))
CR() = Split(s, Chr(13))

If UBound(LF) > UBound(CR) Then
    CD = Chr(10)
Else
    CD = Chr(13)
End If

End Function
```

Then we define and establish the memory allocation for the global "buff" variable which temporarily stores segments of data. This variable length is chosen according to the maximum length of the first line of any sequence from the file.

Next a FASTA file is opened through sequential data reading with "seek" function. It returns a value specifying the current read/write position within a file, or sets the position for the next read/write operation in the same file.

Every time a header is found in this process of reading the DNA sequence, the amount of information increases twice in "tmp_dat" variable. This is achieved by joining two buffers within the same reading cycle. If a new header line is found within a FASTA file, it is removed from the two joined buffers. Figure 1 tries to show the reason for which the minimum length of the "tmp_dat" variable is twice the length of a buffer, when new contig headers are found inside a FASTA file.

Through "process_DNA" function, raw data from the file will be filtered, FASTA header line (if any), line feed and carriage return characters will be removed.

The final step is the filtration of the sequence which is implemented through "Replace" function. This function replaces all or just a specified part of a string with another string, which in our case is returned by "CD" function.

To avoid disruptions in the DNA sequence, we introduce the *Buffer_Stream* variable, which allows continuity in the buffer flow, by appending the last sliding window with the new data coming from the file. GENOMIN software was developed in Visual Basic 6 and the shorten implementation of the source code is presented below.

```
Dim buff As Variant
Dim Window As Variant

Private Sub OpenFASTA_Click()
    Dim FileNum As Integer
    Dim sFile As String
    Dim alta_secventa As Boolean
    Dim dat As String
    Dim i As Long

    sFile = "path_to_fasta_file"
    buff = 132

    FileNum = FreeFile
    Open sFile For Binary As #FileNum
```

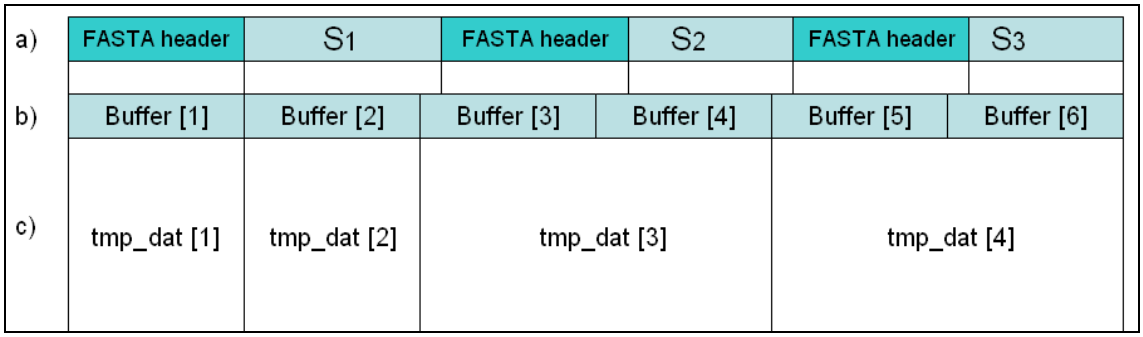

Fig. 1. Representation of correlation between a DNA sequence and "tmp_dat" variable. Section a) represents a FASTA file which contains three contig sequences, section b) represents the distribution of the buffers in comparison with the FASTA sequence headers, and section c) represents "tmp_dat" length.

```
    lungime = LOF(FileNum)

    dat = String$(buff, vbNullChar)

    For i = buff To lungime Step buff
1:
        Get #FileNum, , dat
        Seek #FileNum, i + 1

        tmp_dat = tmp_dat & dat

        If InStr(dat, ">") Then
            i = i + buff
            alta_secventa = True
            GoTo 1
        End If

        Call process_DNA(tmp_dat,
alta_secventa)
        tmp_dat = Empty
        alta_secventa = False

    Next i

    Close #FileNum

End Sub

Function process_DNA(ByVal x As
String, _
ByVal alta_seq As Boolean)
x = LCase(x)

If alta_seq = True Then
    tmp_1 = Split(x, ">")(0)
    tmp = Split(x, ">")(1)

        If InStr(tmp, Chr(10)) Then
            tmp_2 = Split(tmp,
Chr(10))(1)
        Else
            tmp_2 = ""
        End If

    x = tmp_1 & tmp_2

End If

x = Replace(x, Chr(13), "")
x = Replace(x, Chr(10), "")
Buffer_Stream = Window & x

