



Published in final edited form as:

Proc Congr Evol Comput. 2014 July ; 2014: 1038–1045. doi:10.1109/CEC.2014.6900443.

Multi-Dimensional Scaling and MODELLER-Based Evolutionary Algorithms for Protein Model Refinement

Yan Chen, Yi Shang, and Dong Xu

Yan Chen, Yi Shang, and Dong Xu are with the Department of Computer Science, University of Missouri, Columbia, MO 65211 USA. Dong Xu is also with the Christopher S. Bond Life Science Center, University of Missouri. (yc235@mail.missouri.edu, shangy@missouri.edu, and xudong@missouri.edu)

Abstract

Protein structure prediction, i.e., computationally predicting the three-dimensional structure of a protein from its primary sequence, is one of the most important and challenging problems in bioinformatics. Model refinement is a key step in the prediction process, where improved structures are constructed based on a pool of initially generated models. Since the refinement category was added to the biennial Critical Assessment of Structure Prediction (CASP) in 2008, CASP results show that it is a challenge for existing model refinement methods to improve model quality consistently.

This paper presents three evolutionary algorithms for protein model refinement, in which multidimensional scaling(MDS), the MODELLER software, and a hybrid of both are used as crossover operators, respectively. The MDS-based method takes a purely geometrical approach and generates a child model by combining the contact maps of multiple parents. The MODELLER-based method takes a statistical and energy minimization approach, and uses the remodeling module in MODELLER program to generate new models from multiple parents. The hybrid method first generates models using the MDS-based method and then run them through the MODELLER-based method, aiming at combining the strength of both. Promising results have been obtained in experiments using CASP datasets. The MDS-based method improved the best of a pool of predicted models in terms of the global distance test score (GDT-TS) in 9 out of 16test targets.

Keywords

Multidimensional scaling; MODELLER; evolutionary algorithm; protein model refinement

I. INTRODUCTION

Proteins are essential biochemical compounds that contribute to many processes in life. Functional properties of cells depend on correctly folded protein structures[1]. Misfolded proteins may lead to diseases, such as Alzheimer's, Parkinson's, Type II Diabetes, and even cancers[2] [3]. The knowledge of protein tertiary structure can help in basic research on

protein functions, as well as in drug development. Both experimental methods and computational methods can be used in protein structure acquirement. Among experimental methods, X-ray crystallography and nuclear magnetic resonance (NMR) are the most widely used. However, it is slow, costly, and difficult to find protein tertiary structures through experimental technologies [4]. Computational techniques are aimed at predicting protein structures from primary sequences information and have the potential of being fast, cheap, and widely available. Therefore, obtaining an accurate prediction of the three-dimensional structure of a protein by automatic prediction is one of the most important problems in bioinformatics and has been actively studied for many years.

The Critical Assessment of Structure Prediction (CASP) is a biennial world-wide event in the structure prediction community to assess the current protein modeling techniques and identify their quality. Different prediction software programs from various research groups predict the structure of an unknown protein or refine an existing model of a protein. Generating high-quality protein models and refining existing models are two major steps in the computational process of protein structure prediction.

In model refinement, improved structures are constructed based on a pool of initially generated models. A model refinement category was first added to the CASP8 competition in 2008 to evaluate the state-of-the-art of this area. Participants were given a pool of protein models submitted by the prediction servers, with the best model identified. CASP results show that it is a challenge for existing model refinement methods to improve model quality consistently [5–10]. In CASP10 in 2012, i3 Drefine is the only fully automated server that can improve both local and global structures of prediction models, even though its improvement is small. This method iteratively minimizes an energy function consisting of physics and knowledge-based force fields, and uses a hydrogen bonding (HB) network optimization technique [5]. Galaxy Refine has the best performance in CASP10 in terms of improving local structure quality. It rebuilt side chains, repacked them, and relaxed the models by molecular dynamics simulation [6]. KoBaMIN is another method based on minimization of a knowledge-based potential of mean force [7]. I-TASSER is an automated pipeline for predicting protein 3D structure by multiple threading alignments and iterative structure assembly simulations, and it has an internal refinement module as well [8].

Although existing methods sometimes perform well on model refinement, especially in template-based modeling, further improvement is needed for practical use. During CASP8 and CASP9 [11] [12], only a few groups were able to improve the protein model quality consistently. In CASP10, only two groups improve the protein model quality consistently. The maximum improvement in high accuracy of GDT-TS (GDT-HA) is only about 0.1 [13].

In this paper, three new evolutionary algorithms for protein model refinement, in which multidimensional scaling (MDS), the MODELLER software, and a hybrid of both are used as crossover operators, respectively. The MDS-based method takes a purely geometrical approach and generates a child model by combining the contact maps of multiple parents. The MODELLER-based method takes a statistical and energy minimization approach and uses the remodeling module in MODELLER program to generate new models from multiple parents. The hybrid method first generates models using the MDS-based method and then

run them through the MODELLER-based method, aiming at combining the strengths of both. Promising results have been obtained in experiments using CASP datasets.

This paper is organized as follows. Section II introduces the basics of major techniques used in the proposed methods and some related work. Section III presents the three new evolutionary algorithms to refine protein models in details. Section IV presents experimental results on CASP datasets. Finally, Section V concludes the paper.

II. BASICS OF KEY TECHNIQUES AND RELATED WORK

A. Protein Model Quality Evaluation

Assessing the quality of a computationally generated protein model is essential in protein structure prediction and refinement. A basic hypothesis of protein models is that the native structure has the minimum free energy in general [1]. Most model quality evaluation methods use energy or scoring functions, either physics-based or knowledge/statistics-based [14]. Physics-based functions are designed based on physics laws, such as the rmdynamic equilibrium, to evaluate the models' quality, while knowledge-based functions are designed based on information and properties of protein structures derived from known structures. Another major approach is consensus based: given a pool of predicted models, the quality of a model is the average similarity between it and other models in the pool. In CASP competitions, consensus methods perform much better than scoring functions [15]. However, a major problem of consensus methods is that they require a pool of diverse models of generally high quality to perform well, which is not practical in many real applications, whereas scoring functions can evaluate a single model.

1) Consensus methods based on structure similarity—A key element of consensus methods is the similarity measurement between two 3-D structures or models. Commonly used pairwise similarity metrics include the Root-Mean-Squared Deviation (RMSD), the Template Modeling Score (TM-score), and the Total Score of Global Distance Test (GDT-TS) [2,15,16].

