Selçuk Üniversitesi *ISSN 1302/6178* *Journal of Technical-Online*
Teknik Eğitim Fakültesi *Volume 15, Number:2-2016*
Teknik-Online Dergi
Cilt 15, Sayı:2-2016

AN IMPLEMENTATION FOR PERFORMING A COMPUTER BASED MUTATION ANALYSIS

Brwa ABUBAKER[1], Halgurd MOHAMMED[2], Rıdvan SARAÇOĞLU[1,*]

[1]Yüzüncü Yıl University, Engineering and Architecture Faculty, Electric-Electronics Engineering Department, Van TURKEY

[2]Yüzüncü Yıl University, Faculty of Medicine, Medical Biology Department, Van TURKEY

brwa.pshdary@gmail.com, mhalgurd@ymail.com, *ridvansaracoglu@yyu.edu.tr

Abstract

The history of Mutation Analysis can be sketched back from 1971 by Richard Lipton. It is vital to identify the variations occurred in DNA due to mutation. The aim of this work is to develop a new software that helps to predict the mutated sequence position found between the any two sequences whether it maybe DNA or Protein or it may be both. Moreover this approach is most effective and accurate to analyze sequences. The software is developed that helps to provide necessary input and get desired output. The output file will show the position were the mutation occur for protein 1 mutation occur in 1 for K and W and for C mutation occur in 40 position. Thus, the system runs to progress quality of testing and provide advance efficiency by means of various mutation operators. Computerized mutation analysis is performed without manual intervention.

Keywords: Mutation Analysis, Computerized mutation analysis, DNA or Protein

BİLGİSAYAR TABANLI MUTASYON ANALİZİ İÇİN BİR UYGULAMA

Özet

Mutasyon analizi tarihi 1971 yılında Richard Lipton tarafından yapılan çalışmalara dayanmaktadır. Mutasyon nedeniyle DNA içerisindeki oluşan varyasyonların belirlenmesi kritik önem taşımaktadır. Bu çalışmanın özü; DNA, Protein veya her ikisi de olabilen herhangi iki sıra arasında bulunan mutasyon geçirmiş pozisyonların tahmin edilmesine yardımcı olacak yeni bir yazılım geliştirmektir. Üstelik sıra analizi için çok verimli ve doğru sonuç üreten bir yaklaşımdır. Bu yazılım, gerekli girişlerin kolayca sağlaması ve arzu edilen çıkışların alınmasına yardımcı olacak şekilde geliştirilmiştir. Çıktı dosyası, 40 pozisyon içindeki 1 pozisyondaki oluşan K,W ve C mutasyonunun yerini gösterecektir. Böylece bu sistemle, kaliteli bir test süreci gerçekleştirilmekte ve çeşitli mutasyon operatörleri vasıtasıyla verimlilikte ilerleme sağlanmaktadır. Bilgisayar tabanlı mutasyon analizi, manüel müdahale olmaksızın gerçekleştirilmiş olmaktadır.

Anahtar Kelimeler: Mutasyon Analizi, Bilgisayarlı mutasyon analizi, DNA veya Protein

1. Introduction

The history of Mutation Testing can be sketched back from 1971 by Richard Lipton [1]. The birth of the field can also be identified in other papers published in the late 1970s by DeMillo et al. [2] and Hamlet [3].It is vital to identify the variations occurred in DNA due to mutation. For that genetic code which is used plays a crucial role. DNA is a major controller of ON/OFF mechanism of genes. Some parts of DNA are not having any functional properties and some have the properties of translation to protein.When there is an error like a base deleted or added or a wrong base incorporated in the sequence of DNA, it is called a mutation.

Existing nucleic acid molecules in a living organism acts as a genetic template to transfer the genetic info from one generation to the next. Nucleic acid molecules are organized as genes which code for a particular phenotype via specific proteins and the gene expression is regulated by both external and internal factors which aid the developmental process of an organism. This relation between genes and proteins forms the “central dogma of life".