Call Slide_Window(Buffer_Stream)

End Function


Function Slide_Window(ByVal x As
String)

For i = 1 To Len(x) - Window_Length
  Window = Mid(x, i, Window_Length)
  Call Process_Window(Window)
Next i

End Function


Function Process_Window(ByVal Window
As String)

Call EXEMPLE_1(Window)
Call EXEMPLE_2(Window)
Call EXEMPLE_3(Window)

DoEvents
End Function
```

Nucleotide Files

Our tests were conducted on FASTA files downloaded from NCBI servers. We tested the sequences of all human chromosomes, human mitochondrial genome, genes associated with obesity and NAIP gene (NG_008724.1, *Homo sapiens* NLR family, apoptosis inhibitory protein), NAIP pseudogene (NG_006114, Homo sapiens NLR family, apoptosis inhibitory protein pseudogene) and the sequences corresponding to TA278 genome (the prototype of group 1 of Torque teno virus, 3.8 kb)[3,6].

We created three examples to demonstrate how the program can be used for different applications in biology: the estimation of GC content, identification of repetitive elements, for example dinucleotides, $TACA_6$ (Thymine Adenine Cytosine Adenine) or duplicated regions and search for sequences with different biological functions (*e.g.* potential binding sites for transcription factors). There were a total of 187 tests. Each file has undergone at least three GENOMIN trials. The GENOMIN software was tested on a computer equipped with a 2.8GHz CPU, 500MB RAM, 80GB HDD.

Cytosine and Guanine content of a Clean Sequence (CGCS)

The “clean sequence” term refers to the elimination of IUPAC characters for nucleic acids (M, S, W, B, U, D, R, H, Y, V, K, N, – characters). The numerical difference between the file sizes and the clear sequence produced by GENOMIN, is due to CR, LF, IUPAC characters and sequence headers elimination. CGCS correlation is carried out depending on the length of each chromosome (clean sequence). CGCS is calculated as follows CGCS = (100/L)×CG, where L is the length of the clean sequence.

Table 1

The size of chromosomes from Homo sapiens genome and CGCS (CG content of clean sequence lenght)

Homo sapiens Chr.	GENOMIN GRCh37 file size	GENOMIN GRCh37 clean sequence	CGCS	Base composition estimation (%GC)	
				Previously published data*	GENOMIN estimation
Chr 1	230178751	225588139	1.848	42	41.745
Chr 2	241763884	238016407	1.690	40	40.243
Chr 3	197580391	194737999	2.038	40	39.695
Chr 4	191380292	188440241	2.029	38	38.25
Chr 5	180285019	177610671	2.224	40	39.516
Chr 6	203700622	194915031	2.070	40	40.359
Chr 7	158266526	155408472	2.622	40	40.754
Chr 8	145059076	142811900	2.813	40	40.176
Chr 9	123265711	120200482	3.458	41	41.31
Chr 10	133597254	131186080	3.170	42	41.587
Chr 11	133246882	131095534	3.171	42	41.572
Chr 12	132397294	130377498	3.130	41	40.81
Chr 13	96955989	95503787	4.033	38	38.524
Chr 14	89550929	88269084	4.632	41	40.888
Chr 15	83396744	81584796	5.172	42	42.198
Chr 16	80062828	78819961	5.682	44	44.789
Chr 17	81053284	79545597	5.724	45	45.537
Chr 18	75829978	74602674	5.333	40	39.786
Chr 19	56913098	55968375	8.637	49	48.343
Chr 20	60457460	59428183	7.426	44	44.132
Chr 21	35690033	34994801	1.167	41	40.851
Chr 22	35393494	34822954	1.378	48	47.991
Chr X	153514546	150844219	2.617	39	39.491
Chr Y	26123297	25391744	1.572	39	39.934

* Acording to J. Craig Venter 2001

RESULTS

GENOMIN source code and binaries can be downloaded from: http://genomin.novusordo.ro. It runs on all Windows operating systems, no installation required and the complete package has 4.28Mb. GENOMIN memory requirements are between 6.9Mb and 8Mb, depending on Windows OS version.

On average, GENOMIN scanned the human genome files in about two hours whereas genes, viral or mitochondrial genomes were analyzed in several seconds (Table 2).

EXAMPLE 1 – detection of C and G percentage. The C and G content is plotted for each sliding window or buffer on the y-axis, as the maximum and minimum percentage.

