3D-Judge - A Metaserver Approach to Protein Structure Prediction

Wojciech Jaskowski *, Jacek Blazewicz * †, Piotr Lukasiak * † , Maciej Milostan *, Natalio Krasnogor ‡

Abstract. Analysis and prediction of three dimensional (3D) protein structure become one of the crucial task in science nowadays. Unfortunately, it is hard to identify one methodology which gives the best prediction of 3D protein structure for different protein sequences. Trying to solve this problem, the concept of metaserver has been introduced. In this paper, we propose a new metaserver method — 3D-Judge, which uses an artificial neural network (ANN) to select the best model from among models produced by individual servers. This decision is made basing on the mutual similarities between models produced by the servers and the knowledge obtained during the training. ANN is trained on historical data, e.g., models from CASP experiment. Here, we compare 3D-Judge with 3D-Jury that is a popular and effective metaserver method. The obtained results indicate that the 3D-Judge is competitive to 3D-Jury and, in some cases, outperformes it.

Keywords: Protein structure prediction, neural networks, meta-server

1 Introduction

In the last decade the number of protein sequences gathered in databases [5] increased tremendously, but the key to understand the function of a protein is not its sequence but its structure. Experimental determination of a three dimensional native folded structure of a particular protein usually can be done via crystallography [13] or NMR techniques [51]. However, such experiments are time and money consuming. This

[1]Work partially supported by KBN grant no. 3T11F00227, EPSRC grant no. GR/T07534/01 and BBSRC grant no. BB/C511764/1.

*Institute of Computing Science, Poznan University of Technology, Poznan

†Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznan, Poland

‡School of Computer Sciences and Information Technology, University of Nottingham, Nottingham, NG81BB, United Kingdom

simple fact makes the secrets of nature harder to reveal, but at least it enforced the scientific world to start thinking about computer aided modeling techniques.

Large number of algorithms for modeling have been proposed and implemented [6]. In the universe of methods one can find approaches based on simplified, reduced (e.g. lattice) models with force fields and Monte Carlo simulation or some metaheuristics as a simulation engine [7, 25, 45, 33, 34, 32, 17, 26, 46, 2], descriptor and fragment based methods[11], threading [22], several kinds of homology modeling strategies[50]. Most of the methods are freely available for academic community either as web-services(servers) or standalone programs. Independent assessment of a current state of the art in protein structure prediction is done during the biennial community wide experiment for critical assessment of the techniques for protein structure prediction (CASP)[12]. Six editions of CASP showed that there is no method, that is always the leader and overall prediction quality depends on many factors which are unknown when the prediction is made. For example, in many cases one does not know if the given protein has distant or close homologs, or if it has a new fold.

The fact that no one method dominates in all prediction instances brings into sharp focus the following important research issue: How do we decide which is the best method for predicting the structure for a specific sequence? Nowadays, this question has not many answered, but at least great effort has been done to get better idea of the possible solution. These efforts brought to life a concept of consensus based approaches e.g. [28] and meta-strategies, so called meta-servers [9]. The concept is trivial and brilliant if one cannot choose easily the best method for a given sequence a priori (before applying all the methods) then maybe it is easier to do it a posteriori by a comparison of models predicted, i.e., by applying all the methods and choosing the most realistic one. One of the simplest form of the choice strategy can be found in the so called 3D Jury method proposed by Ginalski et al. [14]. The 3D Jury computes structure similarities between models using MaxSub measure and generally speaking, chooses as the best prediction the one which has the most popular fold in the set of given models. The algorithm proposed in the following sections tries to find more sophisticated relationships between models and underlying methods. Historical knowledge about similarity patterns among models and real structures, is incorporated in the prediction scheme by means of a powerful mechanism of neural networks.

In the following sections one can find the concept of a novel approach proposed to solve this problem, the detailed description of the computational experiment, a discussion of the results and conclusions.

2 Approach

Currently, there are hundreds of different fully automated approaches to protein 3D structure prediction. Unfortunately, if there is no significant similarity between the target sequence and sequences of proteins with known structure or when the sequence alignments between the target and the template contain large errors then fully automated methods can produce reliable sequence-structure assignments for only a frac-

tion of target sequences [10]. Another problem is to identify the best model among the top candidates, although, as CASP5 [12] has shown, quite often the correct fold is within the best ten models. The confidence score that should help to choose the best model is often untrustworthy. As a result, good models are buried among false positives.

