International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34720-34725

Interface for protein structure prediction using self-organizing genetic algorithm

Amouda Venkatesan[1], Dheebika Kuppusamy[2], Jeyakodi Gopal[3], Kamelesh Gasva[4]
[1,2,3,4]Centre for Bioinformatics, School of Life Sciences, Pondicherry University, Pondicherry 605014, India.
amouda@bicpu.edu.in[1], dheebika.kuppusamy@gmail.com[2], rjeyakodi02@gmail.com[3], harisewak@gmail.com[4]

Abstract

Background: *Even though several published studies have suggested that experimental methods play a vital role in solving problems, their extensive use in research still remains expensive and time consuming. Viable alternative, insilico methods are adopted by developing algorithms, databases and tools to overcome the limitations. In the field of bioinformatics, many such tools are used to annotate the exponential growth of biological data. One of the essential tasks, Protein Structure Prediction (PSP), a NP-hard problem became an active field of research in bioinformatics due to its importance in drug design.*

Methods: *Despite of many tools available to solve PSP using Genetic Algorithm (GA), yet another computational tool, Self-Organizing Genetic Algorithm (SOGA) is developed with a special feature of blending self-organizing concept to automate the tuning of parameter of GA without domain experts. It predicts the structures of varied lengths by using a popular tool, tinker. In an attempt, a major drawback faced is the manual input of the sequence, which makes the process time consuming and error prone intake of larger proteins. In order to overcome the drawback the process of input is automated.*

Results and Conclusion: *An interface designed for protein structure prediction tool SOGA with tweaked tinker is developed and tested. It is observed that the overall time and effort consumed to predict the structure is profoundly reduced, which proves the performance efficiency of the tool. The low energy conformation of the predicted model correlates well to the stability of protein structure. The interface is available at www.sogapsp.bicpu.edu.in.*

Keywords- Genetic algorithms, databases, protein structure, crossover, mutation.

1. Introduction

In computer scenario "Tool" is an accessory that helps the user to manipulate the application in an efficient manner. Enormous biological data accessible in various databases necessitate being annotated and analysed using a number of tools to get desired result. The most challenging problem in bioinformatic community is integrating various databases, tools and algorithms yet to be dealt, which can be facilitated by computer scientists and statisticians.

One of the challenging and essential areas in structural biology is protein structure prediction (PSP) because it determines the molecular function of a protein which in turn can be used for drug design. Among the available experimental structure solving methods, X-ray crystallography, Nuclear Magnetic Resonance (NMR) and Electron Microscopy are most accurate and widely used.

Due to technological advances genome sequencing has become an easy task. Unfortunately, the advances have not been paralleled in the field of protein structure determination. As the experimental methods to determine protein structure are time and labor consuming, the gap between structure and sequence has been rising. The probability of a protein domain to have a solved structure has dropped to 0.7% by the end of 2008; this was 1.2% in 2007 and 2% in 2004[1].

To address this challenge, PSP through computational approach has become an active field of research which can be generally classified into three types; homology modeling (also called template modeling), threading (fold recognition) and ab initio modeling (free modeling). Homology modeling requires a known template structure for building the target structure. Instead threading relies on lesser related multiple structures to identify different folds which are then aligned together. Ab initio modeling is a complete physiochemical approach to PSP. It endeavours
to solve the PSP problem without any evolutionary knowledge, building the three dimensional (3D) protein structures from scratch. Of the three, homology modeling has considerable success, followed by threading. But the compulsion of having a known related structure is a liability since only 2% of the known protein sequences have determined structures. Ab initio modeling has got a little success, recently being able to solve structures of less than 120 amino acids but with the aid of huge computational resources and time [2-4].

The main problem of PSP is the large conformational space to be searched for. To overcome this, the use of torsion angles to determine 3D confirmation of proteins has risen. These torsion angles determined from experimental structures are considered to save a lot of time as the conformational space is limited [5-6].

In existence, tinker, a popular tool for ab initio structure prediction intakes torsion angles manually as input. This highly limits the application of tinker to predict the

structures of larger proteins (>50 amino acid) as the process is laborious, error prone, cumbersome and time consuming. The above limitation has been addressed in the current research by tweaking the PSP module ("protein") to read the input through file [7].