GDT-TS [16] is a global quality measure of the corresponding positioning of amino acid sequences between two protein models. It is one of the major quality assessment metrics in CASP competitions. The GDT-TS score is calculated by averaging the percentage of corresponding residues (represented by the C- α atoms) between two models within a certain cutoff distance after the two models are optimally superimposed over each other. The GDT-TS value is calculated as follows:

$$GDT-TS(S_i, S_j) = (P_1 + P_2 + P_3 + P_4) / 4 \quad (1)$$

Where S_i and S_j are two protein 3D structures and P_d is the percentage that the C- α atoms in S_i is within a defined cut off distance d , $d \in \{1, 2, 4, 8 \text{ \AA}\}$, from the corresponding C- α atoms in S_j [15]. GDT-TS values range from 0 to 1 with higher value indicating two structures are more similar.

In consensus methods, GDT-TS is commonly used to measure the similarity of a pair of models. Given a set of prediction models S and a reference set R , and the consensus score, the CGDT-TS score of each model S_i is defined as:

$$CGDT-TS(S_i) = \sum_{r \in R} GDT-TS(S_i, S_r) / N \quad (2)$$

The reference set R is a subset of S and can be the same as S . CGDT-TS values also range from 0 to 1.

2) Energy or scoring functions—Many software programs have been developed using energy or scoring functions to evaluate the quality of predicted models. OPUS_Cα [17] has a knowledge-based potential function just using the information of C-α positions in a model. This software is based on seven major representative molecular interactions in proteins: distance-dependent pairwise energy with orientation preference, hydrogen bonding energy, short-range energy, packing energy, tri-peptide packing energy, three-body energy, and salvation energy. dDFIRE [18] treats each polar atom as a dipole and is based on the orientation angles in dipoles interactions and distance between two atoms dipoles. It considers the hydrogen bonding interactions, the physical dipole-dipole interactions, orientation-dependent interactions between polar and nonpolar atoms, and interactions between non-hydrogen-bonded polar atoms. It has an all-atom parameter-free statistical energy function. calRW [19] has two major functions: a) a pairwise distance-dependent atomic statistical potential function using an ideal random walk chain as reference state and b) a side chain orientation-dependent energy function. GOAP [20] is a generalized orientation and distance-dependent all-atom statistical potential that is determined by the relative orientations of the planes, which rely on each heavy atom in interacting pairs. It only considers the distance and angle information between representative atoms or blocks of side-chain or polar atoms. ProQ2 uses support vector machines to validate each residue quality and the global quality of protein models. It combines previously used features with updated structural and predicted features to evaluate the predicted models [21]. Because ProQ2 is the best single-model quality assessment (QA) method in CSAP10, it is used as one of the QA methods in our proposed algorithms to represent single-model QA methods. Its performance is compared with a consensus method within our new evolutionary algorithm framework.

B. Multi-dimensional scaling (MDS)

The contact map of a protein model is a two dimensional matrix, in which each value represents the distance between two residues' C-α atoms in the protein model. Under certain conditions the contact map can reconstruct the 3D coordinates of a protein [22]. Contact map predictions have been used in the modeling of protein 3D structures [23–26].

MDS is a set of data analysis techniques originated in psychometrics and psychophysics [27–29]. MDS starts with one or more dissimilarity matrices that are presumed to have been derived from points in a multidimensional space, and it finds a placement of the points in a low-dimensional space, where the distances between points resemble the original dissimilarities [30]. MDS is often used as part of exploratory data analysis or information visualization. It is also related to principal component analysis, factor analysis, and cluster

analysis. In this work, we use the weighted MDS (WMDS). The goal of WMDS is to find a configuration of points in a multidimensional space such that the inter-point distances are related to the provided proximities by some transformation (e.g., a linear transformation) while a weight of a distance entry corresponds to its importance. If the proximity data were measured without error in the Euclidean space, then WMDS would exactly recreate the configuration of points. Otherwise, it minimizes the sum of the weighted least squared errors between the proximity and the computed distances between all pair of points.

Using MDS, a 3D protein model could be constructed given a contact map. MDS can also be used to compare orthologous sequence sets [31], predict the protein models binding [32], and obtain improved, clash-free placement of loops obtained from a database of protein models [33].

C. MODELLER

MODELLER is a popular software program for protein modeling. It tries to satisfy spatial restraints deduced from homology to template structures and energy objective functions [34–36]. The spatial restraints include homology-derived restraints, stereo chemical restraints, statistical preferences for dihedral angles and non-bonded inter-atomic distances, and optional manually curated restraints. Those restraints presented as probability density functions are optimized by a combination of conjugate gradients and molecular dynamics with simulated annealing. Its basic inputs include an alignment of a sequence, the atomic coordinates of the templates and a script file. MODELLER calculates many distances and angle restraints from the alignment with the template tertiary structure and generates an atomic model.

III. THREE NEW EVOLUTIONARY ALGORITHMS FOR PROTEINMODELREFINEMENT

Evolutionary algorithms mimic natural evolution process to optimize an objective function, commonly called a fitness function, in a parametric space. They start with a population of individuals, select individuals for reproduction proportional to their fitness, use genetic operators such as crossover and mutations to generate new individuals to form a new population, and repeat these steps until a certain termination condition is met.

In this section, we present three new evolutionary algorithms within a general framework for protein model refinement. Figure 1 shows the framework, where the three algorithms differ in their crossover operators. One generation consists of three stages: protein model quality evaluation, selection, and crossover.

A. Initial pool of models

In our experiments on CASP datasets, the initial pool of models contains all predicted models submitted by different teams [37].

B. Evaluation

In the evaluation stage, redundant models and partial models are removed and the best 200 models are selected to form the initial population. In the subsequent generations, the best 200 models selected from a pool containing newly generated models and previous models forms a new population. One of two QA methods, ProQ2 or consensus GDT-TS, can be used to calculate the fitness value of a model. ProQ2 is a single-model scoring method, i.e., it can assess the quality of a single model, and is selected due to its good performance in CASP10. Figure 2 shows ProQ2 against true GDT-TS values (i.e., the true quality, the GDT-TS value of a model to its native structure.) of 200 models in the initial population of CASP target T0654. The correlation is weak.