```
Function EXEMPLE_1(ByRef Window As String)

For i = 1 To Len(Window)
   nucleotide = Mid(Window, i, 1)
   If nucleotide = "a" Then a = a + 1
   If nucleotide = "t" Then t = t + 1
   If nucleotide = "g" Then g = g + 1
   If nucleotide = "c" Then c = c + 1
Next i

Total_CG = (100 / (c + g + t + a)) * (c + g)

par = Picture1.ScaleWidth / total_sequence
y = Picture1.ScaleHeight / 100
x = par * position_in_sequence

Picture1.Line (x, Total_CG - 1 * y)-(x, _
Total_CG * y), vbRed

Line1.X1 = (par * position_in_sequence) + 1
```

```
 21 Line1.X2 = (par *
position_in_sequence) + 1
 22
 23 DoEvents
 24 End Function
```

EXAMPLE 2 – detection of dinucleotide repeats in a DNA sequence. The number of dinucleotide repeats are plotted on y-axis, as the maximum percentage for each buffer (or sliding window) separately. As can be seen below, we consider the order and position of each nucleotide from the window content.

```
  1 Function EXEMPLE_2(ByRef Window
As String)
  2 Dim CG nr() As String
  3
  4 Rep = 1
  5 DubleN = "CG"
  6
  7 nucleo_test = LCase(DubleN)
  8 For ye = 1 To Val(Rep)
  9 rep_CG = rep_CG & nucleo_test
 10 Next ye
 11
 12 CG_nr = Split(Window, rep_CG)
 13 CG_nr_buff = UBound(CG_nr)
 14
 15 op = CG_nr_buff * (2 * Val(Rep))
 16 Total_CG = (100 / Len(Window)) *
op
 17
 18 par = Picture1.ScaleWidth /
total_sequence
 19 y = Picture1.ScaleHeight / 100
 20 x = par * position_in_sequence
 21
 22 If CG_nr_buff > 0 Then
 23  Global_CG_nr_buff =
Global_CG_nr_buff + CG_nr_buff
 24  GCNT.Caption = "Total (CG)" &
Rep.Text & " = "
 25   & Global_CG_nr_buff
 26 End If
 27
 28 DoEvents
 29
 30 If Rec_buff.Value = 0 Then
 31  Call add_tmp_result("EXAMPLE 2 -
No. buffer: ["
 32   & Int(x) & "] - > (GC)<font
size=2>n</font>, n="
 33   & Rep.Text & " -> percentage:"
& Int(Total_CG) & "%" & vbCrLf)
 34 End If
 35
 36 Picture2.Line (x, 100)-(x, 100 -
Total_CG), vbBlue
 37
 38 Line2.X1 = (par *
position_in_sequence) + 1
 39 Line2.X2 = (par *
position_in_sequence) + 1
 40
 41 DoEvents
 42 End Function
```

Table 2

Results obtained by GENOMIN program after scanning the sequence of all human chromosomes, human mitochondrial genome and the TTV virus genome

Chromosome *Homo sapiens*	GENOMIN processing time	Base composition estimation (%GC)		(GC)n					Motif (TACA) 6
		Previously published data*	GENOMIN estimation	n=3	n=5	n=7	n=9	n=12	
Chromosome 1	9 min	42	41.745	3474	167	38	14	3	43
Chromosome 2	8 min	40	40.243	2753	111	29	6	1	38
Chromosome 3	7 min	40	39.695	1868	99	24	3	1	20
Chromosome 4	7 min	38	38.25	1708	78	25	7	1	34
Chromosome 5	6 min	40	39.517	1793	96	28	6	1	32
Chromosome 6	6 min	40	40.359	2719	118	27	5	0	28
Chromosome 7	5 min	40	40.754	2216	83	19	4	0	34
Chromosome 8	5 min	40	40.176	1727	63	18	5	0	26
Chromosome 9	4 min	41	41.31	1766	84	19	2	0	27
Chromosome 10	4 min	42	41.587	1884	65	12	3	0	28
Chromosome 11	4 min	42	41.572	1841	92	18	1	0	24
Chromosome 12	4 min	41	40.81	1706	82	17	5	1	25
Chromosome 13	3 min	38	38.524	886	39	7	1	0	18
Chromosome 14	3 min	41	40.888	1212	51	11	2	1	21
Chromosome 15	3 min	42	42.198	1269	51	18	2	0	14
Chromosome 16	3 min	44	44.789	1693	56	16	6	0	9
Chromosome 17	2 min	45	45.537	2113	85	17	5	0	13

Table 2 (continued)

Chromosome 18	2 min	40	39.786	904	48	10	1	0	7
Chromosome 19	1 min	49	48.343	2411	78	11	3	1	15
Chromosome 20	2 min	44	44.132	1146	41	11	2	0	11
Chromosome 21	1 min	41	40.851	537	20	4	0	0	8
Chromosome 22	1 min	48	47.991	1029	29	5	2	0	6
Chromosome X	9 min	39	39.491	1220	72	16	4	0	23
Chromosome Y	1 min	39	39.934	151	8	3	1	0	10
Chromosome MT	2s		44.421	0	0	0	0	0	0
TTV Chromosome	2ms		48.301	5	0	0	0	0	0

* Acording to J. Craig Venter 2001

Dinucleotide ((*XX*)*n* ,where *X* can be any type of nucleotide) searching can be done either directly on the buffer variable or on each sliding window separately. If the searching of dinucleotide repeats is performed separately for each sliding window, then the sliding window size should be larger than (*XX*)*n* (the value of *n* can be modified by the user) but can not exceed the maximum length of a buffer (Annexe 4).

EXEMPLE 3 – Searching of "motif" sequences in a sequence file. The function begins by declaring the *motif* sequence that will be searched in the current *Window* variable, provided by the "*Process_Window*" function. The sequence of interest can be the recognition site for endonucleases, repetitive sequences (like tandem repetitions), a duplicated exon or a cis-regulatory element (a binding site for transcription factors). If a sequence of interest is found, then the *flag* variable will take the value 100, otherwise will be zero. The result is represented on a graph with one or more vertical lines at the position at which the motif was found within the chromosome file.