The protein is having complete set of amino acids and every protein has unique amino acid arranged in a specific sequence. The information to synthesize proteins with unique amino acid sequence is provided by the nucleic acid present within the nucleus. In a pre-set sequence, DNA present in the nucleus give rise to the specific RNA sequence and that in turn guide the cellular machinery to synthesize protein.

The genetic code is conventional information that translates the information encoded in genetic material into proteins in living cells. The DNA codes with four letters A, T, G, and C. These protein coding DNA are said to be Codons. These codons are a group of three adjacent nucleotides specify the signals to protein. The stop codon implies the completion of the afresh fabricated protein.

Many Computational program design languages as a white box unit test method. For example, FORTRAN programs [4-6], Ada programs [7], [8], C programs [9-11], Java programs [12-14], C# programs [15-19], SQL code [20, 21] and Aspect programs [22, 23].C# is a modest, object-oriented programming language established by Microsoft and permitted by European Computer Manufacturers Association and InternationalStandards Organization. It is based on C and C++ programming language [16].

It was developed by Anders Hejlsberg and his team using .Net Framework. C# is intended for Common Language Infrastructure (CLI), consists of the executable code and runtime situation that permits various high-level languages on different computer platforms and architectures.

The reasons behind C# a widely used professional language is modern with well-structured language, object as well as component oriented, produce efficient programs, and compile variety of platforms.

The .Net framework applications are multi-platform applications. These has been applicable for C#, C++, Visual Basic, Jscript, COBOL, etc., for access the framework as well as converse with each other[18]. The .Net framework contains enormous library codes used by the client languages such as C#. Some components of .Net framework are Common Language Runtime, ASP.Net and ASP.Net AJAX, etc.

C# source code files can be made using a basic text editor, like Notepad, and compile the code into assemblies using the command-line compiler, which is again a part of the .NET Framework. Mono is an open-source version of the .NET Framework which includes a C# compiler and runs on several operating systems, including various flavors of Linux and Mac OS.

The purpose of this work is to develop a new software that helps to predict the mutated sequence position found between the any two sequences of DNAand those sequences will processed for translation to Protein sequences. It is possible to track mutation in protein sequences as well. Moreover it most effective and accurate to analyses sequences. The software is developed based on C# Program language that helps to provide necessary input and get desired output.

2. Materials and Methods

2.1.DNA Matching

DNA sequenceis fabricated with four bases (A, C, T, and G), anwell-organized fixed-length encoding system [24] can be used.In molecular biology, DNA sequences carryvital information foreach species and a comparison between DNA sequences is an interesting and more complicated. There are numerouscomparison tools to provide approximatematching. Our DNA matching algorithm are fast matching algorithm to match lengthy sequences in fastest approach.

2.2.Implementation of Mutation Analysis Program

FASTA format: A sequence book in a FASTA format including (first line) a single-line description (sequence name), followed by line(s) or (second line)of sequence data. The first character of the denote line is a greater-than (">") symbol. like that

>HSBGPG Human gene for bone gla protein (BGP)

GGCAGATTCCCCCTAGACCCGCCCGCACCATGGTCAGGCATGCCCCTCCTCATC

GCTGGGCACAGCCCAGAGGGT

FASTA can be utilized to deduce functional and evolutionary linkages amidst sequences also help identify members of gene families [25].

“Protein”

- ✓ Protein to protein FASTA.
- ✓ Protein to protein Smith–Waterman (ssearch).
- ✓ Global protein to protein (Needleman–Wunsch) (ggsearch)
- ✓ Global/local protein to protein (glsearch)
- ✓ Protein to protein with unordered peptides (fasts)
- ✓ Protein to protein with mixed peptide sequences (fastf)

“Nucleotide “

- ✓ Nucleotide to nucleotide (DNA/RNA fasta)
- ✓ Ordered nucleotides vs nucleotide (fastm)
- ✓ Unordered nucleotides vs nucleotide (fasts)

In FASTA algorithm Nucleotide or protein sequence is taken as input.