In the current work, the developed Self-Organizing Genetic Algorithm (SOGA) has been employed to solve PSP. Upon careful observation of literature studies involving GA, it is evident that success of PSP is largely a function of prudent use of Crossover (CO) and Mutation (M) genetic operators aided by an efficient fitness method. This can only be achieved by trial-and-error and subsequent changes in the operators [8-10]. SOGA offers the advantage of automated self-evaluation and generation of optimal solution. The value of operator's rate is gradually incremented for each successive generation. The fitness function employed is primarily based on free energy value of the resultant structure, calculated using Discovery Studio, a popular tool for energy minimization. The algorithm is set to run through a specified range of genetic operators, and terminate upon the completion of range. The whole procedure has been coded in JAVA and an intuitive graphical interface is provided which makes the process simple, time saving and efficient [11-13]. The performance of the developed interface is seen by solving PSP for large protein.

2. Methodology

It comprises of three modules.
Module I: Generation of solution space for PSP using SOGA.
Module II: Structure Prediction using modified tinker.
Module III: Optimal solution selection using Discovery Studio.
The pictorial representation of work flow is given in Fig. 1.

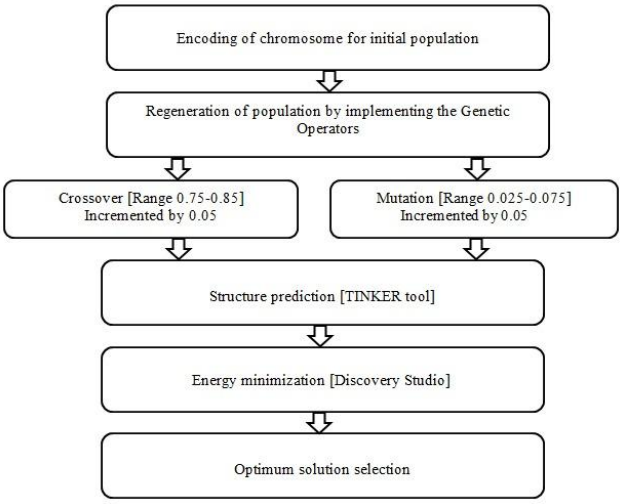

Fig. 1. Work flow of SOGA-PSP.

2.1 Module I: Generation of solution space for PSP using SOGA

2.1.1 Solution Space Representation

The amino acids of protein sequence are represented in the form of torsion angles, cartesian coordinates, etc. to solve PSP by implementing Genetic Algorithm. As an advantage, torsion angle representation allows sufficient degrees of freedom to create variability in the population, hence it is considered for solution space. In this case, combination of torsion angles is considered to encode the chromosome for GA. A set of 3 main chain angles and 5 side-chain angles are chosen from phi - psi angle distribution database and rotamer database respectively [14-15], referred as a gene which constitutes a chromosome. Since some of the amino acids may not have all side chain angles, zero is set for missing side-chain angles in order to preserve the uniformity [16-17]. The symbolic representation of a gene can be referred in the Fig. 2.

Φ	Ψ	ω	S1	S2	S3	S4	S5

Fig. 2. Symbolic representation of gene

Φ - referred as "phi" is a dihedral angle between N and C alpha atoms
Ψ - referred as "psi" is another dihedral angle between C alpha and C atoms
ω - referred as "omega" is a planar angle between C and N atoms
S1 to S5 - constitute the side-chain angles

2.1.2 Solution Space Generation

Based on the availability of torsion and side chain angles of each amino acid, 30 sets of chromosomes are chosen as a population for solution space.

In order to enrich the chromosomes, the population is regenerated using self-organizing genetic operators. The crossover operator interchanges the genes between the two chromosomes, randomly chosen from the population at crossover point whereas the mutation interchanges the genes within a chromosome at mutation point. Interchange of angle within a chromosome leads to conformational change, so the side chain angles at the mutation point have been replaced by alternative angles taken from the database. The ranges [0.75 – 0.85] & [0.025 – 0.075] with 0.5 & 0.025 increment are chosen for crossover and mutation respectively.