Meta-servers were an attempt to solve the above mentioned problems. A meta-server uses results from a set of servers and produces its own result based on 3D structures generated by those servers. This approach has many advantages. Firstly, as many experiments have shown (e.g., CASP5, CASP6), meta-servers produce generally better results than individual servers. Secondly, the results are more stable in case of meta-servers. Currently, one of the best meta-server is 3D-Jury [14]. It was widely used as a helper method in CASP6 and it's effectiveness was proved in LiveBench [10, 42] experiments.

Artificial neural networks (ANN) have been applied to many aspects of predicting protein structures. At the beginning such methods were designed as a quick and dirty demonstration that artificial intelligence-based approaches could solve real-life problems. However, more thorough investigations enabled researchers to include information used by experts into neural network-based tools. Now, some tools are on average as accurate as the best experts; and experts using such tools often arrive at even more accurate predictions. Thus, several neural network-based methods have eventually contributed significantly to advancing the field of bioinformatics. Neural networks were first applied to the prediction of secondary structures [8, 36]. This stimulated many subsequent studies of neural networks [20, 23] in the secondary structure prediction. However, a real breakthrough did not come until the work presented in [38, 39, 40]. Since last decade artificial neural networks have been used very successfully in biological sequence analysis for purposes as diverse as prediction of functional similarities between proteins (layered feed-forward networks usually trained by simple back-propagation [41, 1], and Kohonen maps [1, 24]), recognition of functional motifs[18, 19, 27, 43, 16, 30], fold recognition [22], etc.

In this paper, we propose a new metaserver method — 3D-Judge, which uses artificial neural network (ANN) to select the best model from among models produced by individual servers. This decision is made basing on the mutual similarities between models produced by the servers and the knowledge obtained during the training. ANN is trained on historical data (models from CASP experiment).

Note that 3D-Judge is similar to 3D-Jury, because both methods use similarities between models produced by individual servers. On the other hand, 3D-Judge goes a step further by analysis of provided historical data in order to learn the possible relationships of model similarities.

3 Methods

3.1 The model

3D-Judge is a selector meta-predictor. It selects one model from those produced by n predictor servers. The decision which model should be chosen is made by artificial neural network trained on the historical data.

Let us introduce the following notation. Let P be the set of predictor servers and $M_p(s)$ mean the best model generated by server $p \in P$ for a certain sequence s. Let $sim(M_{p_1}(s), M_{p_2}(s))$ be the *similarity* of the models M_{p_1} and M_{p_2}. In particular, when the real 3D structure $M_*(s)$ is known for a certain sequence s, $sim(M_p(s), M_*(s))$ is the evaluation score for model produced by predictor p. In this context, historical data H is a set of sequences $s \in H$ for which the 3D-structures $M_*(s)$ are known, accompanied by models produced by all individual servers from set P for those sequences.

3.2 The similarity measure

During last few years many methods for the evaluation of a similarity of protein 3D-structures were introduced. The most known methods include MaxSub [44] and LGA [54]. Although the differences between algorithms used are serious, the common features of them is that all of them take as arguments two protein models and produce a value, so called similarity score, that can be interpreted as a percentage of similarity between those models. 100% means identity and 0% means no similarity at all (in practice, not enough similarity). It is important to note that the similarity methods described above can be used both to determine the similarity between two models and to evaluate a model against a real protein structure. In our algorithm similarity methods were used in both mentioned contexts.

As a protein structure comparison method we used GDT (Global Distance Test) that is a part of LGA package and was used to evaluate quality of predicted protein structure models in CASP6 [12], however, any similarity comparison method could be used instead, or indeed a similarity metaserver such as ProCKSi[4]. The distance cutoff was set to 4.0 Angstroms.

In our method, we assume that the similarity measure is symmetrical, i.e., $sim(M_1, M_2) = sim(M_2, M_1)$. For structure comparison methods (such as GDT) that does not satisfy this assumption, we use the average of $sim(M_1, M_2)$ and $sim(M_2, M_1)$ as the similarity between M_1 and M_2.

3.3 The artificial neural network

The ANN used by 3D-Judge has three layers: input, hidden and output. The input layer consists of $(|P|^2 - |P|)/2$ neurons. Each of them corresponds to a pair of predictors p_1,p_2 and processes similarity score $sim(M_{p_1}(s), M_{p_2}(s))$. For the output, ANN

uses one-of-N encoding, thus it has $|P|$ output neurons. Each neuron corresponds to one predictor server. ANN chooses the predictor model for which the output neuron has the highest value. The number of neurons in the hidden layer of ANN was set to $|P|^2$ in result of some preliminary experiments. Hidden layer consists of sigmoid symmetric neurons, whereas output layer neurons are sigmoid asymmetric neurons (see e.g. [31] for details on neurons types).