```
  1 Function EXEMPLE_3(ByRef Window
As String)
  2 Dim n_motifs() As String
  3
  4 motif_sequence = "aagctt"
  5
  6 n_motifs = Split(LCase(Window),
LCase(motif_sequence))
  7
  8 tmp_motif = UBound(n_motifs)
  9 If tmp_motif > 0 Then flag = 100
Else flag = 0
 10
 11 motif_count = motif_count +
tmp_motif
 12 Motif_F.Caption = "Total motifs
found: " & motif_count
 13
 14 par = Picture1.ScaleWidth /
total_sequence
 15
 16 Line3.X1 = (par *
position_in_sequence) + 1
 17 Line3.X2 = (par *
position_in_sequence) + 1
 18
 19 If flag = 100 Then
 20
 21 Total_CG = flag
 22 y = Picture1.ScaleHeight / 100
 23 x = par * position_in_sequence
 24
 25 Picture3.Line (x, 0)-(x, Total_CG
* y), &H8000&
 26
 27 If UTN.Value = 1 Then
 28 Picture3.CurrentX = x + 1
 29 Picture3.CurrentY = 20
 30 Picture3.Font.Size = 8
 31 Picture3.Print "M=" & tmp_motif
 32 End If
 33
 34 Call add_tmp_result("EXAMPLE 3 -
No. motifs found: [" _
 35  & tmp_motif & "] - > Relative
chromosome position:" _
 36  & position_in_sequence & "b" &
vbCrLf)
 37
 38 End If
 39
 40 DoEvents
 41 End Function
```

Large DNA sequence files are difficult to manipulate through simple programming techniques. For example memory necessary for loading the "hs_ref_GRCh37_chr1.fa" file (size: 224Mb) is obviously very high. Instead, a sequential data reading needs less memory and preserves the computer resources.

DISCUSSION

Analysis of genomic raw data[9], may bring new features in data handling and better visualization systems[10–12], which would be difficult to grasp by using standard software. The main advantage of GENOMIN program is the power of visualization and the ability to handle large-scale DNA sequences. Through GENOMIN, one can develop user-friendly applications for Hidden Markov Models[7] or for converting DNA sequences into other types of digital signals[8].

In order to use visualization and data mining techniques for distinguishing relevant DNA sequences, it is necessary to represent symbolic sequences by vectors. Thus improves the ability of other software to identify genes, pseudogenes, segmental duplication or low complexity sequences[13, 14].

Analysis on a DNA sequence in GENOMIN can be made on buffers or sliding windows with lengths established by the user. The "automatic setting optimization" option adjusts buffer size based on the analyzed sequence length in an attempt to optimize the scanning time. Scan results can be saved in a long (analysis of chromosome 1 sequence can produce a file up to 4.5MB) or short HTML format. This format helps the user to publish the results online.

GENOMIN was developed in Visual Basic programming language. Although Visual Basic syntax is less common in bioinformatics open-source applications[15], ensures a greater portability to all BASIC like programming languages or other scripting languages (*eg.* VBS or ASP.NET). Starting with Visual Basic 6 programming language[16], the powerful "*Split*" function was introduced, which can operate very easy on string arrays. This function used in GENOMIN saves the coding effort of having to set up loops and using combinations of other basic time-consuming string functions to perform the equivalent tasks. "Split" function based on delimiter criterion, creates one-dimensional array containing a specified number of substrings.