2.2 Module II: Structure Prediction using modified Tinker

2.2.1 Integrated Interface for PSP

Tinker, a molecular modeling package is used for protein structure prediction. Among all the modules, "protein" and "xyzpdb" play a vital role in predicting three dimensional (3D) structures. The "protein" module intakes the amino-acids with torsion angles and builds internal and cartesian coordinates with extension ".int (internal coordinates) ", ".seq (sequence)" and ".xyz (cartesian coordinates)". The "xyzpdb" module converts the generated cartesian coordinates from ".xyz" to ".pdb" to visualize the predicted structure.

2.2.2 Need for Modified Tinker

Tinker a widely used tool for PSP accepts the input in the form of torsion angles of each amino acid manually. Unfortunately, only the command line is available for manual input which makes the process tedious, time consuming, and error prone. In case of errors due to manual input, the user is compelled to redo the entire process from scratch. This drawbacks highly restrict the tool to predict the structure of large proteins. A sample of manual input of tinker tool is shown in Fig. 3.

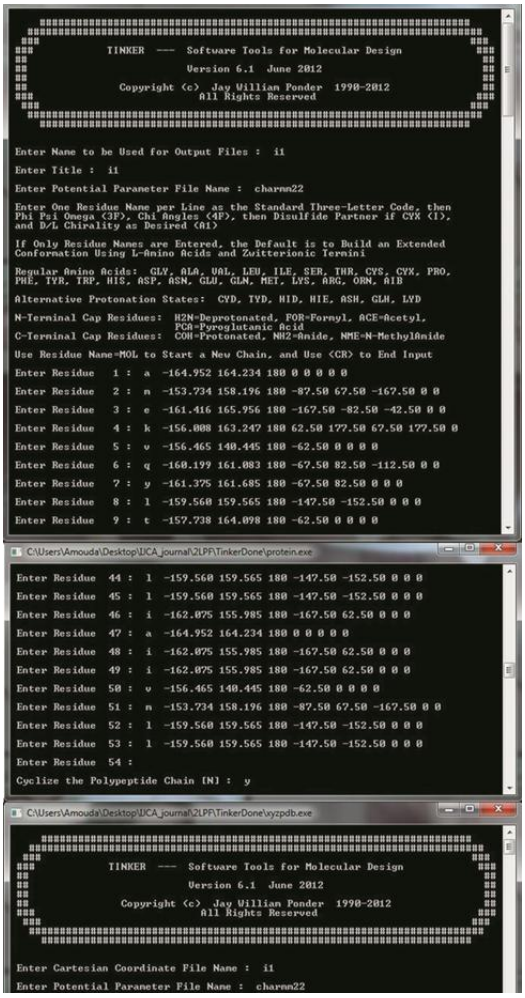

Fig. 3. Tinker execution.

To overcome the limitations, the current work focuses on tweaking the PSP module ("protein") by allowing the file input consisting of torsion & side chain angles of each amino acid [7]. As tinker and SOGA, are two different applications, an integrated interface SOGA-PSP, is also designed and developed in one platform. The newly developed interface reduces the time and effort constraints by automating the SOGA operations and tinker's manual input. The screenshot of the interface is given the Fig. 4.

Fig. 4. Screenshot of SOGA-PSP Interface.

2.2.3 Interface for PSP

The back end of the designed SOGA-PSP web interface is as a stand-alone application in JAVA which provides menu such as "File" and "Procedure" for uploading the sequence stored in a text file to implement SOGA operations. The process provides procedure menu. A sample of an input sequence is shown in the Fig. 5.

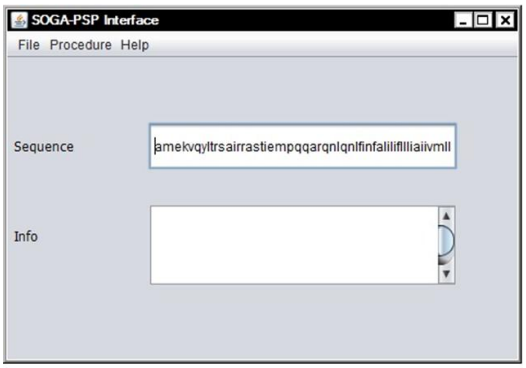

Fig. 5. Interface for SOGA-PSP to enter input sequence.