The consensus GDT-TS method, CGDT-TS, needs a collection of models to assess the quality of one model. It performed very well in the CASP QA category, outperforming single-model QA methods by a large margin. It is based on pairwise GDT-TS values calculated using the TM-score software [38]. The true quality of a model is the GDT-TS value of it against the native structure of the protein. Figure 3 shows CGDT-TS against true GDT-TS (i.e., the true quality) values of 200 models in the initial population of CASP target T0654. The correlation is very strong.

C. Selection

Fitness proportional selection is a simple and widely used selection method in evolution algorithms. Individuals are selected for reproduction in proportional to their fitness values [39]. In our experiments, each new model is a child of a group of current models selected according to the selection scheme. The group of models is passed through a crossover operator, which is to be presented in the next subsection, to generate a new model. Thus, to generate n child models, n groups of models will be selected and the crossover operations will be executed n times.

D. Three New Crossover Operators

1) MDS-based method—The MDS-based crossover algorithm is shown in Fig. 4.

Given k models and their respective ProQ2 scores, the algorithm generates one new model. For each given model, its ProQ2 score contains a global score for the overall quality of the model and a vector of local scores representing the quality of each predicted C- α atom position.

First, the contact map of each model is calculated and the average contact map becomes a proximity matrix as the first input to the WMDS function. Then, the weight matrix for each model is computed based on the local ProQ2 scores of as follows:

$$WM = \begin{bmatrix} 1 & q_1 \times q_2 & \cdots & q_1 \times q_n \\ \vdots & \ddots & & \vdots \\ q_n \times q_1 & q_n \times q_2 & \cdots & 1 \end{bmatrix} \quad (3)$$

Where q_i is the local ProQ2 score of the i th C- α atom. The second input to the WMDS function is the average weight matrix of all given models. Residues with larger weights will have a stronger influence on the results.

In our implementation, the MATLAB function ‘MDSCALE’ (non-classical multidimensional scaling) is used to perform WMDS. “metricstress” is chosen as the default parameter when running the WMDS function. In the metric scaling, “MDSCALE” tries to find a configuration whose pairwise Euclidean distances approach the proximity input.

In the end, the program xyz2pdb is used to construct the PDB file of the new model from the coordinates generated by WMDS.

2) MODELLER-based method—The MODELLER-based crossover algorithm is shown in Figure 5. Given a set of models, the protein sequence, and a script configuration file, MODELLER generates one or more models. In our experiments, the default “automodel” modeling protocol in MODELLER is used.

3) Hybrid method—The hybrid crossover operator is shown in Figure 6. Given a set of models, their ProQ2 scores, the protein sequence, and a script file containing a MODELLER configuration, the function first runs the MDS-based crossover operator multiple times, e.g. 3 times, to generate multiple new models. Then, the MODELLER-based crossover operator is run once on these new models to generate one final model as output.

IV. EXPERIMENTAL RESULTS

In our experiments, CASP10 predicted models of 16 targets were used. The computational time and solution quality of the proposed evolutionary algorithms using the MDS-based, MODELLER-based, and hybrid crossover operator, respectively, are compared. The crossover operators use two or three models to create one new model. Table I shows the name for different EA methods. The results trend to be stable after several iterations. In addition, considering that the computation time should not be too long, all evolutionary algorithms in this experiment ran for 10 generations.

Figures 7 and 8 compare the computational times of 6 different algorithms. The results show that the MDS-based method is much faster than the MODELLER-based method because the MDS-based crossover operator is much faster. In all of these algorithms, the selection step took very little time. In terms of evaluation methods, ProQ2 is much faster than CGDT-TS. For example, comparing P_2P_W and C_2P_W, CGDT-TS used more than 8 minutes, whereas ProQ2 used less than 2 minutes, 4 times faster.

1) Using ProQ2 in evaluation

Figures 9, 10, and 11 compare the solution quality of three evolutionary algorithm settings, P_2P_W, P_3P_M, and P_2P_H, all using ProQ2 in the evaluation step. In Fig. 9, the true GDT-TS value of the best model in the final population is shown. P_2P_W improves over the initial models in 9 out of 16 cases and is the same in 5 cases. The most improvement happens on target T0680, where the best model after refinement has a true GDT-TS value of

0.833, much better than the initial best model, 0.763. P_3P_M improves over the initial models in 8 out of 16 cases, with the most improvement, a difference of 0.046 in true GDT-TS value, occurred on target T0662. P_2P_H performs slightly worse than P_3P_M, with the biggest improvement happened to target T0669, an increase of true GDT-TS value by 0.0412. On the other hand, both P_3P_M and P_2P_H perform poorly on T0700, with a final model much worse than the initial best model.

Figures 10 and 11 compare the solution quality in terms of the average true GDT-TS value of the best 10 and all models in the final population, respectively. For the best 10 cases, P_2P_W slightly improves the initial models on 13 targets. For example on target T0698, P_2P_W, P_3P_M, and P_2P_H raised the average true GDT-TS value by 0.0427, 0.0501 and 0.0358, respectively. On the other hand, they perform poorly on three targets. The average quality of total generated models was improved. Figure 11 shows that in terms of the average of all models, in most cases, these algorithms improve over the initial population significantly.

2) Using CGDT-TS in evaluation

Figures 12, 13, and 14 compare the solution quality of three evolutionary algorithm settings, C_2P_W, C_3P_M, and C_2P_H, all using CGDT-TS in the evaluation step. In Fig. 12, the true GDT-TS value of the best model in the final population is shown. The final solutions of these algorithms are generally worse than the initial best models. The result is similar for the best-10-models case, as shown in Fig. 13. In contrast, these algorithms improve the average true GDT-TS value of all models in the final population over the initial population, as shown in Fig. 14.

These refinement algorithms using CGDT-TS evaluation perform worse than using ProQ2. The reason is that CGDT-TS gives models most similar to other models get higher scores, leading to premature convergence.

Finally, Figure 15 shows the average improvement in terms true GDT-TS values for five different algorithm settings. P_2P_W improves over the initial population in terms of the best model, the average of best 10 models, and the average of all models. The results of the other algorithms are mixed. Using ProQ2 in the evaluation step is much better than using CGDT-TS.

The result shows that the MDS-based crossover is fast and can generate better solutions to refold existing predicted models can be a promising approach to improve the best predicted protein models' quality.

