Fault Detection Techniques Prioritization using Bee Colony Optimization and then Comparison with Ant Colony Optimization

Mandeep Kaur Bedi
Department of CSE & IT
LPU, India

Sheena Singh
Department of CSE & IT
LPU, India

ABSTRACT

Research in software testing has experienced a significant growth in recent years. One topic of special interest is fault detection techniques to reduce human interference and detect maximum faults. Priority is given to these techniques so that only higher priority techniques should be used instead of using individual techniques of lower priority. Bee Colony Optimization based upon natural phenomena algorithm is used in it to find out the best results. The work presented in this paper expresses the idea of implementing of fault detection techniques to provide them priority using bee colony optimization and ant colony optimization and then compare their results with ant colony optimization. The results shows bee colony optimization is better than ant colony optimization and consume less time as compare to manual process.

Keywords

Bee Colony Optimization, Ant Colony Optimization, Fault Detection techniques.

1. INTRODUCTION

Software Engineering is the discipline of computer science which applies engineering principles to create, operate, modify and maintain of software components. Software engineering can be subdivided into many sub disciplines. Software testing is one of them. It forms the important component of software quality. Testing is used to find bugs. Testing is static and dynamic process both. Testing is further divided into many subcategories like unit testing, black-box testing, white-box testing, system testing etc. To detect faults is also come under testing. Throughout the development of the software, verification and validation is the main phases. The

main aim of the verification is to check whether the system is working according to its specification. Validation aims is whether the system behaves like the customer's requirements

[14].The purpose of the fault detection techniques is to focus at the user point of view to inspect and test the technique. Software testing cannot be done until the implement phase is completed. It is necessary to find the faults at early phases to avoid the cost. Testing is done at all the phases of the development. At all the different stages new faults are find out. So there are many types of fault detection techniques available in software testing. These techniques can be in static and dynamic in nature [1].

Fault detection techniques [14] can be broader classified into various categories. These are as follow:

- Specification Based Testing Techniques
- Experimental Based Testing Techniques
- Structure Based Testing Techniques
- Machine Learning.

These categories are used to detect faults. Further these categories are sub divided into many categories.

1.1 SPECIFICATION BASED TESTING TECHNIQUES:

It is also known as black box testing. In this technique we do not know the internal structure of the system. It is based upon some functional and non-functional based specification. It has following types.

1.1.1. Equivalence Partitioning (EP): Equivalence Partitioning is a technique in which divide the software or system into groups or into smaller parts which are also behave in a same manner. The values in the equivalence partitioning should be valid and invalid. In valid data values are those values which are accepted and invalid data vales are those values which are rejected. If any value from the group is used, result should be same [12].

1.1.2 Boundary Value Analysis (BVA): It is very easy to apply and provide high results for fault finding. A valid boundary value is that which gives valid boundary and invalid boundary values is that which gives invalid boundary. Maximum and minimum values are its boundary values [1][14].

1.1.3 Decision Table Testing (DTC): Logical conditions are used in decision table. The result should be in the form of true or false and 1 and 0. It can be applied on those situations where actions of the results depend on several logical conditions like complex records [1].

*1.1.4 State Transition Testing (STT)***:** This testing shows the relationship between input and states and their valid and invalid transitions. It also shows the actions from transition change and their trigger time. Diagrammatic representation also require in it [1].

*1.1.5 Functional Testing (FT)***:** Functional testing is based upon requirements and functionality. No knowledge of internal structure is required. Data driven tests are used in it. It deals with the external behavior of the system [2].

1.2 EXPERIMENTAL BASED TESTING TECHNIQUES:

It is also known as white-box testing. In this we have the idea of the external structure of the code or system. It designs the structure of test cases and measure test coverage. Its sub categories are given below:

1.2.1 Test Coverage (TC): It measures the amount of the testing done by the set of tests. Then it also measures the coverage known as test coverage. It helps in finding faults which are not finding by set of tests. Code coverage is one of its types [1].

1.2.2 Decision Coverage and Decision Testing (DCDT) [1] [14]: Decision coverage executes both true and false conditions. Here each statement may have one or two possible outcomes. Decision coverage is stronger than statement coverage.

1.2.3 Block Coverage (BC): Block coverage is same as statement coverage. The basic difference is that in the block coverage sequence non-blocking statements are used to execute the statements except unit of code.