The "*CD*" function from our software is run only once before the process of reading the entire FASTA file, to determine what data should be replaced in the buffer (respectively CR or LF characters).

The limit of 80 characters for each line is considered for compatibility issues with reference to other older software. In some old software the memory preallocation was made for fixed line sizes which are prone to buffer overflow dangers. Programming languages commonly associated with buffer overflows include C++ and C, which provide no protection for overwriting data in any part of memory.

The content and distribution of GC into the genome could have some functional relevance (*i.e.* to detect gene promoters). We estimated that the average G+C content of human chromosomes sequence ranged between 38,2% (the chromosome 4) and 48.3% (the chromosome 19). The average G+C content of human genome estimated by us (41,6% (SD±2,68) is similar with previously published data (*i.e.* 40.9–41.5%)[17,18].

Prediction algorithms based on nucleotide sequences increase the number of annotations[19] regarding genes and pseudogenes structure, alternative splicing sites[20], transcription factor binding sites[21], CpG islands[22] or physical characteristics involve in particular DNA-dependent processes[23]. The "sliding-window" method improves the efficiency of these algoritms to detect these sites in genomic studies. In addition, the signal processing methods used by GENOMIN can improve the identification of isochores and of GC-poor regions (which can represent deserts of genes).

The chromosome 5 presents multiple segmental duplications located 5p14, 5p13, 5q13, 5q15–5q21[24]. The duplicated segments mapped 5q13 has 500 kb and contains several duplicated genes included the NAIP gene and the NAIP pseudogene. The NAIP pseudogene lacks several exons (eg. including the first two coding exons, the 5th exon). Thus, we tested the presence of sequence corresponding to the 3th exon of NAIP (NG_008724.1, sequence between 9331-9456) in human chromosome 5. GENOMIN found this sequence in two regions which correspond to NAIP gene and NAIP pseudogene (Figure 2). The Blast of this sequence against human genome confirms this result and the absence of this sequence in other regions of human genome.

The sequencing of the human insulin gene was a landmark in the genetic research[25]. Following studies demonstrated the complex structure of the human insulin promoter[26–30]. This gene has been selected because it known that it is associated both with Type 1 and Type 2 diabetes[31,32]. According to our view regarding the pathogenesis of diabetes, the high number of genes associated with this broad syndrome includes not only the secretory molecules of the β cells (pre-proinsulin/proinsulin/

insulin and pre-proamylin/proamylin/amylin) but also the machinery of their inclusion in the sophisticated secretory vesicles whose final maturation take place by the close cooperation between the endoplasmic reticulum / Golgi Apparatus and the cytoplasmic milieu[27–29]. Only by such cooperation the secretory vesicles can be not only promptly and efficiently exocited but also the response will be also proportional with the level of stimulus.