2.2.4 Initial Population

The chromosomes of initial population of SOGA-PSP interface are generated by "Procedure→ Initiation → Chromosome generation", to generate a text file "chromosome.txt". This output file consists of torsion angles of each amino acid for a given protein sequence. As a test case, a protein "cardiac phospholamban" with 53 amino acids is chosen. Automated tinker accepts the input as a list of amino acids in a single letter representation followed by their torsion angles stored in a text file. The drop down menu of the developed interface has "Procedure→Initiation→Chromosome Separation" options to generate the required format. As a result, files "1.chr" to "30.chr" are generated by making use of the "chromosome.txt" file which contains chromosomes.

2.2.5 Self Organizing Genetic Operators

Unlike other genetic algorithms, in SOGA the ranges of rates of genetic operation are automated using self-organizing concept. Based on self-organizing range of rate of crossover and mutation, SOGA-PSP interface chooses appropriate

options from the menu to produce three generations. A snapshot of a complete solution space generated using SOGA is given in Fig. 6.

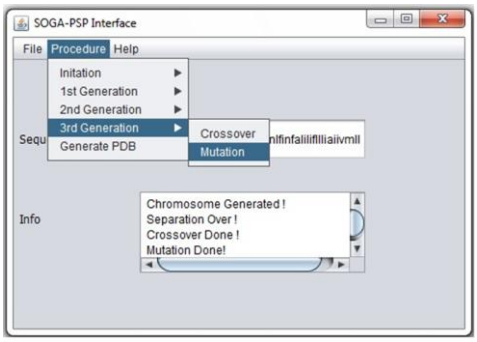

Fig. 6. Interface snapshot for solution space refinement.

2.2.6 Structure Prediction

In order to generate the coordinate files (.xyz, .int, .seq) the developed interface which is a backend of the web interface provides the option "Procedure→generate PDB". Tinker executes to generate the coordinate files by entering path of all "(.chr)" files folders.

Once the generation of coordinate files is complete, terminal ask for the path of 'xyzpdb' module to be entered to generate (.pdb) files. These output files of 'xyzpdb' module is further visualized to view the three dimensional structures of the protiens. The Execution of interface of automated tinker is shown in the Fig. 7.

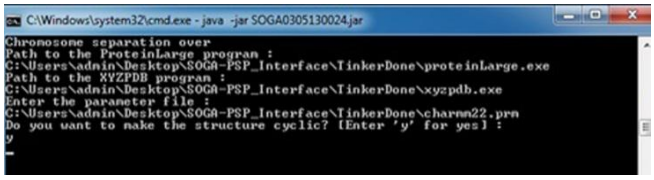

Fig. 7. Tinker execution for .pdb file generation.

2.3 Module III: Optimal Solution Selection using Discovery Studio

As a structure is much stable at lower energy, the structure with minimal energy is chosen as an optimal solution. Energy minimization is done in order to find the most stable conformation of the predicted structure. Discovery studio, a commercial package for Molecular Modeling and Simulations of small and macro molecules is used to minimize the energy of each predicted structure. As a result, the energy values produced are recorded for the selection of optimum solution.

3. Results and Discussion

To validate the efficiency of developed interface, a protein "cardiac phospholamban" (PDB ID: 2LPF) which regulates the activity of the calcium pump of cardiac sarcoplasmic reticulum is considered. The predicted structure of the protein with least energy is shown in Fig. 8. It is observed from Tables 1, 2 and 3 the minimal energy value of the structure is considerably less compare to the original energy (-2464.35) as reported [18]. This minimal energy of predicted structure of protein elucidates a better conformation. A clear cut additional advantage of developed interface is that the overall time and effort consumed is profoundly reduced which is seen in Table 4 and Fig. 9.

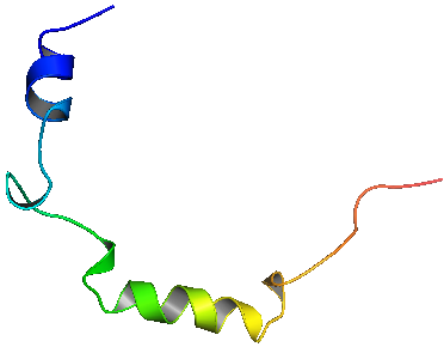

Fig. 8. Predicted structure with least energy.