The hurry and sensitivity is controlled by the parameter called ktup, which specifies the gauge of the word. This program uses the word hits to identify potential matches between the query sequence and database sequence (Fig. 2.1). Initially it review for segment's containing several there about hits.

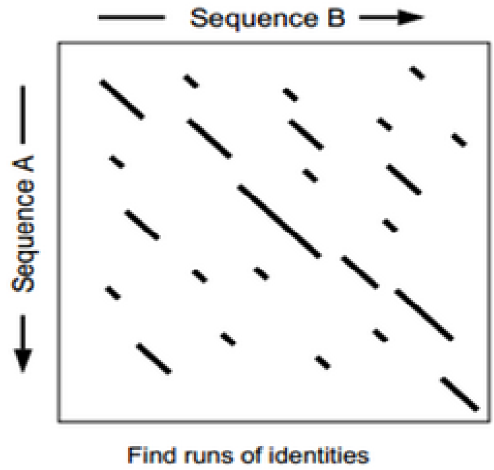


Fig. 2.1. FASTA algorithm (FASTA Alignments)

FASTA algorithm has Dot matrix comparisonsWords matches in 2 sequences I & J can be represented as a dot matrix (as shown Fig. 2.2), thus

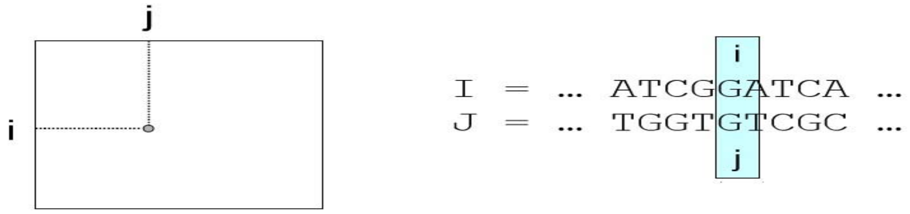


Fig. 2.2 Dot matrix comparisons

The flowchart of program’s algorithm is shown in Figure 2.3 in that the inputer sequences of DNA are in the form of FASTA format. Once the DNA is in FASTA format then the comparison between the two sequences has to be done based on color differences. Followed by transcription and translation to RNA and Protein. Then comparison between these two mutated protein sequences has to be analysed. The result has to be shown in data grid view.