3.4 Training ANN

Artificial neural network (ANN) has to be trained on historical data H in order to properly choose the best model among models produced by the servers.

As it was stated before, for each sequence $s \in H$ the models produced by the servers as well as the real 3D-structures are known. ANN learns to produce evaluation scores as its output given the similarity scores as its inputs. More precisely, given $sim(M_{p_1}(s), M_{p_2}(s))$ for each pair p_1,p_2 to the input layer, ANN is supposed to return $sim(M_p(s), M_*(s))$ for each $p \in P$ by its output layer of neurons. In other words, ANN learns how to *evaluate* a set of models basing on the similarity values between them.

3D-Judge uses standard back-propagation algorithm as a training algorithm for ANN. While offering pretty good quality, back-propagation is one of the fastest ANN learning algorithms. Other parameters of the learning process such as learning rate, desired error, etc. were set to the values recommended by FANN library (see [31] for details).

3.5 Implementation

3D-Judge was written in C++ with use of Fast Artificial Neural Network library (FANN) [31]. Basing on the description given in [14], we also implemented 3D-Jury meta-server in order to compare it with 3D-Judge.

4 Experiments and Results

For testing of 3D-Judge, we used publicly available data from CASP6 [12]. In CASP6 the servers were examined on 69 targets, thus we disposed 69 data elements (sequences, models produced by servers and real 3D structures of the proteins). As the data set was small, we used leave-one-out as a testing method.

We chose 3D-Jury as a method to compare with our approach, because of several reasons. First, likewise to 3D-Judge, it uses the similarity between models in order to select the best model from among models produced by individual servers. Second, 3D-Jury is widely used as a helper method, because it has proven competitive results. Finally, we wanted to demonstrate that ignoring the historical information, as 3D-Jury does, can lead to poor results in some cases.

Table 1: The average evaluation scores for 3D-Judge, 3D-Jury and all individual servers from among which 3D-Judge and 3D-Jury selected the final model.

Server	Score
MAX	0.536
3D-Jury	0.505
3D-Judge	0.484
Eidogen-EXPM[35]	0.464
RAPTOR[52]	0.461
mGenTHREADER[29]	0.453
nFOLD[29]	0.429
ACE[53]	0.427
Zhousp3[55]	0.420
ZHOUSPARKS2[56]	0.417
BAKER-ROBETTA[3]	0.415
FORTE2[49]	0.404
fams[21]	0.384

The results of the first experiment are shown in Table 1. Since not all servers in CASP6 experiment produced 3D-structures for given sequences, we drawn at random 10 servers servers out of those that provided structures for at least 60 of sequences. We compared them with two metaservers (3D-Judge and 3D-Jury). In Table 1, MAX is the potentially maximum possible result that could be obtained by selecting always the best model from models produced by those 10 servers. No selector metaserver (including 3D-Judge and 3D-Jury) can outperform MAX.

The table shows that the result of 3D-Judge is significantly better than the result of any of the individual servers it relayed on. The distance to optimal MAX is 0.05. In this case, 3D-Jury performed a bit better than 3D-Judge, however, this is not a rule. In Table 2 we presented results obtained from a different set of servers. This time, 3D-Jury was outperformed by 3D-Judge and even by one of the individual servers — Eidogen-EXPM. Notice that in this experiment, the range of evaluations of individual servers was higher than in the first one. Also, the difference between the best of the servers (Eidogen-EXPM) and the second one (mGenTHREADER) is high. This is the probable reason for poor result of 3D-Jury. We believe that the neural network of 3D-Judge is more resistant for such cases and this is why 3D-Judge wins for this set of servers.

Our results suggest that 3D-Judge behaves particularly well in comparison of 3D-Jury when the set of servers contains servers from the same server family or different versions of the same server. Such situation occur often in practice when having models produced by different versions of the same server, we would like to choose the best one. The problem is that, usually, we can not unambiguously state which one of those servers is the best for all the sequences. The different versions of the same algorithm can perform well for certain types of sequences, but it is hard to determine the exact

Table 2: Average evaluation scores for different set of servers.