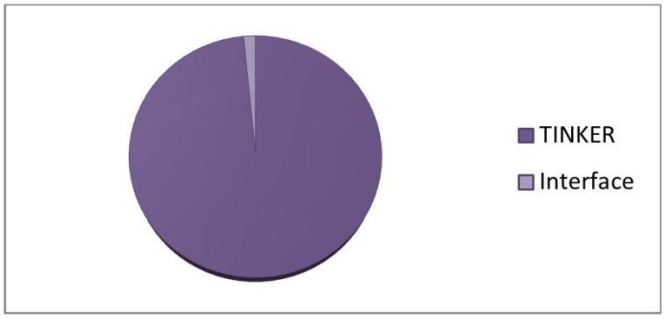

Fig. 9. Graphical representation of time difference between Tinker & SOGA-PSP Interface (in minutes).

Table 1. Energy values in 1st generation.

Generation 1			
Cross over		Mutation	
Structure name	Min. potential energy	Structure name	Min. potential energy
c1	-1775.28	m1	-1787.33
c2	-1755.2	m2	-1735.84
c3	-669.332	m3	-691.596
c4	-874.259	m4	-997.111
c5	-2088.75	m5	-2082.55
c6	-1975.9	m6	-1980.69
c7	-2101.9	m7	-2140.82
c8	-1517.4	m8	-1454.7
c9	-2271.19	m9	-2253.72
c10	**-2485.088**	m10	-2322.31
c11	-2282.99	**m11**	**-2441.16**
c12	-2276.3	m12	-2247.82
c13	-2051.77	m13	-2173.55
c14	-2136.45	m14	-2132.89
c15	-2189.43	m15	-2178.64

c16	-2190.85	m16	-2180.25
c17	256.7315	m17	266.3925
c18	-1956.37	m18	-1952.56
c19	-2091.18	m19	-2070.96
c20	-644.186	m20	-656.166
c21	-2038.34	m21	-2002.42
c22	-2038.34	m22	-2019.55
c23	-2135.9	m23	-2063.28
c24	-2338.6	m24	-2324.69
c25	-2210.71	m25	-2180.47
c26	-2151.19	m26	-2136.27
c27	-714.792	m27	-739.811
c28	-2145.62	m28	-2234.03
c29	-2093.29	m29	-2139.64
c30	-598.42	m30	-419.731

Table 2. Energy values of 2nd generation. Generation 2

Cross over		Mutation	
Structure name	Min. potential energy	Structure name	Min. potential energy
cc1	-1794.72	mn1	-1786.29
cc2	-1724.53	mn2	-1715.6
cc3	-2027.91	mn3	-2034.01
cc4	-1383.17	mn4	-1174.37
cc5	-2101.06	mn5	-2108.49
cc6	-1946.28	mn6	-1951.23
cc7	-2092.89	mn7	-2095.32
cc8	2921.031	mn8	2744.374
cc9	-2256.4	mn9	-2263.45
cc10	**-2525.03**	**mn10**	**-2521.70**
cc11	-2228.94	mn11	-2255.87
cc12	-2233.67	mn12	-2247.47
cc13	-2215.92	mn13	-2178.73
cc14	-2166.3	mn14	-2152.93
cc15	-2197.82	mn15	-2169.11
cc16	-2194.18	mn16	-2169.11
cc17	-704.456	mn17	-774.587
cc18	-1921.05	mn18	-1917.2
cc19	-2123.05	mn19	-2187.46
cc20	-686.361	mn20	-651.413
cc21	-2033.61	mn22	-2031.3
cc22	-2033.61	mn21	-2143.01
cc23	-2107.06	mn23	-2207.84
cc24	-2248.17	mn24	-2248.17
cc25	-2193.74	mn25	-2193.74
cc26	-2204.55	mn26	-2158.68
cc27	-369.296	mn27	-361.128
cc28	-2172.5	mn28	-2179.14
cc29	-2105.53	mn29	-2064.28
cc30	-423.519	mn30	-578.347