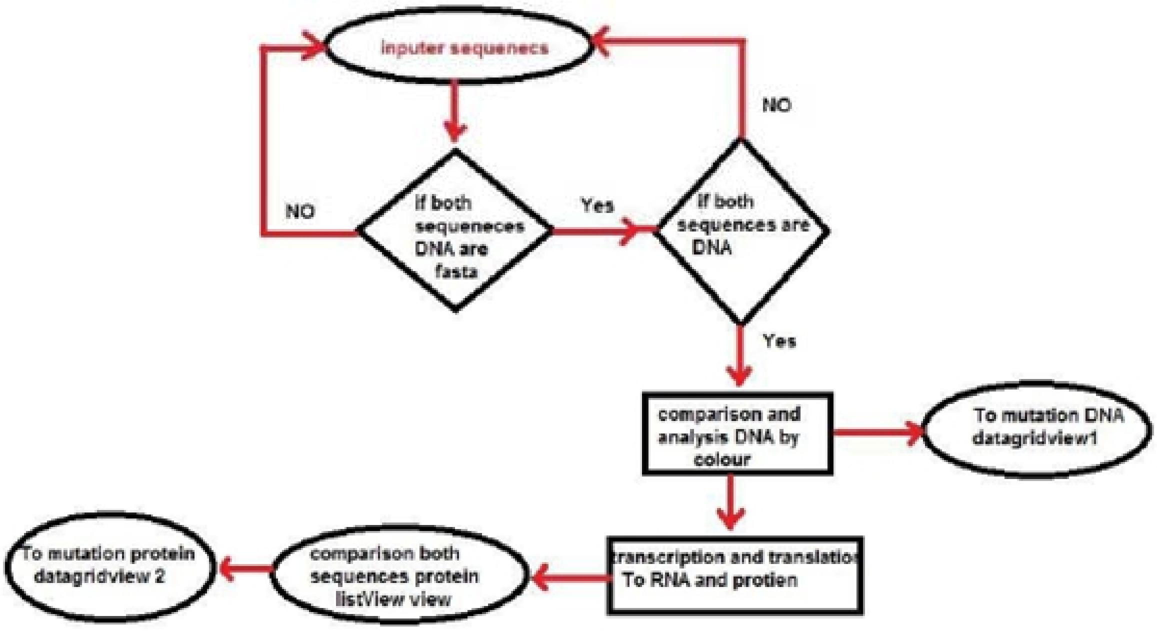

inputer sequenecs
NO
If both sequeneces DNA are fasta
Yes
If both sequences are DNA
NO
Yes
comparison and analysis DNA by colour
To mutation DNA datagridview1
transcription and translatio To RNA and protien
comparison both sequences protein listView view
To mutation protein datagridview 2

Fig. 2.3. Overview of program

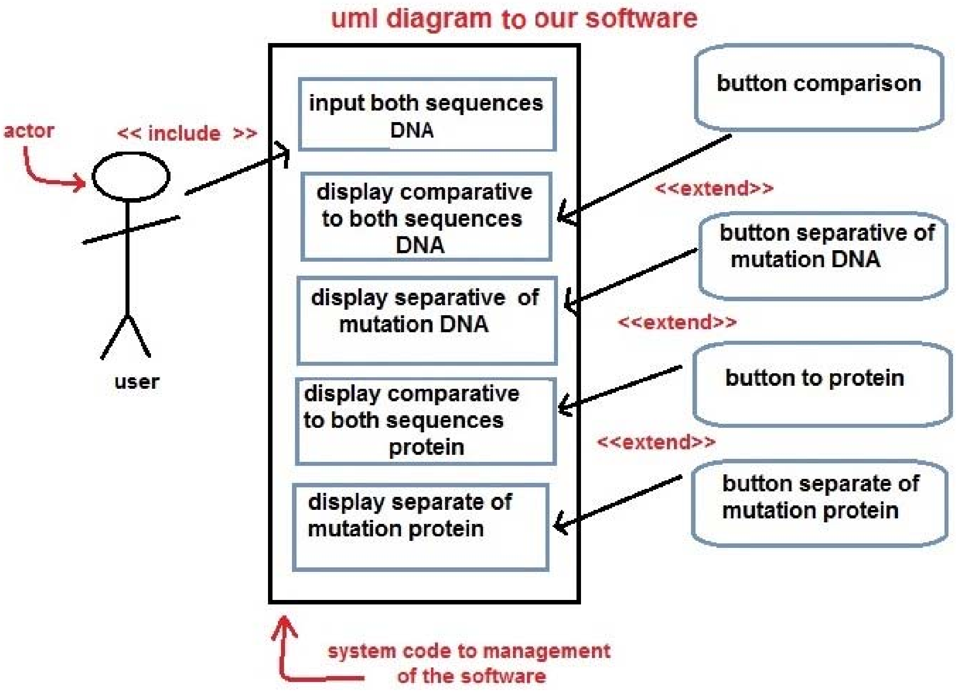

uml diagram to our software
actor
<< include >>
user
input both sequences DNA
display comparative to both sequences DNA
display separative of mutation DNA
display comparative to both sequences protein
display separate of mutation protein
button comparison
<<extend>>
button separative of mutation DNA
<<extend>>
button to protein
<<extend>>
button separate of mutation protein
system code to management of the software