We used GENOMIN to test the presence of sequence corresponding to different regulatory elements like the negative regulatory element (NRE) (5'GAGACATTTGCCCCCAGCTGT sequence which lies between –279 to –258 nucleotides) and the E2 motif (GCCACCGG starting at –239 position) in the sequence of human insulin gene.

The software indicated the presence and the order of these sequences in the region between 2216015 and 2210045 from *Homo sapiens* chromosome 11 (alternative assembly Celera whole genome shotgun sequence) as was expected.

When in 1973-74 has been reported the first association between type 1 diabetes with HLA related genes[31], launched the immunogenetic hypothesis of this phenotype of diabetes, some enthusiastic researchers claimed that its cure is very close, "here over the corner". The doubt expressed by one of us[28], has been fully confirmed. In the decades which followed, nothing happened in this direction. A new hope raised in early '90, when genomic sequences begin. First, of *Haemophilus influence* (middle of 95) and than, year by year, for *S. Cervisae* , *E. Coli, C. Elegance, D. Melanogaster* and in 2001 the first draft of *Homo Sapiens*. In 2003 and once again in 2010, this draft has been revised. Meanwhile, at the end of first decade of the new century, several Genome Wide Scans (GWS) studies have been

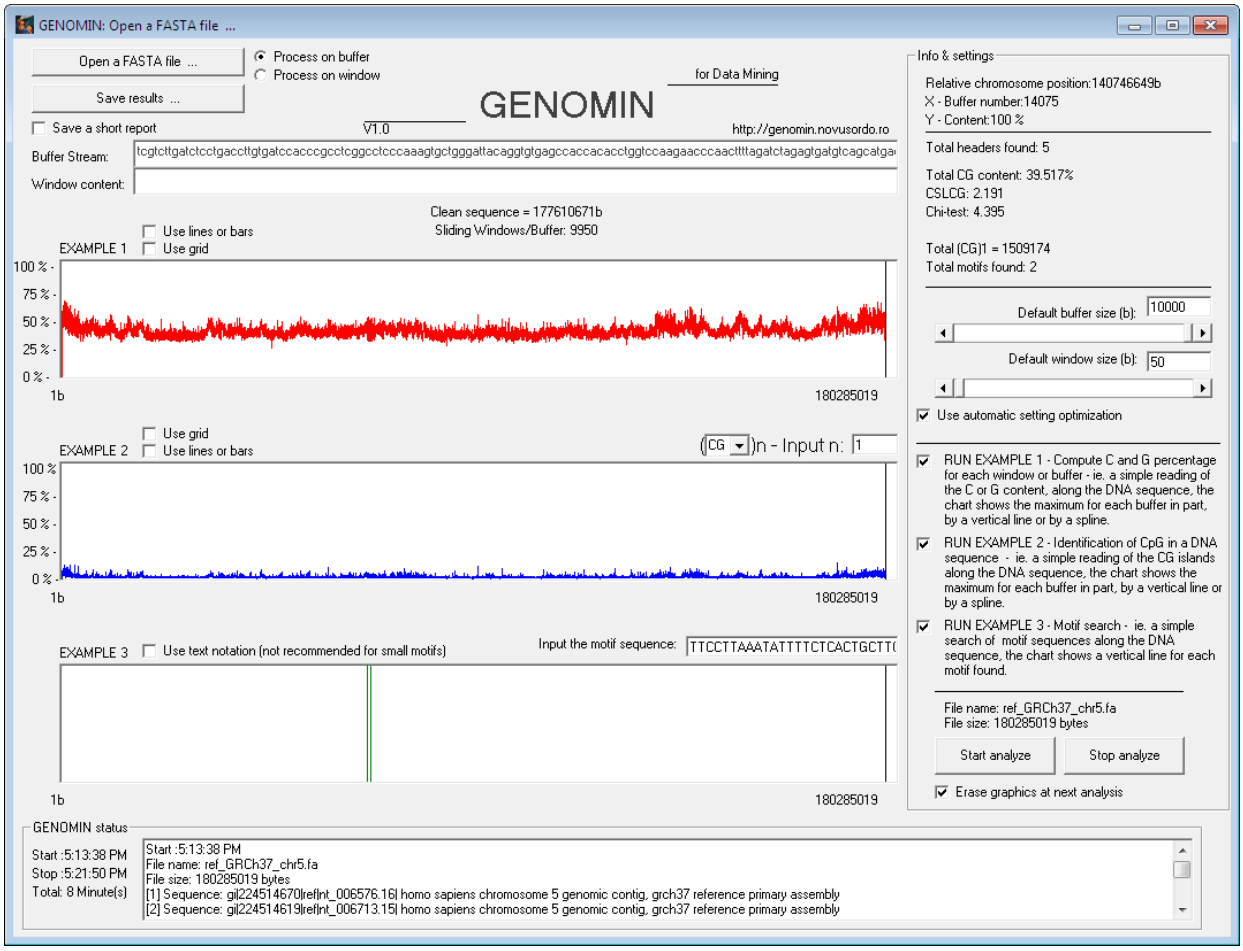

Fig. 2. The compiled implementation of GENOMIN software for read the Cytosine and Guanine content (the first chart inside the screen capture), (CG)n (the second chart) and motif sequences (the third chart) along the "ref_GRCh37_chr5.fa" file which contains the DNA sequence of chromosome 5 from *Homo sapiens sapiens* [3]. The two vertical green lines from the third chart, represent the 3th exon of NAIP gene and pseudogene.

obtained by various groups of researchers increasing the number of genes associated not only with diabetes, but also with obesity or other metabolic derangements[33–35]. It is obvious now that not the number of genes is important, but their complex arrangement. Moreover, an alternative mode of gene expression allows the production of more than 1 protein from a single gene. In addition, more than 50% of our genome consists of short repeated sequences. Within the human population are millions of single base differences (SNPs – *Single Nucleotide Polymorphisms*) making each human to differ to the next by ~ 1 base pair in every 1000 or even 500 bp. Here can be found the identity and unicity of each human being and of course, of the above mentioned metabolic disorders.

Monthly, maybe daily the international human genome data base increases, sometimes with brute information. On the other hand, the full screening of the SNPs in entire genome is in fact based on the *hypothesis free*. That means there is no assumption on gene or a specific genome region known to be involved. That is why the development of new software tools is needed in order to put order in such unexpected explosion of information with appearance of chaos.

Unlike other standard bioinformatics tools, in which software programs are guided by developers in some limits, GENOMIN platform is limited only by the researcher programming skills. The results of our tests have been shown that GENOMIN can perform various tests on large sequences files and can work with different algorithms used in biology.