Server	Score
MAX	0.580
3D-Judge	0.535
Eidogen-EXPM	0.535
3D-Jury	0.508
mGenTHREADER	0.453
ACE	0.427
FORTE2	0.404
FORTE1[47]	0.397
LOOPP	0.376
PROSPECT[15]	0.369
FORTE1T[48]	0.364

Table 3: Results for a small set of servers

Server	Score
MAX	0.476
3D-Judge	0.463
RAPTOR	0.461
FORTE2	0.404
3D-Jury	0.401
FORTE1	0.397
FORTE1T	0.364

rules.

The Table 3 shows the results for a small set of 4 servers. Three of them come from the same family of FORTE servers. Clearly, 3D-Jury was misled by the similarities between the models produced by FORTE servers. The result of 3D-Judge is, on the other hand, much better than the result of 3D-Jury. Notice that, in this case, 3D-Judge differs only by 0.13 from MAX.

5 Conclusion

Experiments have shown that the 3D-Judge meta-server proposed in this paper is competitive to the other successful methods — 3D-Jury. Although the question whether 3D-Judge is better than 3D-Jury is still open, the usage of additional (historical) data that 3D-Jury does not take into consideration, allows for the statement that 3D-Judge has a greater potential. We have shown that for some set of servers, 3D-

Jury can perform very poor whereas 3D-Judge outperforms all the individual servers. The practical problem with our method is that good quality historical data are essential and the method cannot operate without them. However, public databases (such as PDB) are still growing and this practical problem could be less important in the future. Another aspect of this work that we will investigate in more detail in the future is the use of other than GDT structural comparison methods (e.g., MaxSub or ProCKSi) both at the time of training the neural network and of assessing a prediction.

References

[1] M. Arbib, *The handbook of brain theory and neural networks.* Bradford Books/The MIT Press, Cambridge, MA. 1995.

[2] C. J. Bacardit, M. Stout, J.D. Hirst, J. Blazewicz, and N. Krasnogor. "Coordination number prediction using learning classifier systems: performance and interpretability," Proceedings of the 2006 Genetic and Evolutionary Computation Conference (GECCO), pp. 247 - 254. ACM Press New York, NY, USA, 2006.

[3] D. Chivian, D.E. Kim, L. Malmstrom, J. Schonbrun, C.A. Rohl, D. Baker, "The Robetta and Robetta_04 protocols", *CASP6 Abstracts,* pp. 12-13, 2004.

[4] D. Barthel, J.D. Hirst, J. Blazewicz and N. Krasnogor, "ProCKSi: A Meta-Server for Protein Comparison using Kolmogorov and other Similarity Measures," European Conference on Computational Biology, 2007 (submitted).

[5] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, "The Protein Data Bank," *Nucleic Acids Research,* vol. 28, pp. 235-242, 2000.

[6] J. Blazewicz, P. Lukasiak Piotr, M. Milostan "Some operations research methods for analyzing protein sequences and structures," *4OR: A Quarterly Journal of Operations Research,* vol. 4, no. 2, pp. 91-123, 2006.

[7] J. Blazewicz, P. Lukasiak, M. Milostan "Application of tabu search strategy for finding low energy structure of protein," *Artificial Intelligence in Medicine,* vol. 35, pp. 135-145, 2005.

[8] H. Bohr, J. Bohr, S. Brunak, R.M. Cotterill, B. Lautrup, L. Norskov, O.H. Olsen, S.B. Petersen, "Protein secondary structure and homology by neural networks. The alpha-helices in rhodopsin," *FEBS Lett.,* vol. 241, pp. 223-228, 1988.

[9] J.M. Bujnicki, A. Elofsson, D. Fischer, L. Rychlewski, "Structure prediction meta server," *Bioinformatics,* vol. 17, pp. 750-751, 2001.

[10] J.M. Bujnick, A. Elofsson, D. Fischer, L. Rychlewski, "LiveBench-2: Large-scale automated evaluation of protein structure prediction servers," *Proteins*, vol. 45, 184-191, 2001.

[11] C. Bystroff, D. Baker, "Prediction of local structure in proteins using a library of sequence-structure motifs," *J Mol Biol,* vol. 281, pp. 565-77, 1998.

[12] http://predictioncenter.org

[13] J. Drenth, *Principles of protein X-ray crystallography.* Springer-Verlag Inc. NY: 1999.

[14] K. Ginalski, A. Elofsson, D. Fischer, L. Rychlewski, "3D-Jury: a simple approach to improve protein structure predictions," *Bioinformatics.* vol. 19(8), pp. 1015-18, 2003.