Fig. 2.4. UML daigram of our software

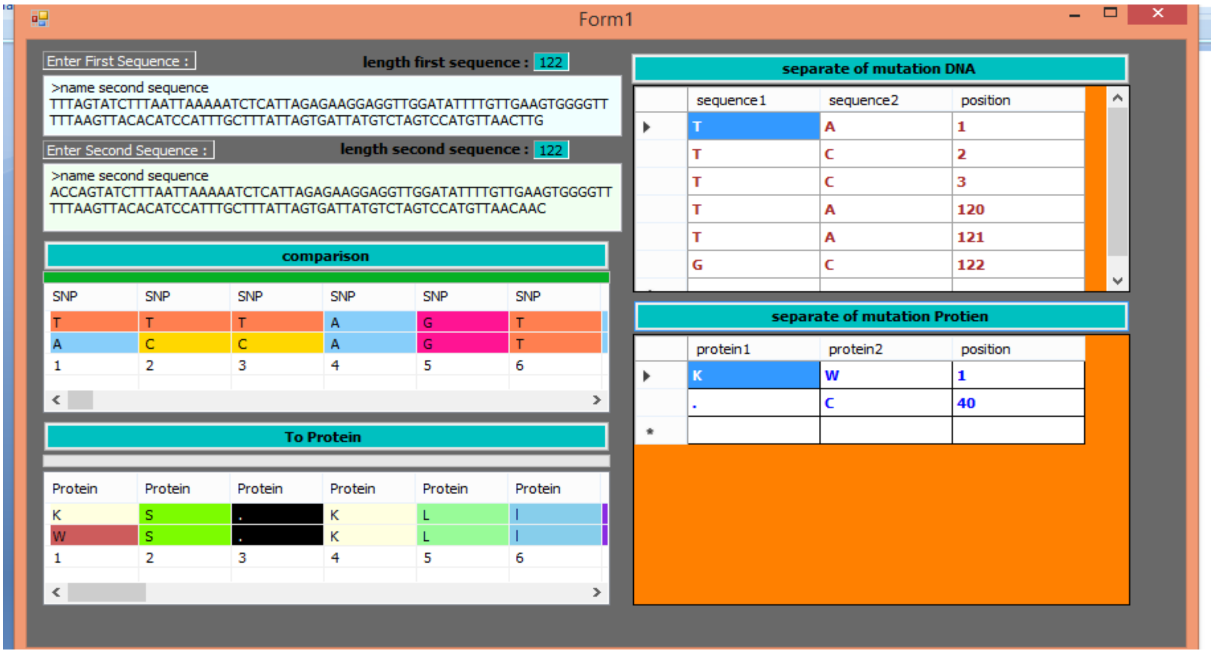
UML daigram of our software is shown in Fig. 2.4.

2.3.Retrieve sequences from database

The sequence which is going to be analyzed has to be retrieved from the specific proteins database for analysis. Important point is sequences are must be in the form of FASTA format. Those FASTA sequences are imported to our software by using a suitable cods.

3. Experimental Results

The complete view of our software in that the sequences which are going to be sequenced are retrieved and paste to the following box and select RUN. Then comparison will start processing once the process is complete the result will show in right side of the dialogue box (as shown in Fig. 3.1).



Form1

Enter First Sequence : length first sequence : 122

```
>name second sequence
TTTAGTATCTTTAATTAAAAATCTCATTAGAGAAGGAGGTTGGATATTTTGTTGAAGTGGGGTT
TTTAAGTTACACATCCATTTGCTTTATTAGTGATTATGTCTAGTCCATGTTAACTTG
```