V. SUMMARY

The paper presents an evolutionary algorithm framework and three new crossover operators, MDS-based, MODELLER-based, and hybrid, for protein model refinement. Their performances are compared based on the ProQ2 and CGDT-TS evaluation. The MODELLER-based method is much slower than the MDS-based method. Using ProQ2 to evaluate models' quality is much faster than CGDT-TS. All methods improve the overall

quality of the population, whereas only P_2P_W improves the top 1 and top 10 models determined by comparing with the native structures. The MDS-based crossover operator is purely geometric-based and fast, and is a promising complement to energy function based methods.

In this study, the global ProQ2 score was used for quality assessment and the local ProQ2 score was used to construct the weight matrices. In our future work, other promising local score program such as IDDT [40] could be used.

Acknowledgments

The authors would like to thank valuable inputs from Son P. Nguyen, Jingfen Zhang, Zhiquan He, Hongbo Li, Jiong Zhang, Junlin Wang, and Wei Xiong.

This work was supported in part by NIH under Grant R01-GM100701.

References

1. Anfinsen C. Principles that govern the folding of protein chains. *Science*. 1973; 181:223–230. [PubMed: 4124164]
2. Anjum C. Protein Tertiary Model Assessment Using Granular. 2012
3. Reynaud E. Protein Misfolding and Degenerative Diseases. *Nature Education*. 2010; 3(9):28.
4. Johnson MS, Srinivasan N, Sowdhamini R, Blundell TL. Knowledge-based protein modeling. *Crit Rev BiochemMolBiol*. 1994; 29:1–68.
5. Bhattacharya D, Cheng J. i3Drefine Software for Protein 3D Structure Refinement and Its Assessment in CASP10. *PLoS ONE*. 2013; 8(7):e69648. [PubMed: 23894517]
6. Heo L, Park H, Seok C. GalaxyRefine: protein structure refinement driven by side-chain repacking. *Nucleic Acids Research*. 2013; 41:384–388.
7. João PGL, Rodrigues M, Levitt M, Chopra G. KoBaMIN: a knowledge-based minimization web server for protein structure refinement. *Nucleic Acids Research*. 2012; 40:323–328. [PubMed: 21908404]
8. Xu D, Zhang J, Roy A, Zhang Y. Automated protein structure modeling in CASP9 by I-TASSER pipeline combined with QUARK-based ab initio folding and FG-MD-based structure refinement. *Proteins*. 2011; 79(Suppl 10):147–160. [PubMed: 22069036]
9. Park H, Seok C. Refinement of unreliable local regions in template-based protein models. *Proteins*. 2012; 80:1974–1986. [PubMed: 22488760]
10. Park H, Ko J, Joo K, Lee J, Seok C, Lee J. Refinement of protein termini in template-based modeling using conformational space annealing. *Proteins*. 2011; 79:2725–2734. [PubMed: 21755541]
11. MacCallum JL, Hua L, Schnieders MJ, Pande VS, Jacobson MP, Dill KA. Assessment of the protein structure refinement category in CASP8. *Protein*. 2009; 77(Suppl 9):66–80.
12. MacCallum JL, Perez A, Schnieders MJ, Hua L, Jacobson MP, Dill KA. Assessment of protein structure refinement in CASP9. *Proteins*. 2011; 79(Suppl 10):74–90. [PubMed: 22069034]
13. Nugent T, Cozzetto D, Jones DT. Evaluation of predictions in the CASP10 Model Refinement Category. *Proteins: Structure, Function, and Bioinformatics*. 2013
14. Skolnick J. In quest of an empirical potential for protein structure. *CurrOpinStructBiol*. 2006; 16:166–171.
15. Wang Q, Shang Y, Xu D. Improving a Consensus Approach for Protein Structure Selection by Removing Redundancy. *IEEE/ACM transactions on computational biology and bioinformatics*. 2011; 8(6):1708–1715. [PubMed: 21519117]
16. Zemla A. LGA: a method for finding 3D similarities in protein structures. *Nucleic acids research*. 2003; 31:3370–3374. [PubMed: 12824330]

17. Wu Y, Lu M, Chen M, Li J, Ma J. OPUS-Ca: A knowledge-based potential function requiring only Ca positions. *Protein Science*. 2007; 16(7):1449–1463. [PubMed: 17586777]
18. Yang Y, Zhou Y. Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins*. 2008; 72:793–803. [PubMed: 18260109]
19. Zhang J, Zhang Y. A Novel Side-Chain Orientation Dependent Potential Derived from Random-Walk Reference State for Protein Fold Selection and Structure Prediction. *PLoS ONE*. 2010; 5(10):1–13.
20. Zhou H, Skolnick J. GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction. *Biophysical Journal*. 2011; 101:2043–2052.
21. Ray A, Lindahl E, Wallner B. Improved model quality assessment using ProQ2. *BMC Bioinformatics*. 2012; 13:224. [PubMed: 22963006]
22. Vassura M, Margara L, Di Lena P, Medri F, Fariselli P, Casadio R. Reconstruction of 3D Structures From Protein Contact Maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2008; 5(3):357–367. [PubMed: 18670040]
23. Tegge AN, Wang Z, Eickholt J, Cheng J. NNcon: Improved Protein Contact Map Prediction Using 2D-Recursive Neural Networks. *Nucleic Acids Research*. 2009; 37:w515–w518. [PubMed: 19420062]
24. Bacardit J, Widera P, Márquez-Chamorro A, Divina F, Aguilar-Ruiz JS, Krasnogor N. Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features. *Bioinformatics*. 2012; 28(19):2441–2448. [PubMed: 22833524]
25. Wang Z, Xu J. Predicting protein contact map using evolutionary and physical constraints by integer programming (extended version). *Bioinformatics*. 2013; 29(13):266–273. [PubMed: 23162058]
26. Ding W, Xie J, Dai D, Zhang H, Xie H, Zhang W. CNNcon: Improved Protein Contact Maps Prediction Using Cascaded Neural Networks. *PLoS ONE*. 2013; 8(4):e61533. [PubMed: 23626696]
27. Togerson, W. *Theory and methods of scaling*. New York: Wiley; 1958.
28. Abdi, H. Metric multidimensional scaling. In: Salkind, NJ., editor. *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage; 2007. p. 598-605.
29. Takane, Y.; Jung, S.; Oshima-Takane, Y. Multidimensional scaling. In: Millsap, R.; Maydeu-Olivares, A., editors. *Handbook of quantitative methods in psychology*. London: Sage Publications; 2009. p. 219-242.
30. Cox Trevor, F.; Cox Michael, AA. *Multidimensional Scaling*. Second. Chapman and Hall/CRC; 2000.
31. Pelé J, Abdi H, Moreau M, Thybert D, Chabbert M. Multidimensional Scaling Reveals the Main Evolutionary Pathways of Class A G-Protein-Coupled Receptors. *PLoS ONE*. 2011; 6(4):e19094. [PubMed: 21544207]
32. Mogi, H.; Taguchi, Y-h. Protein binding prediction using non-metric multidimensional scaling method. *Bioinformatics and Biomedicine Workshops (BIBMW)*; 2011 IEEE International Conference; Atlanta, GA. 2011.
33. Holtby D, Cheng Li S, Li M. LoopWeaver: Loop Modeling by the Weighted Scaling of Verified Proteins. *Journal of Computational Biology*. 2013; 20:212–223. [PubMed: 23461572]
34. Sali A, Blundell TL. Comparative Protein Modeling by Satisfaction of Spatial Restraints. *Journal of Molecular Biology*. 1993; 234(3):779–815. [PubMed: 8254673]
35. Fiser A, Do RK, Sali A. Modeling of loops in protein structures. *Protein Science*. 2000; 9:1753–1773. [PubMed: 11045621]
36. Eswar, N.; Marti-Renom, MA.; Webb, B.; Madhusudhan, MS.; Eramian, D.; Shen, M.; Pieper, U.; Sali, A. *Current Protocols in Bioinformatics*. Vol. Supplement 15. John Wiley & Sons, Inc.; 2006. Comparative Protein Structure Modeling With MODELLER; p. 5.6.1-5.6.30.
37. Wang Q, Vantasin K, Xu D, Shang Y. MUFOLD-WQA: A new selective consensus method for quality assessment in protein structure prediction. *Proteins*. 2011; 79(Suppl 10):185–195. [PubMed: 21997748]
38. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. *Proteins*. 2004; 57(4):702–710. [PubMed: 15476259]