[15] J. Guo, W.J. Chung, K. Ellrott, R. Zhou, D. Gelonia, B.D. Silverman, A.K. Royyuru, A. Curioni, A. Logean, Y. Xu, "Fold recognition using PROSPECT", *CASP6 Abstracts,* pp. 149-150, 2004.

[16] J.E. Hansen, O. Lund, N. Tolstrup, A.A. Gooley, K.L. Williams, S. Brunak, "NetOglyc: Prediction of mucin type O-glycosylation sites based on sequence context and surface accessibility," *Glyconjug. J.,* vol. 15, pp. 115-130, 1998.

[17] M. Hao, H.A. Scheraga, "On foldable protein-like models; a statistical-mechanical study with Monte Carlo simulations," *Physica A,* vol. 244, issue: 1-4, pp. 124-146, October 1, 1997.

[18] J.D. Hirst, M.J.E. Sternberg, "Prediction of ATP-binding motifs a comparison of a perceptron-type neural network and a consensus sequence method," *Protein Engineering,* vol. 4, pp. 615-623, 1991.

[19] J.D. Hirst, M.J.E. Sternberg, "Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks," *Biochem.,* vol. 31, pp. 615-623, 1992.

[20] H. Holley, M. Karplus, "Protein Secondary Structure Prediction with a Neural Network," *Proc. Natl Acad. Sci. USA,* vol. 86, pp. 152-156, 1989.

[21] M. Iwadate, K. Kanou, G. Terashi, D. Takaya, M. Takeda-Shitaka, H. Umeyama, "Full automatic homology modeling server including the transformation of amino acid residues: SKE(Sophia Kai Ergon) FAMS", *CASP6 Abstracts,* pp. 55-56, 2004.

[22] DT. Jones, "GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences," *Journal of Molecular Biology,* vol. 287, pp. 797-815, 1999.

[23] D. Kneller, F. Cohen, R. Langridge, "Improvements in protein secondary structure prediction by an enhanced neural network," *Journal of Molecular Biology,* vol. 214, pp. 171-182, 1990.

[24] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.,* vol. 43, pp. 59-69, 1982.

[25] A. Kolinski and J. Skolnick, "Reduced Models of proteins and their applications," *Polymer,* vol. 45, pp. 511-524, 2004.

[26] N. Krasnogor, B. Blackburnem, J.D. Hirst, and E.K. Burke. "Multimeme algorithms for protein structure prediction," In H.-G. Beyer J.-L. Fernández-Villacañas H.-P. Schwefel J.J. Merelo Guervós, Adamidis, editor, *7th International Conference Parallel Problem Solving from Nature, Springer Lecture Notes in Computer Science,* vol. 2439, pp. 769-778, Granada, Spain, September 2002

[27] I. Ladunga, F. Czako, I., Csabai, T. Geszti, "Improving signal peptide prediction accuracy by simulated neural network," *CABIOS,* vol. 7, pp. 485-487, 1991.

[28] J. Lundstrom, L. Rychlewski, J. Bujnicki, A. Elofsson, "Pcons: A neural-network based consensus predictor that improves fold recognition," *Protein Sci.,* vol. 10, pp. 2354 - 2362, Nov 2001.

[29] L.J. McGuffin, J.S. Sodhi, K. Bryson, D.T. Jones "Fully automated fold recognition using nFOLD and mGenTHREADER", *CASP6 Abstracts,* pp. 99-100, 2004.

[30] H. Nielsen, J. Engelbrecht, S. Brunak, G. von Heijne, "Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites," *Protein Engineering,* vol. 10, pp. 1-6, 1997.

[31] S. Nissen, "Implementation of a fast artificial neural network library (FANN)," Departament of Computer Science University of Copenhagen, The University Report, October 31, 2003.

[32] S. Oldziej, J. Lagiewka, A. Liwo, C. Czaplewski, M. Chinchio, M. Nanias, H. A. Scheraga, "Optimization of the UNRES Force Field by Hierarchical Design of the Potential-Energy Landscape. 3. Use of Many Proteins in Optimization," *J. Phys. Chem. B.,* vol. 108(43), pp. 16950-16959, 2004.

[33] A.R. Ortiz, A. Kolinski, J. Skolnick, "Fold Assembly of Small Proteins Using Monte Carlo Simulations Driven by Restraints Derived from Multiple Sequence Alignments," *J Mol Biol,* vol. 277, pp. 419-448, 1998.