Enter Second Sequence : length second sequence : 122

```
>name second sequence
ACCAGTATCTTTAATTAAAAATCTCATTAGAGAAGGAGGTTGGATATTTTGTTGAAGTGGGGTT
TTTAAGTTACACATCCATTTGCTTTATTAGTGATTATGTCTAGTCCATGTTAACAAC
```

comparison

SNP	SNP	SNP	SNP	SNP	SNP
T	T	T	A	G	T
A	C	C	A	G	T
1	2	3	4	5	6

To Protein

Protein	Protein	Protein	Protein	Protein	Protein
K	S	.	K	L	I
W	S	.	K	L	I
1	2	3	4	5	6

separate of mutation DNA

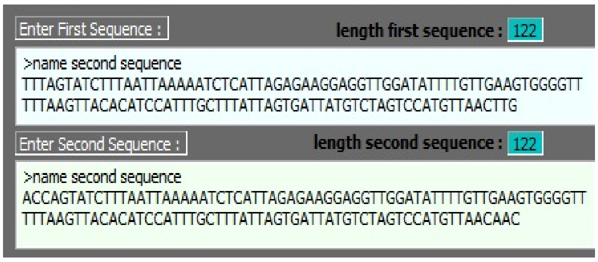
sequence1	sequence2	position
T	A	1
T	C	2
T	C	3
T	A	120
T	A	121
G	C	122

separate of mutation Protien

protein1	protein2	position
K	W	1
.	C	40

Fig. 3.1. Represent the Whole Software

output file



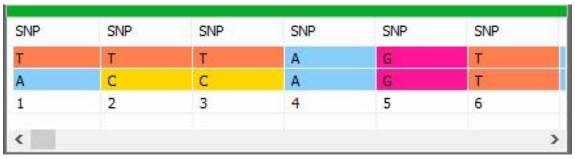
Enter First Sequence : length first sequence : 122

```
>name second sequence
TTTAGTATCTTTAATTAAAAATCTCATTAGAGAAGGAGGTTGGATATTTTGTTGAAGTGGGGTT
TTTAAGTTACACATCCATTTGCTTTATTAGTGATTATGTCTAGTCCATGTTAACTTG
```

Enter Second Sequence : length second sequence : 122

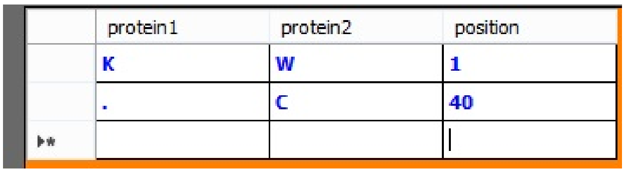
```
>name second sequence
ACCAGTATCTTTAATTAAAAATCTCATTAGAGAAGGAGGTTGGATATTTTGTTGAAGTGGGGTT
TTTAAGTTACACATCCATTTGCTTTATTAGTGATTATGTCTAGTCCATGTTAACAAC
```

Simple sequences for analysis



SNP	SNP	SNP	SNP	SNP	SNP
T	T	T	A	G	T
A	C	C	A	G	T
1	2	3	4	5	6

List View in C#



protein1	protein2	position
K	W	1
.	C	40

Data Grid View Output file

Fig. 3.2. Output file shows separate mutation of protein.

We select two sequences which are going for analysis is retrieved as a FASTA Format and the sequence has to be undergone for mutation analysis. Before that nucleotide sequences variation done by means of list view command. The thymine residues are in orange color, adenine residues are in blue color, guanine is in Rose and cytosine is in yellow. The output file provide the position were the mutation occur for protein 1 mutation occur in 1 for K and W and for Cmutation occur in 40 position (Fig. 3.2).

Compare between our software with another tool (by name Transcription and Translation Tool) is shown in Table 3.1.

Blast and Fasta are two algorithms these are utilized to compare sequences of amino acids,DNA, proteins and nucleotides of diverse species and look for the similarities. those genetic algorithms were written keeping speed in mind in order to as the data bank of the sequences swelled once DNA was isolated in the lab by the scientists in 1980s there increased a need to compare and find corresponding genes for more research at high speed.