Table 3. Energy values in 3rd generation. Generation 3

Cross over		Mutation	
Structure name	Min. potential energy	Structure name	Min. potential energy
ccc1	-1786.29	mmm1	-1785.78
ccc2	-1715.6	mmm2	-1720.87
ccc3	-2034.01	mmm3	-645.08
ccc4	-1174.37	mmm4	-1359.76
ccc5	-2108.49	mmm5	-2088.2
ccc6	-1951.23	mmm6	-1977.26
ccc7	-2095.32	mmm7	-2069.2
ccc8	2744.374	mmm8	-1590.23
ccc9	-2263.45	**mmm9**	**-2396.01**
ccc10	**-2521.70**	mmm10	-2321.42
ccc11	-2255.87	mmm11	-2253.97
ccc12	-2247.47	mmm12	-2253.71
ccc13	-2178.73	mmm13	-2125.49
ccc14	-2152.93	mmm14	-2083.20
ccc15	-2273.64	mmm15	-2181.77
ccc16	-2169.11	mmm16	-2192.39
ccc17	-774.587	mmm17	-931.514
ccc18	-1917.2	mmm18	-1890.47
ccc19	-2187.46	mmm19	-2137.83
ccc20	-651.413	mmm20	-654.106
ccc21	-2143.01	mmm21	-1968.33
ccc22	-2031.3	mmm22	-2049.2
ccc23	-2207.84	mmm23	-2125.43
ccc24	-2248.17	mmm24	-2253.07
ccc25	-2193.74	mmm25	-2198.3
ccc26	-2158.68	mmm26	-2155.23
ccc27	-361.128	mmm27	-710.66
ccc28	-2179.14	mmm28	-2181.63
ccc29	-2064.28	mmm29	-2071.51
ccc30	-578.347	mmm30	-516.73

Table 4. Efficiency of the tool in terms of Time.

Tool Name	Time (in minutes) per generation	Time(in minutes) / Generation (max. 3 generations)	Time (in minutes) to convert ".xyz" to ".pdb" format
Tinker	960	3360	120
SOGA-PSP Interface	2	2	10

4. Conclusion

It is a fact that the protein structure determination through experimental methods is time consuming and effort demanding. To overcome these limitations, computational approaches play a vital role. One such algorithmic approach, SOGA is developed and implemented on PSP using tinker to model the peptides and it has been proven successful in the previous work [18]. The major drawback of tinker is manual input of torsion angles which is time consuming and error prone. This problem has been addressed in the current work by tweaking the "protein module" of tinker which automates the input through file and eliminates the time constraint and error to a greater extent. Also the development of web interface to use the tweaked tinker and SOGA, in turn makes the process much easier and quicker especially, for long proteins. As result, it is shown that the overall time

consumption to predict a structure of long protein is exponentially reduced, thereby reducing the effort. And also, the energy values of predicted protein structure are proven to be minimized to a greater extent than the native one as it counts much for the stability of the protein.

5. Conflict of Interest Statement

None declared.

6. Acknowledgement

The authors acknowledge the University Grants Commission, New Delhi, India for providing financial support (F. No. 42-862/2013 (SR)) for smooth functioning of the project. The authors are grateful to Ms. C. Manimozhi and Mr. P. Elavarasan, Centre for Bioinformatics, Pondicherry University for their constant support all through the work.

References

[1] Athanasia Pavlopoulou, 2011, "State-of-the-art bioinformatics protein structure prediction tools," *International Journal of Molecular Medicine*, Vol. 28, pp 295-310.

[2] Yang Zhang, 2009, "Protein Structure Prediction: Is It Useful?," Current Opinion in Structural Biology, Vol. 19, pp. 145-155.

[3] Yang Zhang, 2008, "Progress and challenges in protein structure prediction," *Current Opinion in Structural Biology*, Vol. 18, pp. 342-348 .

[4] DomenicoCozzetto, 2008, "The Evaluation of Protein Structure Prediction Results," *Molecular Biotechnology*, Vol. 39, pp. 374-384

[5] Roland L Dunbrack Jr, 2006, "Sequence comparison and protein structure prediction," *Current Opinion in Structural Biology*, Vol. 16, pp. 374-384.