[34] A.R. Ortiz, A. Kolinski, J. Skolnick, "Nativelike topology assembly of small proteins using predicted restraints in Monte Carlo folding simulations," *Proc Natl Acad Sci USA,* vol. 95, pp. 1020-1025, 1998.

[35] A. Poleksic, J.F. Danzer, D.A. Debe, "Automated structure modeling with Eidogen's suite of algorithms", *http://www.foracasp.org*, 2004.

[36] N. Qian, T. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," *Journal of Molecular Biology,* vol. 202, pp. 865-884, 1988.

[37] J. Qiu, J. Pillardy, T. Galor, C.B. Lowe, L. Meyerguz, R. Elber, "Fold recognition by machine learning approach", *CASP6 Abstracts,* pp. 91-92, 2004.

[38] B. Rost, C. Sander, "Improved Prediction of Protein Secondary Structure by Use of Sequence Profiles and Neural Networks," *Proc. Natl Acad. Sci. USA,* vol. 90, pp. 7558-7562, 1993.

[39] B. Rost, C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *Journal of Molecular Biology,* vol. 232, pp. 584-599, 1993.

[40] B. Rost, C. Sander, R. Schneider, "PHD - an automatic server for protein secondary structure prediction," *CABIOS,* vol. 10, pp. 53-60, 1994.

[41] D.E. Rumelhart, J.L. McClelland, *Parallel distributed processing. Explorations in the microstructure of cognition.* MIT Press, Cambridge, M.A., U.S.A, 1986

[42] L. Rychlewski, D. Fischer "LiveBench-8: the large-scale, continuous assessment of automated protein structure prediction," *Protein Sci,* vol. 14, pp. 240-245, 2005.

[43] G. Schneider, P. Wrede, "Development of artificial neural filters for pattern recognition in protein sequences," *J. Mol. Evol.,* vol. 36, pp. 586-595, 1993.

[44] N. Siew, A. Elofsson, L. Rychlewski, D. Fischer, "An Automated Measure for the Assessment of Protein Structure Prediction," *Bioinformatics,* vol. 16, pp. 776-785, 2000.

[45] J. Skolnick, A. Kolinski, C. Brooks, A. Godzik, A. Rey, "A method for predicting protein structure from sequence," *Curr Biol,* vol. 3, pp. 414-423, 1993.

[46] M. Stout, J. Bacardit, J.D. Hirst, N. Krasnogor, J. Blazewicz, "From hp lattice models to real proteins: coordination number prediction using learning classifier systems," *4th European Workshop on Evolutionary Computation and Machine Learning in Bioinformatics, Springer Lecture Notes in Computer Science,* vol. 3907, pp. 208-220, Budapest, Hungary, April 2006.

[47] K. Tomii, "FORTE1: a simple profile-profile comparison method applied to fold recognition", *CASP6 Abstracts,* p. 58, 2004.

[48] K. Tomii, "FORTE1T: profile-profile comparison with improved profiles for fold recognition", *CASP6 Abstracts,* p. 59, 2004.

[49] K. Tomii, "FORTE2: automated fold recognition server with enhanced profile library", *CASP6 Abstracts,* pp. 59-60, 2004.

[50] C. Venclovas, M. Margeleviius, "Comparative modeling in CASP6 using consensus approach to template selection, sequence-structure alignment, and structure assessment," *Proteins: Structure, Function, and Bioinformatics,* vol. 61, issue S7, pp. 99-105, 2005.

[51] K. Wuthrich, *NMR of proteins and nucleic acids.* John Wiley & Sons, 1986.

[52] J. Xu and M. Li "Regression-based approaches to fold recognition", *CASP6 Abstracts,* pp. 123-124, 2004.

[53] L. Yu, J. Xu and M. Li, "ACE: Consensus Fold Recognition by Predicted Model Quality", *http://www.foracasp.org*, 2004.

[54] A. Zemla, "LGA - a Method for Finding 3D Similarities in Protein Structures," *Nucleic Acids Research,* vol. 31, no. 13, pp. 3370-3374, 2003.

[55] H. Zhou and Y. Zhou, "Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments in CASP6," *CASP6 Abstracts,* pp. 154-155, 2004.

[56] H. Zhou and Y. Zhou, "Application of SPARKS 2.0 fold recognition server in CASP6", *CASP6 Abstracts,* pp. 155-156, 2004.