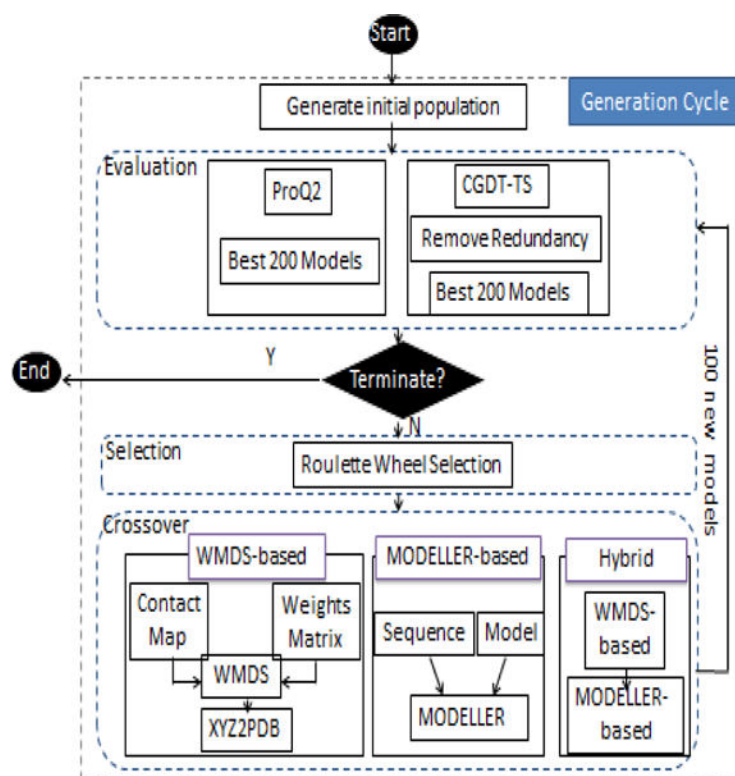
39. Norainii MR, Geraghty J. Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. 2011
40. Mariani V, Biasini M, Barbato A, Schwede T. IDDT:a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*. 2013; 29(21):2722–2728. [PubMed: 23986568]

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Fig. 1.**

The framework of the three new evolutionary algorithms for protein model refinement. The three algorithms differ in their crossover operators.

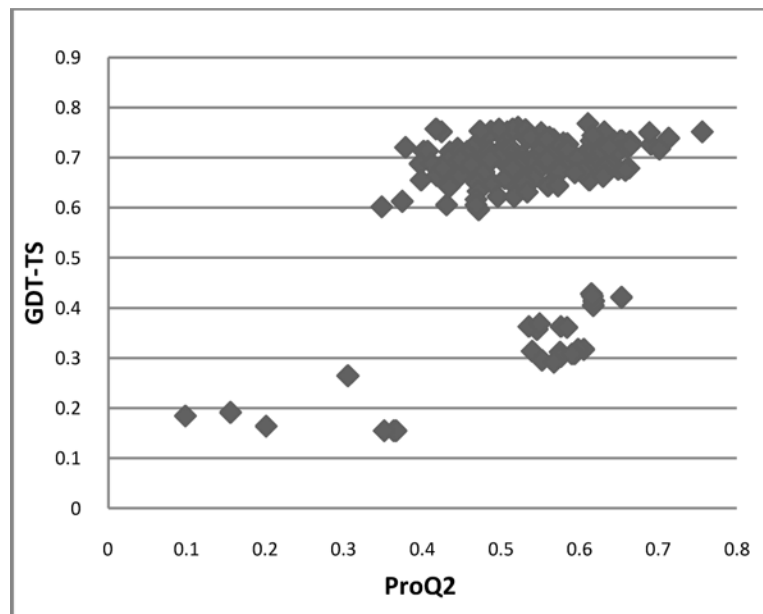


Fig. 2.

The corresponding ProQ2 and true GDT-TS values of the 200 models in the initial population for CASP target protein T0654.