1.3 EXPERIENCE BASED TESTING TECHNIQUES [14]:

It is a technique which is based upon the person's knowledge and experience.

1.3.1 Error Guessing Techniques (EGT): It is formal test design technique which generates the lists of error and then generates test cases to reproduce them.

1.3.2 Exploratory Testing Techniques (ETT): Exploratory testing is also known as monkey testing. This technique is used when specification is weak and time is limited. Without creating test conditions, test scripts and test design, executions of test cases occur simultaneously.

1.4 MACHINE LEARNING:

1.4.1 Support Vector Machine Algorithm (SVM): SVM is a technique which is used to solve pattern recognition problem. It is supervised learning based upon the learning algorithms. It takes set of inputs and predicts it to given corresponds to output [3].

1.4.2 Decision Tree Algorithm (DTA): It is rule based classification which is used for prediction and classification Rules are represent by it which is human understandable and as a database knowledge is used.

1.4.3 Fuzzy Logic (FL): Fuzzy logic is logic of multivalve which defines intermediate values of conventional evaluations like black or white, it represents the system behavior [4]. It values not limit on 0 and 1 it lays between them. It is also known as diffuse logic [fl]. It can be used to interpreted neural networks.

1.4.4 Neural Networks (NN): A neural network is resembled with capabilities of brains. Neurons are used in it to learning and for calculations. Synaptic weights are also attaches with the neurons to store the capability of training data [3].

2. REVIEW OF LITERATURE:

Jovanovic Irena have explained different types of software testing methods and techniques[5].Two generally classified testing technique like black box testing and white box testing is explained in this paper. They also explained all other types of testing and their sub classes and find out their complexity. They also explained about automatic testing, structural testing, web testing, agile testing etc. and difference between them [5]. **Carina Anderson, Thomas Thelin, Per Runeso, Nina Dzamashvili [6]** has discussed about the detection of the faults using some testing techniques in a particular software. They tested a documentation corresponding to a program and evaluate two types of techniques.UBR is a usages based reading and UBT is a usage based testing. Both of these techniques are related to a reviewer of a testing. In this they take two different versions and find out 14 and 15 faults respectively using UBR and UBT. When fault detection techniques are combined than their effectiveness and efficiency increase. **Shivkumar Hasmukhrai Trivedi [2]** also explained many software testing techniques and their objectives. Later then explained about software life cycles and phases. What are the different methods to handle different types of testing techniques. In the last phase he mentioned about the detail of testing methodology. **Dr. Arvinder Kaur and Shivangi Goyal [7]** mentioned the bee colony optimization algorithm concept to find out the optimal path from number of other algorithm to get the best results at the end. This algorithm is used for the fault coverage. Here BCO algorithm is used to find the maximum faults in minimum execution time. Two examples are taken in it for comparison and two find the optimal path. This algorithm explores and exploits the path with the help of natural phenomena techniques. It is implemented in CPP compiler. But human interface is more in this work. **Bharti Suri and Shewta Singhal [8]** introduced the concept of ant colony optimization which is natural technique. It is used to explore the path which is close to the optimal solutions. This paper represented implementation of Test case selection and prioritization using ant colony optimization's c++ compiler is used which represent no. of fault covered and their execution time.**Wei liu and Sanjay Chwala [9]** discussed about the decision tree algorithm which is new technique and to make decision tree robust class confidence proportion measured is introduce. **Mandeep Kaur Bedi and Sheena Singh [10]** discussed the comparison between Bee Colony Optimization and Ant colony Optimization. They also discussed their advantages, disadvantages and their applications. At the last they concluded that BCO is better than ACO.

3. RELATED WORK:

This section represents an overview of the prioritization of the fault detection techniques and then bee colony optimization which is applied upon the fault detection techniques to find the optimal solution.

In the selection process we have choose data sets from the different experimental results of the fault detection techniques according to solution of our problem.