ACKNOWLEDGEMENTS

This work represents a part of the Research Project PNII Partnerships 42-161/2008 from the Romanian Ministry of Education and Research. This paper is partially supported by the Sectoral Operational Programme Human Resources Development, financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/89/1.5/S/64109.

REFERENCES

1. Lipman DJ, Pearson WR. Rapid and sensitive protein similarity searches. Science 1985; 227 (4693):1435-1441.
2. FASTA format description, (Accessed October 01, 2010, at site <http://www.ncbi.nlm.nih.gov/blast/fasta.shtml>).
3. NCBI RefSeq Database, (Accessed October 01, 2010, at site < ftp://ftp.ncbi.nih.gov/genomes/>).
4. Donald E. Walker. Knowledge resource tools for accessing large text files. Proceedings of the Conference on Theoretical and Methodological Issues. In Machine Translation of Natural Languages, Colgate University, Hamilton, New York, August 14-16, 1985
5. Sayers EW, Barrett T, Benson DA, Bolton E, Bryant SH, *et al.*, Database resources of the National Center for Biotechnology Information. Nucleic Acids Res 2010;38(Database issue):D5-16.
6. TT virus genotype 1a DNA, complete genome (accession number: AB017610.1), (Accessed October 06, 2010, at site <http://www.ncbi.nlm.nih.gov/nuccore/5478532?report=fasta>) .
7. De Fonzo V, Aluffi-Pentini F, Parisi V. Hidden Markov Models in Bioinformatics. Current Bioinformatics 2007;2:49-61.
8. Cristea PD. Conversion of nucleotides sequences into genomic signals. J Cell Mol Med 2002;6(2):279-303.
9. Greenbaum D, Luscombe NM, Jansen R, Qian J, Gerstein M. Interrelating different types of genomic data, from proteome to secretome: 'oming in on function. Genome Res 2001;11(9):1463-1468.
10. Chiang JH, Shin JW, Liu HH, Chin CL. GeneLibrarian: an effective gene-information summarization and visualization system. BMC Bioinformatics 2006;7:392.
11. Gans JD, Wolinsky M. Genomorama: genome visualization and analysis. BMC Bioinformatics 2007;8:204.
12. Durand P, Canard L, Mornon JP. Visual BLAST and visual FASTA: graphic workbenches for interactive analysis of full BLAST and FASTA outputs under MICROSOFT WINDOWS 95/NT. Comput Appl Biosci 1997;13(4):407-413.
13. Mathé C, Sagot MF, Schiex T, Rouzé P. Current methods of gene prediction, their strengths and weaknesses. Nucleic Acids Res 2002;30(19):4103-4117.
14. Zheng D, Frankish A, Baertsch R, Kapranov P, Reymond A, Choo SW, Lu Y, Denoeud F, Antonarakis SE, Snyder M, Ruan Y, Wei CL, Gingeras TR, Guigó R, Harrow J, Gerstein MB. Pseudogenes in the ENCODE regions: consensus annotation, analysis of transcription, and evolution. Genome Res 2007;17(6):839-851.
15. Fourment M, Gillings MR. A comparison of common programming languages used in bioinformatics. BMC Bioinformatics 2008;9:82.
16. David I. Schneider, Computer Programming Concepts and Visual Basic. ISBN 0-536-60446-0, 1999.
17. Han L, Su B, Li WH, Zhao Z. CpG island density and its correlations with genomic features in mammalian genomes. Genome Biol 2008;9(5):R79.
18. Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, *et al.*, Initial sequencing and analysis of the human genome. Nature 2001;409(6822):860-921.