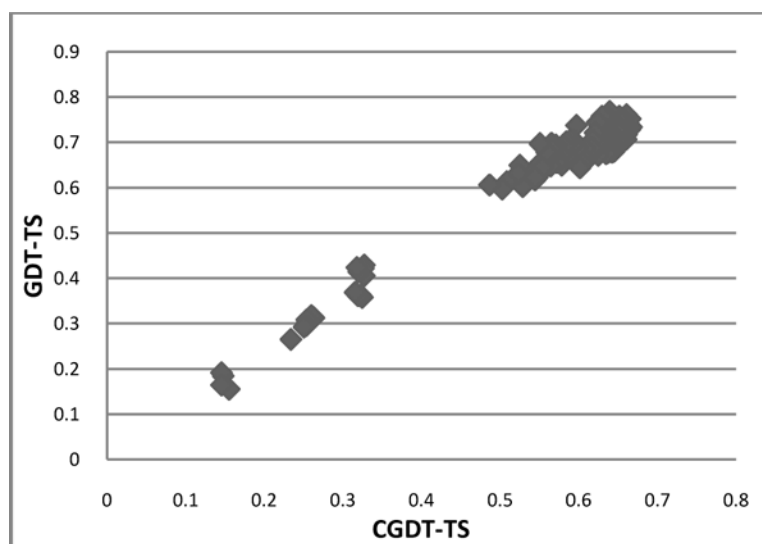


Fig. 3.
The corresponding CGDT-TS and true GDT-TS values of the 200 models in the initial population for CASP target protein T0654.

```

Function WMDS-C(Models, PrQ2S) returns a model
inputs: Models, a set of individuals
       PrQ2S, PrQ2 score for each individuals
K ← SIZE(Models)
for i = 1 to K do
    CMi ← Contact-Map(Modelsi)
    WMi ← Weights-Matrix(Modelsi, PrQ2Si)
end for
then XYZ ← WMDS( $\sum_{i=1}^K \frac{CM_i}{K}$ ,  $\sum_{i=1}^K \frac{WM_i}{K}$ )
    PDB ← XYZ2PDB(XYZ)
return PDB file of individual model

```

Fig. 4.
The MDS-based crossover operator.


```
Function MODELLER-C(Models, SeqF, ScriptsF) returns a model
inputs: Models, a set of individuals
        SeqF, Sequences file
        ScriptF, Script file for MODELLER
do run MODELLER(Models, SeqF, ScriptsF)
return the best model generate by MODELLER
```

Fig. 5.
The MODELLER-based crossover operator.

```

Function Hybrid-C(Models, ProQ2S, SeqF, ScriptsF) returns a model
  NK ← SIZE(Models)
  N ← Number of Parents run MODELLER-C
  for i = 1 to N do
    K ← Number of Parents run WMDS-C
     $M_i$  ← WMDS-C( $Models_k$ ,  $ProQ2S_k$ )
  end for
  Nmodels ← set of M generate by WMDS-C
  then run MODELLER-C(NModels, SeqF, ScriptsF)
  return the best model generate by MODELLER

```

Fig. 6.

The hybrid crossover operator using both MDS-based and MODELLER-based crossover operations.

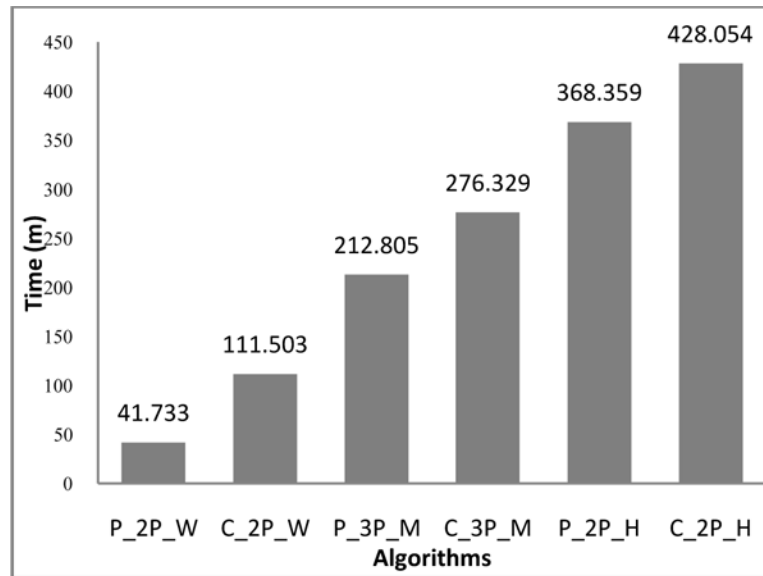
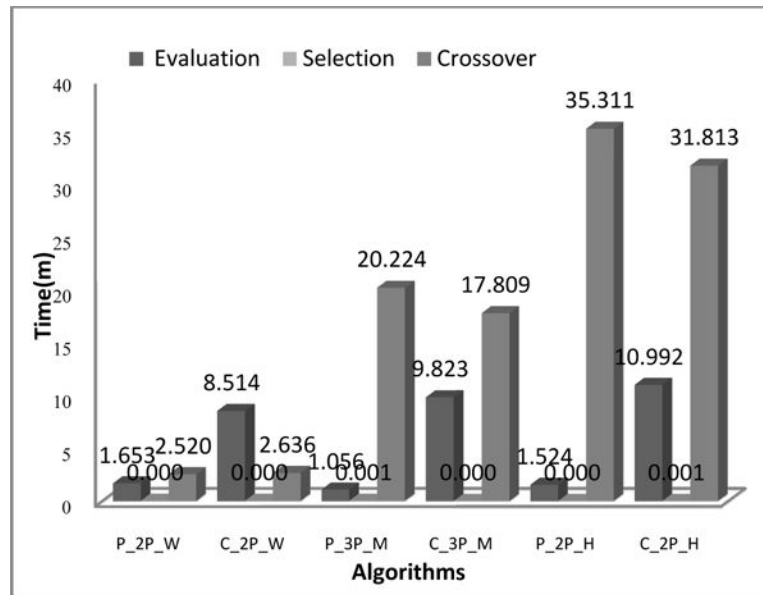


Fig. 7.
Comparison of computational times(in minutes) of 6 different algorithms.

**Fig. 8.**

The breakdown of computation times (in minutes) of 6 different algorithms.