The datasets are based upon the fault detections approaches as we discussed above. In [9], proposed by **Wei Liu and Sanjay Chawla David A. Cieslak and Nitesh V. Chawla** discussed about the decision tree algorithm which is new technique and to make decision tree robust class confidence proportion (CPP) measured is introduce. They gave experiment result for error detection using CPP at the end. In [11], presented by **Xia**

Cai and Michael R. Lyu explained about the code coverage for their effectiveness. In this paper they evaluate the performance of code coverage using coverage testing and mutation testing and using different testing profiles to detect faults. In [12] discussed by **Stuart C. Reid** about the experiment which is compare EVP, BVA to tell about the effectiveness based upon the 20,000 lines of ADA code. He considered all the inputs value that satisfy and all the input values that cause fail of module and comparison for them and find out probability of detection. In [4], presented by **Praveen Ranjan Srivastava, Sirish Kuma**r *et.al* discussed about the estimatination of testing effort, selection of suitable test team, test cases designing, software execution test cases and examining the results produced by those executions. It indicates software development cost for testing, with the percentage for testing critical software being even higher. This paper makes an attempt using fuzzy logic to estimate reliable software testing effort. In [3] proposed by **Meenakshi Vanmali, Mark Last, Abraham Kandel** in which they discussed about neural network used as an automated oracle for testing software.BPA algorithm is applied upon the set of test cases. It is based upon the black box testing. The experimental results find out the faults using two layers neural network. In [13] explained by **Sujun Hua and Zhirong Sun** represented new approach for the recognition of patterns using Support vector Machine.SVM performance is better than the traditional approach.

3.1 Bee Colony Optimization:

Bee colony is optimization technique which is based upon the natural phenomena and find out optimal solution at the end. It is self organizing technique. Two types of bee available in the bee hive. Upon these two bees all the food collecting technique is depended. Scot bee is going outside in the search of the food and come back to the bee hive when out of energy. Waggle dance is performed in shape of digit 8 by the scout bee in the bee hive to communicate with the forager bee. With the communication forager bee came to know about the best quality of food in the direction of sun then follow the same path for the collection of the food. Scout bees explore the path where as forager bees exploit the path. A BCO algorithm is used to find out maximum number of faults [7] [10].

3.2 Ant Colony Optimization:

It is a metaheuristic technique which is based upon the natural phenomena. ACO is a probalistic technique which gives solution by using previous results. In this process each ant follow different path to reach to the destination and secrete pheromone liquid on the way to destination. The path which has the highest liquid pheromone is considered as the shortest path and all other ants follow the same path. So pheromone liquid is used to attract the other ant and update the latest information about the path [8] [10].

4. Implementation:

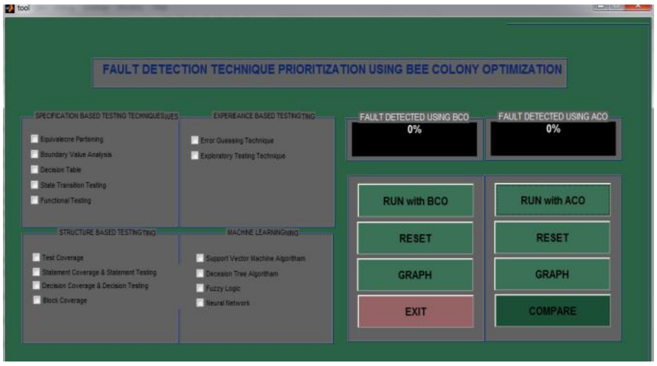
After the selection of datasets for fault detection techniques, then we have created a Graphical User Interface (GUI) automation tools in MATLAB. In the proposed system we want to reduce human interference and execute maximum number of faults. All the fault detection techniques are broader classified. Bee Colony Optimization algorithm is applied on all the fault detection techniques and then compare with ACO algorithm. In the GUI, we have result button which displays the result of each faults detection technique i.e. shows the result of faults covered. Result button display results for BCO and ACO differently.

Here we have different fault detection technique under broader classification. To prioritize these techniques one by one and check their result become time consuming process and manual interface more. We have to reduce human interface so that it consume minimum time and covers maximum faults. For this purpose we have to apply BCO algorithm upon all the techniques and prioritize them. So instead of applying one by one technique and check which is better, combine all the techniques from different classification (choose only one technique from all different category at a time) apply only higher priority techniques according to the table given below. Suppose that all the fault detection techniques are like food sources which are spread in different directions .Scout bees find out food source and forager bees collect the best quality food only among the different available food resources according to BCO algorithm. Similarly after applying BCO algorithm upon the fault detection techniques among the number of combination of different fault detection technique we choose only of best quality i.e. which cover maximum faults. In this help of ACO algorithm we can also find out faults following same above mention criteria but when compare it with ACO, it found that BCO algorithm is better than ACO algorithm i.e. BCO algorithm finds out more faults with same technique than ACO algorithm. Below is a table given which shows priorities of up to 21 from 162 priorities. The technique which covers maximum faults set as 1st priority and so on.