19. Guigó R, Flicek P, Abril JF, Reymond A, Lagarde J, Denoeud F, Antonarakis S, Ashburner M, Bajic VB, Birney E, Castelo R, Eyras E, Ucla C, Gingeras TR, Harrow J, Hubbard T, Lewis SE, Reese MG. EGASP: the human ENCODE Genome Annotation Assessment Project. Genome Biol 2006;7(1):S2.1-31.
20. Zavolan M, van Nimwegen E. The types and prevalence of alternative splice forms. Curr Opin Struct Biol 2006;16(3):362-367.
21. Bulyk ML. Computational prediction of transcription-factor binding site locations. Genome Biol 2003;5 (1):201.
22. Gagniuc P, Cimponeriu D, Panduru N.M., Stavarachi M., Toma M, Ionescu-Tîrgovişte C, Gavrilă L. A sensitive

method for detecting dinucleotide islands and clusters through depth analysis, RJDNMD, Vol. 18, No. 2: 165-170, 2011.
23. Liu F, Tøstesen E, Sundet JK, Jenssen TK, Bock C, Jerstad GI, Thilly WG, Hovig E. The human genomic melting map. PLoS Comput Biol 2007;3(5):e93.
24. Courseaux A, Richard F, Grosgeorge J, Ortola C, Viale A, Turc-Carel C, Dutrillaux B, Gaudray P, Nahon JL. Segmental duplications in euchromatic regions of human chromosome 5: a source of evolutionary instability and transcriptional innovation. Genome Res 2003;13(3):369-381.
25. Bell GI, Pictet RL, Rutter WJ, Cordell B, Tischer E, Goodman HM. Sequence of the human insulin gene. Nature 1980;284(5751):26-32.
26. Hay CW, Docherty K. Comparative analysis of insulin gene promoters: implications for diabetes research. Diabetes 2006;55(12):3201-3213.
27. Ionescu-Tîrgovişte C. A short personal view on the pathogenesis of diabetes mellitus, Proc. Rom. Acad., Series B, 2010, 3, p. 219–224.
28. Ionescu-Tîrgovişte C., Guja C. Proinsulin, proamylin and the beta cell endoplasmic reticulum: the key for the pathogenesis of different diabetes phenotypes, Proc. Rom. Acad., Series B, 2007, 2, p. 113–139.
29. Ionescu-Tirgoviste C., Despa F. Biophysical alteration of the secretory track in b-cells due to molecular overcrowding: the relevance for diabetes, Integrative Biology, 2010, DOI: 10.1039/c0ib00029a.
30. Ionescu-Tîrgoviste C. Proinsulin as the possible key in the pathogenesis of type 1 diabetes, Acta Endocrinologica (Buc), 2009, vol. V, no. 2, p. 233 - 249.
31. Nerup J., Platz P, Andersen OO, Christy M, Lyngsoe J, Poulsen JE, Ryder LP, Nielsen LS, Thomsen M, Svejgaard A. HLA antigens and diabetes mellitus. Lancet; 1974, ii:864-866;
32. Melloul D., Marshak S., Cerasi E.: Regulation of insulin gene transcription. Diabetologia 2002; 45:309-326.
33. Stančáková A., Paananen J., Soininen P., Kangas A.J., Bonnycastle L.L., Morken M.A, Collins F.S., Jackson A.U., Boehnke M.L., Kuusisto J., Ala-Korpela M., Laakso M. Effects of 34 risk loci for type 2 diabetes or hyperglycemia on lipoprotein subclasses and their composition in 6, 580 nondiabetic Finnish men. Diabetes 2011, 60: 1608-1616.
34. Li S., Zhao J.H., Luan J., Langenberg C., Luben R.N., Khaw K.T., Wareham N.J., Loos R.J.F. Genetic Predisposition To Obesity Leads To Increased Risk Of Type 2 Diabetes. Diabetologia 2011, 54: 776-782.
35. Ramos E., Chen G., Shriner D., Doumatey A., Gerry N.P., Herbert A., Huang H., Zhou J., Christman M.F., Adeyemo A., Rotimi C. Replication Of Genome-Wide Association Studies (Gwas) Loci For Fasting Plasma Glucose In African-Americans. Diabetologia 2011, 54: 783-788.