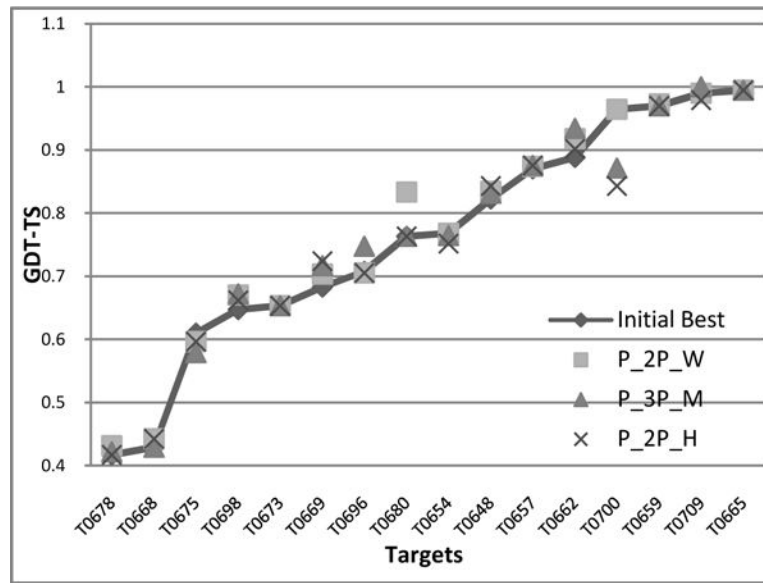


Fig. 9.

Comparison of solution quality, the true GDT-TS value of the best model in the final population, of three evolutionary algorithms against the best model in the initial population, all using ProQ2 in the evaluation step.

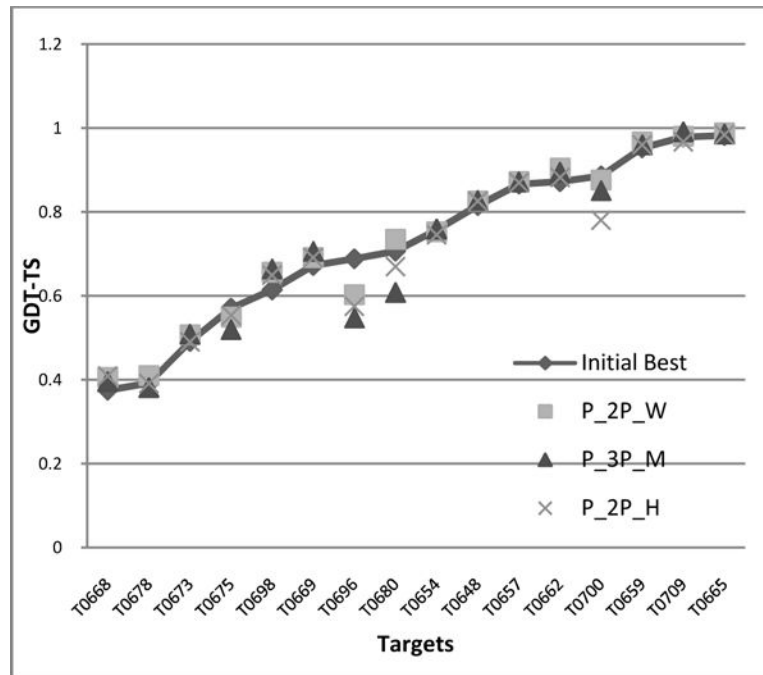


Fig. 10.

Comparison of solution quality, the average true GDT-TS value of the 10 best models in the final population, of three evolutionary algorithms against the initial population, all using ProQ2 in the evaluation step.

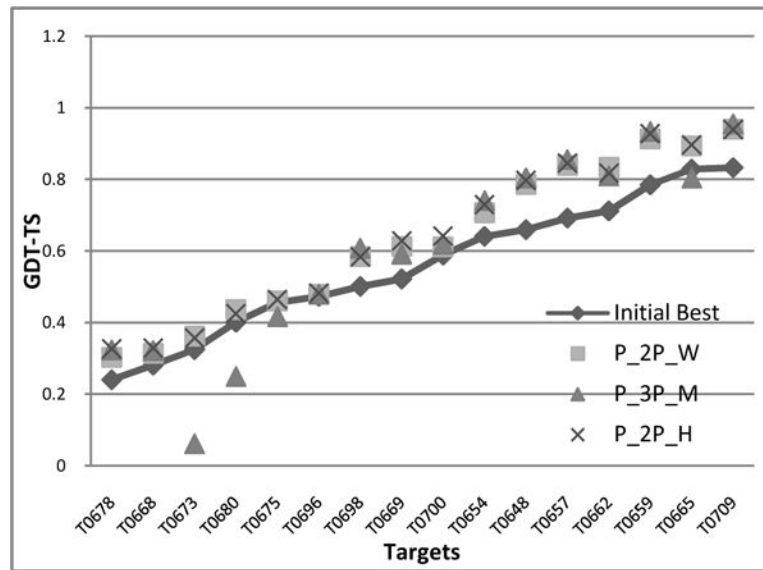


Fig. 11.

Comparison of solution quality, the average true GDT-TS value of all models in the final population, of three evolutionary algorithms against the initial population, all using ProQ2 in the evaluation step.

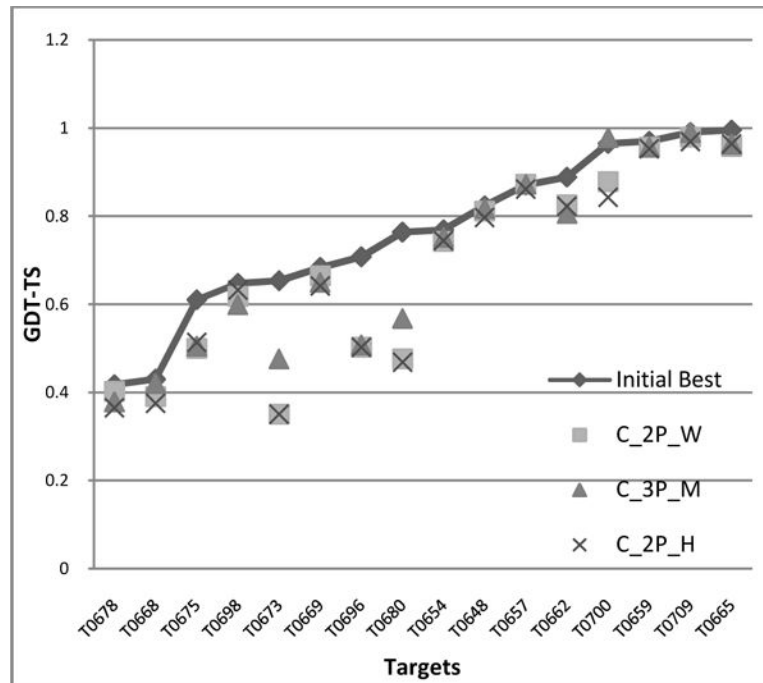


Fig. 12.