Table 1: Priority Table for Fault Detection Techniques using BCO

Priority	**Combination of Fault Detection** **Techniques**	**Result With BCO** **In Percentage**	**Result With ACO** **In Percentage**
1st	EP+EGT+SC&ST+ SVM	41.78	39.98
2nd	EP+ EGT + SC&ST + NN	41.51	39.41
3rd	DT+EGT + SC&ST + SVM	40.77	35.47
4th	EP+EGT+BC+SVM	40.58	36.97
5th	DT+EGT+SC&ST + NN	40.50	38.60
6th	EP + EGT + SC&ST + DTA	40.32	38.92
7th	EP+EGT+ BC + NN	38.80	36.40
8th	DCT + EGT + BC + SVM	39.57	37.47
9th	STT+EGT +SC&ST+ SVM	39.55	38.15
10th	DCT+EGT+SC&ST + DTA	39.31	38.11
11th	DT+EGT+BC + NN	39.30	36.30
12th	STT+EGT+ SC&ST + NN	39.28	37.58
13th	FT+ EGT + SC&ST +SVM	39.17	37.37

14th	FT+EGT+BC+ DTA	39.12	37.22
15th	BVA+EGT+SC&ST +SVM	39.07	36.97
16th	FT+ EGT + SC&ST + SVM	38.90	36.80
17th	BVA + EGT + SC&ST + SVM	38.80	36.40
18th	STT + EGT + BC + SVM	38.35	36.45
19th	DCT + EGT + BC +DTA	39.31	36.41
20th	STT + EGT + SC&ST + DTA	38.09	37.09
21th	FT + EGT + BC + SVM	37.97	37.47


tool

FAULT DETECTION TECHNIQUE PRIORITIZATION USING BEE COLONY OPTIMIZATION

SPECIFICATION BASED TESTING TECHNIQUESIUES
□ Equivalecne Partioning
□ Boundary Value Analysis
□ Decision Table
□ State Transition Testing
□ Functional Testing

EXPEREANCE BASED TESTINGTING
□ Error Guessing Technique
□ Exploratory Testing Technique

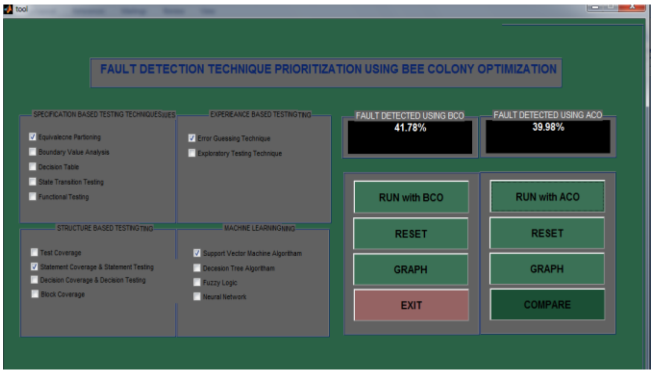
STRUCTURE BASED TESTINGTING
□ Test Coverage
□ Statement Coverage & Statement Testing
□ Decision Coverage & Decision Testing
□ Block Coverage

MACHINE LEARNINGNING
□ Support Vector Machine Algortham
□ Decesion Tree Algortham
□ Fuzzy Logic
□ Neural Network

FAULT DETECTED USING BCO: 0%

FAULT DETECTED USING ACO: 0%

Fig 1: Tool for fault detection technique prioritization using BCO


tool

FAULT DETECTION TECHNIQUE PRIORITIZATION USING BEE COLONY OPTIMIZATION

SPECIFICATION BASED TESTING TECHNIQUESIUES
☑ Equivalecne Partioning
□ Boundary Value Analysis
□ Decision Table
□ State Transition Testing
□ Functional Testing

EXPEREANCE BASED TESTINGTING
☑ Error Guessing Technique
□ Exploratory Testing Technique

STRUCTURE BASED TESTINGTING
□ Test Coverage
☑ Statement Coverage & Statement Testing
□ Decision Coverage & Decision Testing
□ Block Coverage