Table 3.1 Comparison of Software

Our tool	**Transcription and Translation Tool**
Without internet is work	It is need internet to work
It is utilize FASTA format	It is utilize Plain sequence format
It could use color to DNA sequences	It could not use color to DNA sequences
It has account length of sequences DNA & protein	It hasn't account length of sequences
It can loading two sequences	It can loading only one sequence
It can separate mutation DNA sequences	It cannot separate mutation DNA sequences
It cannot display RNA , immediately DNA to protein	It will show RNA before protein
It could use color to protein sequences	It could not use color to protein sequences
It will show position to sequences DNA & protein	It can not

FASTA was the most vastly utilized protein and DNA sequence database search program next the coming of BLAST. It is identical with BLAST in many routes, and is still repeatedly utilized. Such as BLAST, it is a heuristic for approximating the Smith-Waterman algorithm, but utilizes diverse heuristic methods to raise speed. BLAST and FASTA as well utilize slightly different methods to calculate statistical significance. Our software has utilized FASTA therefore all software on FASTA format could not separate part of mutation for segment of DNA and segment of protein, on that our software was additional parts of mutation for proteins and nucleotides by best quality colour.

4. Conclusion

The purpose of the work is to perform a mutation analysis of each DNA sequencs followed by comparison to track the position as well, the structure of the sequances of DNA is 4 types of bases that symbolize by four letter A , C , G and T .this software colured all the bases of DNA sequences by different colour each colour indicates to special nucleotide as deep pink colour to G , gold to C , light sky blue to A and the coral to T that property of this software give the user details about the contain of each type of nucleotide after that translate the DNA toprotein and compare them also by means of this software.

This will be more accurate,also sequence of protein is symbolize by four letter A , C , G and U and each three symbolizes to one amino acid depend on the amino acid coden .also in this bioinformatics tool give each symbol special colour to indicate that four different characters lesstime, easy to predict those regions which are mutated. Thus, the system runs to progressquality of testing and provide advance efficiency by means of various mutation operators.Computerized mutation testing is performed without manual interventin.

In the biological science any change in the structure any DNA sequence allow to change in protein sequence and that maybe appear abonormalty in human body that called mutation .

In this work reslut of this software , it is simple to understanding from the user .if compare this software from speed and efficiency sides , it has high efficiency and much speed . And on the otherhand this software is work offline and easy to download on the windows system .

Acknowledgments

The authors are grateful for the support provided by YüzüncüYıl University.

References

[1] Mathur P. “Mutation Testing”, in Encyclopedia of Software Engineering, J. J. Marciniak, Ed., 1994, pp. 707–713.

[2] DeMillo RA, Lipton RJ, Sayward F G. “Hints on Test Data Selection: Help for the Practicing Programmer,” Computer, vol. 11, no. 4, pp. 34–41, April 1978.

[3] Hamlet RG, “Testing Programs with the Aid of a Compiler,” IEEE Transactions on Software Engineering, July 1977, 3(4): 279–290,

[4]. Acree, A. T., Budd, T. A., DeMillo, R. A. , Lipton, R. J., and Sayward, F. G., “Mutation Analysis,” Georgia Institute of Technology, Atlanta, Georgia, Technique Report GIT-ICS-79/08, 1979.

[5]. Budd TA, DeMillo RA, Lipton RJ, Sayward FG. “The Design of a Prototype Mutation System for Program Testing,” in Proceedings of the AFIPS National Computer Conference, vol. 74. Anaheim, New Jersey: ACM, 5-8 June 1978, pp. 623–627.

[6] Budd TA, Sayward FG. “Users Guide to the Pilot Mutation System,” Yale University, New Haven, Connecticut, Technique Report 114, 1977.

[7]. Bowser JH. “Reference Manual for Ada Mutant Operators,” Georgia Institute of Technology, Atlanta, Georgia, Technique Report GITSERC-88/02, 1988.

[8]. Offutt, A. J., Voas, J., and Payn, J., “Mutation Operators for Ada,” George Mason University, Fairfax, Virginia, Technique Report ISSE-TR-96-09, 1996.