Comparison of solution quality, the true GDT-TS value of the best model in the final population, of three evolutionary algorithms against the best model in the initial population, all using CGDT-TS in the evaluation step.

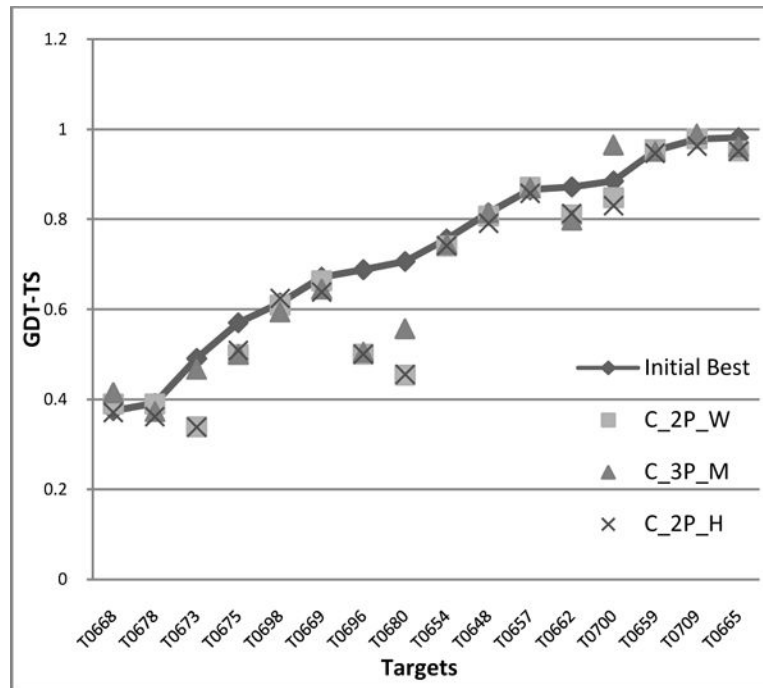


Fig. 13.

Comparison of solution quality, the average true GDT-TS value of the best 10 models in the final population, of three evolutionary algorithms against the initial population, all using CGDT-TS in the evaluation step.

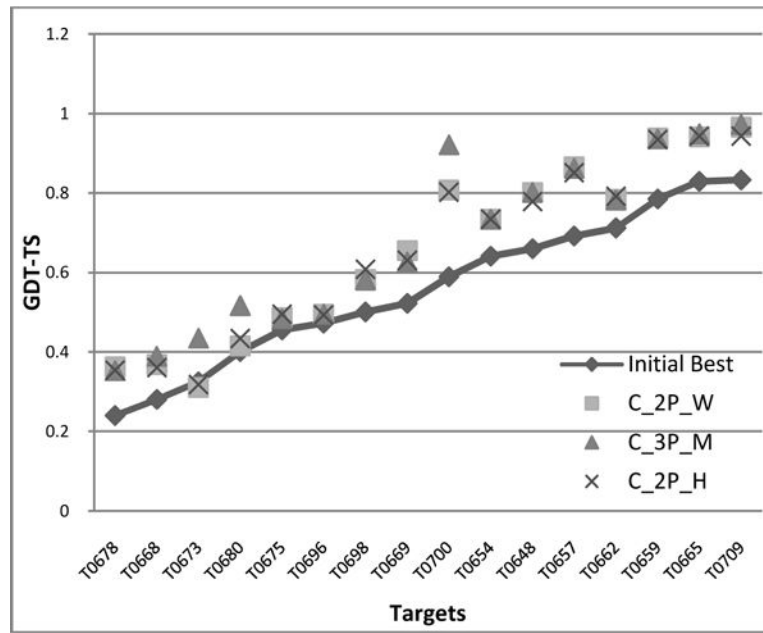


Fig. 14.

Comparison of solution quality, the average true GDT-TS value of all models in the final population, of three evolutionary algorithms against the initial population, all using CGDT-TS in the evaluation step.

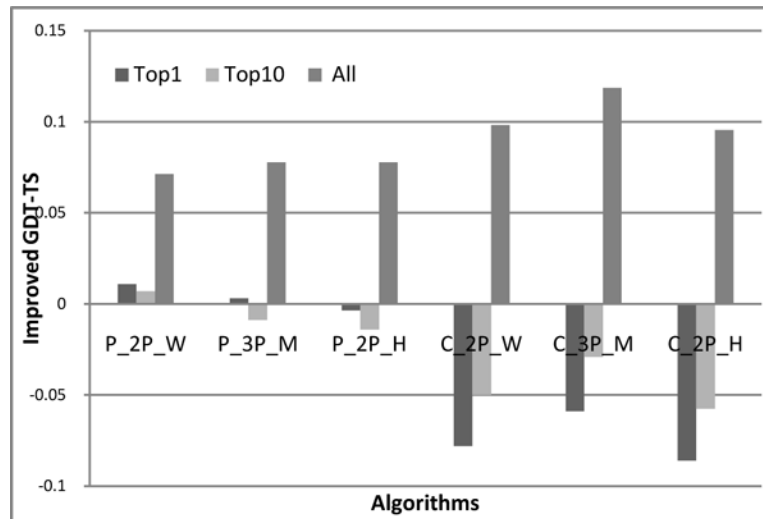


Fig. 15.

Summary of improvement of 5 different evolutionary algorithm settings over initial models. A positive value means an algorithm's final solution improves its input, whereas a negative value means its solution is worse than its input.

TABLE I**NAMING FOR DIFFERENT METHODS**

Evaluation		Num of Parents	Crossover Operator
P_2P_W	ProQ2	2	WMDS_based
P_3P_M	ProQ2	3	MODELLER_based
P_2P_H	ProQ2	2	Hybrid
C_2P_W	CGDT-TS	2	WMDS_based
C_3P_M	CGDT-TS	3	MODELLER_based
C_2P_H	CGDT-TS	2	Hybrid