MACHINE LEARNINGNING
☑ Support Vector Machine Algortham
□ Decesion Tree Algortham
□ Fuzzy Logic
□ Neural Network

FAULT DETECTED USING BCO: 41.78%

FAULT DETECTED USING ACO: 39.98%

Fig. 2: 1st priority results of BCO and ACO

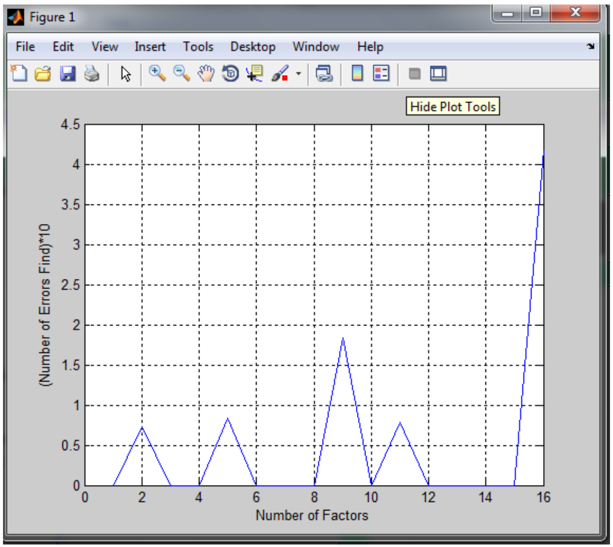


Fig 3: Graph of 1st priority technique using BCO

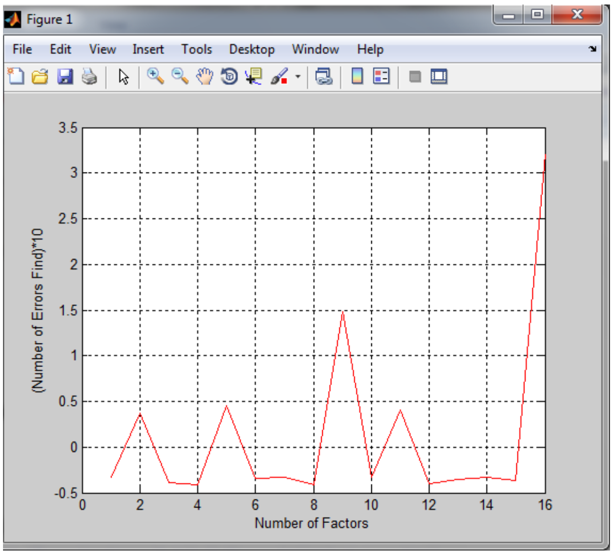


Fig 4: Graph showing technique of 1st priority using ACO

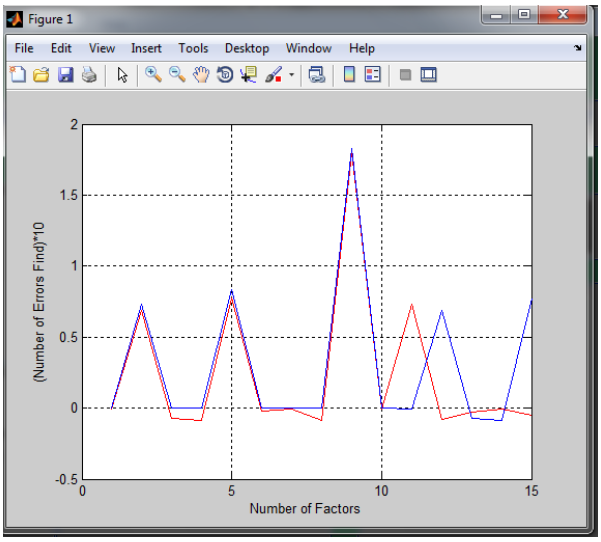


Fig 5: Comparison Graph of 1st priority techniques

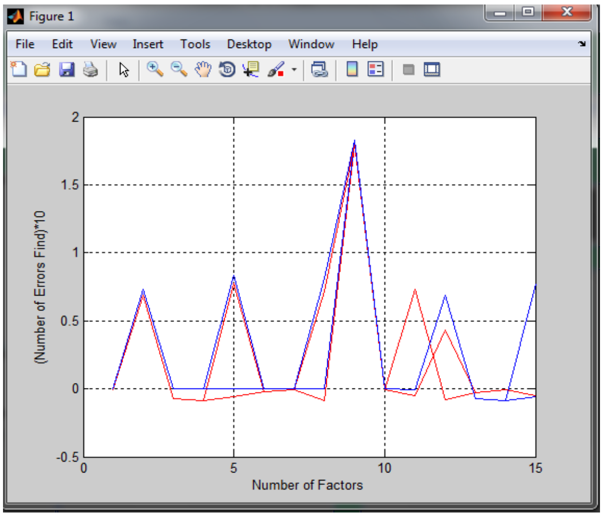


Fig 6: Comparison graph of 16th priority techniques

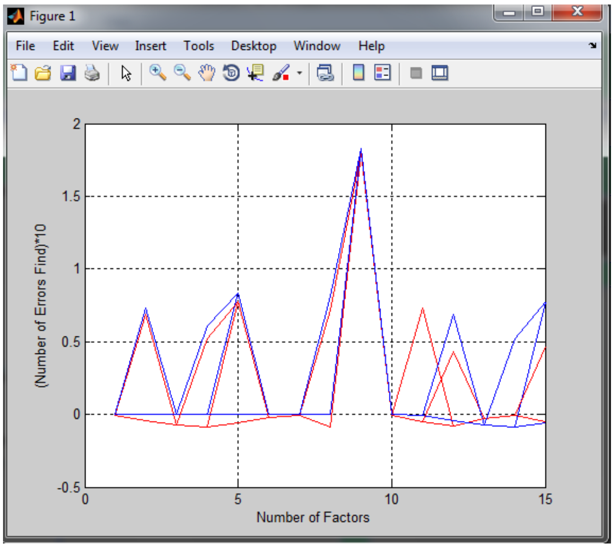


Fig 7: Comparison graphs for 21st priority techniques

5. CONCLUSION

In this paper we collected data sets from various experimental results. After applying BCO algorithm upon all the fault detection techniques we get the optimal solution. At the last we get the best technique which should be use first than others which reduce time consumption. The comparison with ACO shows that BCO algorithm is better than ACO which cover more faults as compared to ACO. In future different fault detection techniques can be used and comparison with other optimization techniques can also be possible.

6. REFERENCES

[1] http://www.buzzle.com/articles/software-testing-techniques.html

[2] Shivkumar Hasmukhrai Trivedi, “Software Testing Technique”, International Journal of Advanced Research in Computer Science and Software Engg 2 (10), October-2012, pp. 433-438

[3] Meenakshi Vanmali, Mark Last, Abraham Kandel, “ Using a Neural network In Software Testing” in 2000