[6] Mark T. Oakley, 2008, "Search Strategies in Structural Bioinformatics," *Current Protein and Peptide Science*, Vol. 9, pp. 260-274.

[7] Tinker molecular modeling package, available at: http://dasher.wustl.edu/tinker/

[8] Richard O. Day, Gary B. Lamont and Ruth Pachter, 2003, "Protein Structure Prediction by Applying an Evolutionary Algorithm," in *Proceedings IEEE International Symposium on Parallel and Distributed Processing*, pp. 155.1

[9] T.W. de Lima, P. H. R. Gabriel, A. C. B. Delbem, R. A. Faccioli and I. N. da Silva, 2007, "Evolutionary Algorithm to ab initio Protein Structure Prediction with Hydrophobic Interactions," in Proceedings of *IEEE Congress on Evolutionary Computation*, pp. 612-619.

[10] A. Piccolboni and G. Mauri, 1998, "Application of evolutionary algorithms to protein folding prediction, Artificial Evolution," *Lecture Notes in Computer Science* Vol. 1363, pp. 123-135

[11] Amouda Nizam and Buvaneswari Shanmugam, 2013, "Self-Organizing Genetic Algorithm: A Survey," *International Journal of Computer Applications*, Vol. 65, pp. 25-32.

[12] Thomas Dandekar and P Argos, 1997, "Applying experimental data to protein fold prediction with the genetic algorithm," *Protein Engineering*, vol.10, pp. 877-893.

[13] ViniciusTragante Do Ó, Tinos and Renato. 2010, "A Self-Organizing Genetic Algorithm for Protein Structure Prediction," *Learning & Nonlinear Models,* Vol. 8, pp. 135-147.

[14] Srinivasan R. Phi, psi distribution of amino-acids from 42 proteins [Internet]. Baltimore MD): RoseLab, Johns Hopkins University; [cited 2013 May 1]. Available from: http://roselab.jhu.edu/~raj/Research/Linus/phipsi.html.

[15] Tuffery P, Etchebest C, Hazout S, Lavery R. 1991, "A new approach to the rapid determination of protein side chain conformations," *Journal of biomolecular structure & dynamics*, Vol. 8, pp. 1267-89.

[16] César Manuel Vargas Benítez and Heitor Silverio Lopes, 2010, "Protein structure prediction with the 3D-HP side-chain model using a master-slave parallel genetic algorithm," *J BrazComputSoc*, Vol. 16, pp. 69-78.

[17] Bruno Contreras-Moreira, 2003, "Novel Use of a Genetic Algorithm for Protein Structure Prediction: Searching Template and Sequence Alignment Space," *PROTEINS: Structure, Function, and Genetics*, Vol. 53, pp. 424-429.

[18] Amouda Venkatesan, Jeyakodi Gopal, Manimozhi Candavelou, Sowjanya Gollapalli, and Kayathri Karthikeyan, 2013, "Computational Approach for Protein Structure Prediction," *Healthcare Informatics Research*, Vol. 19, pp. 137-147.

BIOGRAPHY

Amouda Venkatesan is an Assistant Professor in the Centre for Excellence in Bioinformatics, Pondicherry University, Pondicherry. She received the master degree in Software System from BITS Pilani and Doctorate degree in Computer Science and Engineering from Pondicherry University. She has more than 12 years of teaching experience. Her principle area of interests is Genetic algorithm and published more than 25 research publications.

Dheebika Kuppusamy received her B.Sc. and M.Sc. degrees in Bioinformatics from the Pondicherry University, India, in 2010 and 2012. Her research interest covers structural bioinformatics, sequence analysis and data mining.

Jeyakodi Gopal did her M.C.A in Computer Science in 2000, M.S University, Tamil Nadu, India; M. Phil (Computer Science, 2010); Vinayag Mission University, Salem, Tamilnadu, India

Kamelesh Gasva received his B.Sc. and M.Sc. degrees in Bioinformatics from the Pondicherry University, India, in 2011 and 2013. His research interest covers structural bioinformatics and genomics.