[9]. Agrawal H, DeMillo RA, Hathaway B, Hsu W, Hsu W, Krauser EW, Martin RJ, Mathur AP, Spafford E. “Design of Mutant Operators for the C Programming Language,” Purdue University, West Lafayette, Indiana, Technique Report SERC-TR-41-P, March 1989.

[10] Delamaro ME, Maldonado JC, Mathur AP. “Interface Mutation: An Approach for Integration Testing,” IEEE Transactions on Software Engineering, May 2001, 27(3):228–247.

[11] Vilela P, Machado M, Wong WE. “Testing for SecurityVulnerabilities in Software,” in Software Engineering and Applications, 2002.

[12] Chevalley P. “Applying Mutation Analysis for Object-oriented Programs Using a Reflective Approach,” in Proceedings of the 8th Asia-Pacific Software Engineering

Conference (APSEC 01), Macau, China,4-7 December 2001, p. 267.

[13] Chevalley P, Th´evenod-Fosse P. “A Mutation Analysis Tool for Java Programs,” International Journal on Software Tools for Technology Transfer, November 2002, 5(1):90–103.

[14] Ma, Y. S., Offutt, A. J. and Kwon, Y. R., “MuJava: An Automated Class Mutation System,” Software Testing, Verification & Reliability, vol. 15, no. 2, pp. 97–133, June 2005.

[15] Derezi´nska A. “Object-oriented Mutation to Assess the Quality of Tests,” in Proceedings of the 29th Euromicro Conference, Belek,Turkey, 1-6 September 2003, pp. 417– 420.

[16] Derezi´nska A. “Advanced Mutation Operators Applicable in C# Programs,” Warsaw University of Technology, Warszawa, Poland, Technique Report, 2005.

[17] Derezi´nska A. “Quality Assessment of Mutation Operators Dedicated for C# Programs,” in Proceedings of the 6th International Conference on Quality Software (QSIC’06), Beijing, China, 27-28 October 2006.

[18] Derezi´nska A, Szustek A. “CREAM- A System for Object-Oriented Mutation of C# Programs,” Warsaw University of Technology, Warszawa, Poland, Technique Report, 2007.

[19] Derezi´nska A, Szustek A. “Tool-Supported Advanced Mutation Approach for Verification of C# Programs,” in Proceedings of the 3th International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX’08), SzklarskaPorˆeba, Poland, 26-28 June 2008, pp. 261–268.

[20] Shahriar H, Zulkernine M. “MUSIC: Mutation-based SQL Injection Vulnerability Checking,” in Proceedings of the 8th International Conference on Quality Software (QSIC’08), Oxford, UK, 12-13 August 2008, pp. 77–86.

[21] Tuya J, Cabal MJS, de la Riva C. “SQLMutation: A Tool to Generate Mutants of SQL Database Queries,” in Proceedings of the 2nd Workshop on Mutation Analysis (MUTATION’06). Raleigh, North Carolina: IEEE Computer Society, November 2006, p. 1.

[22] Anbalagan P, Xie T. “Automated Generation of Pointcut Mutants for Testing Pointcuts in AspectJ Programs,” in Proceedings of the 19th International Symposium on Software Reliability Engineering (ISSRE’08). Redmond, Washingto: IEEE Computer Society, 11-14 November 2008, pp. 239–248.

[23] Ferrari FC, Maldonado JC, Rashid A. “Mutation Testing for Aspect-Oriented Programs,” in Proceedings of the 1st International Conference on Software Testing, Verification, and Validation (ICST ’08). Lillehammer, Norway: IEEE Computer Society, 9-11 April 2008, pp. 52–61.

[24] Kim JW, Kim E, Park K. Fast matching method for DNA sequences. InCombinatorics, Algorithms, Probabilistic and Experimental Methodologies, volume4614 of LNCS, pages 271–281, 2007.

[25] Setubal & Meidanis. Introduction to Computational Molecular Biology, PWS Publishing Company, 1997.Chapter3 .