[4] Praveen Ranjan Srivastava**,** Sirish Kumar1**,** A.P. Singh**,** G. Raghurama, “Software testing Effort: An Accessment Through Fuzzy Criteria Approch”, Journal of Uncertain Systems Vol.5, No.3, pp.183-201, 2011

[5] Jovanovic Irena, “Software testing methods and techniques”.

[6] Carina Andersson, Thomas Thelin, Per Runeson, Nina Dzamashvili, “An Experiment Evaluation of Inspection and Testing For detection of Design faults”

[7] Dr. Arvinder Kaur , Shivangi Goyal, “A Bee Colony Optimization Algorithm for Fault Coverage Based Regression Test Suite Prioritization” International Journal of Advanced Science and Technology Vol. 29, April, 2011

[8] Bharti Suri ,Shweta Singhal, “Implementing Ant Colony Optimization for Test Case Selection and Prioritization”, International Journal on Computer Science and Engineering (IJCSE)

[9] Wei Liu_and Sanjay Chawla, “A Robust Decision Tree Algorithm for Imbalanced Data Sets”

[10] Mandeep Kaur Bedi, Sheena Singh, “Comparative study of two natural phenomena based optimization techniques”, International Journal of Scientific & Engineering Research Volume 4, Issue3, March-2013 ISSN 2229-5518

[11] Xia Cai Michael R. Lyu, “The Effect of Code Coverage on Fault Detection under Different Testing Profiles”

[12] Stuart C. Reid, “An Empirical Analysis of Equivalence Partitioning Boundary Value Analysis and Random Testing”

[13] Sujun Hua and Zhirong Sun, “A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure:Support Vector Machine Approach” http://www.idealibrary.com on J. Mol. Biol. (2001) 308, 397±407

[14] http://www.enwikipedia.com