Research Article Open Access

Matthew Conover, Max Staples, Dong Si, Miao Sun, and Renzhi Cao*

AngularQA: Protein Model Quality Assessment with LSTM Networks

https://doi.org/10.1515/cmb-2019-0001
Received September 9, 2018; accepted May 1, 2019

Abstract: Quality Assessment (QA) plays an important role in protein structure prediction. Traditional multi-model QA method usually suffer from searching databases or comparing with other models for making predictions, which usually fail when the poor quality models dominate the model pool. We propose a novel protein single-model QA method which is built on a new representation that converts raw atom information into a series of carbon-alpha (Cα) atoms with side-chain information, defined by their dihedral angles and bond lengths to the prior residue. An LSTM network is used to predict the quality by treating each amino acid as a time-step and consider the final value returned by the LSTM cells. To the best of our knowledge, this is the first time anyone has attempted to use an LSTM model on the QA problem; furthermore, we use a new representation which has not been studied for QA. In addition to angles, we make use of sequence properties like secondary structure parsed from protein structure at each time-step without using any database, which is different than all existed QA methods. Our model achieves an overall correlation of 0.651 on the CASP12 testing dataset. Our experiment points out new directions for QA problem and our method could be widely used for protein structure prediction problem. The software is freely available at GitHub: https://github.com/caorenzhi/AngularQA

1 Introduction

Protein folding prediction proves to be a major hurdle in modern biology (Wei and Zou 2016). While the rate at which genomes can be sequenced has grown rapidly with the advent of automated systems, protein structures have still been limited to expensive, experimental observation through Nuclear Magnetic Resonance or X-ray crystallography (Jacobson and Sali 2004). While great progress has been made in computational prediction methods with the help of machine learning techniques (Manavalan et al. 2017; Lai et al. 2017; Peterson et al. 2017; Shin, Christoffer, and Kihara 2017; D. Li, Ju, and Zou 2016; Wei et al. 2015; Dao et al. 2018; C.-Q. Feng et al. 2018; Chen et al. 2019; Tang et al. 2018; Yang et al. 2018; Huang, Smolensky, et al. 2018; Huang, Zhang, et al. 2018; Manavalan, Basith, et al. 2018; Basith et al. 2018; Manavalan, Shin, et al. 2018; Chen et al. 2017; P.-M. Feng et al. 2013), a long journey still remains.

As biology and medicine progresses, the need for a method of reliably and efficiently predicting tertiary protein structures becomes more apparent. Perhaps the most promising use of ab initio folding prediction is in the use of functional prediction and drug discovery (Jacobson and Sali 2004).

Matthew Conover: Department of Computer Science, Pacific Lutheran University, Tacoma, WA 98447, USA,
E-mail: conovemk@plu.edu
Max Staples: Department of Computer Science, Pacific Lutheran University, Tacoma, WA 98447, USA,
E-mail: staplema@plu.edu
Dong Si: Division of Computing and Software Systems, University of Washington-Bothell, Bothell, WA 98011, USA,
E-mail: dongsi@uw.edu
Miao Sun: JingChi, Sunnyvale, CA 94089, USA, E-mail: sun.miao@jingchi.ai
***Corresponding Author: Renzhi Cao:** Department of Computer Science, Pacific Lutheran University, Tacoma, WA 98447, USA,
E-mail: caora@plu.edu

Open Access. © 2019 Matthew Conover et al., published by De Gruyter. This work is licensed under the Creative Commons Attribution alone 4.0 License.

The prediction process can be divided into two parts, first generating a model of the target based on its sequence, and then determining how accurate the generated model is (J. Li, Cao, and Cheng 2015).

Protein structure prediction is usually categorized as being either template-based modeling such as DeepFold (J. Li, Cao, and Cheng 2015; Liu et al. 2017), FALCON (Wang et al. 2015), MTMG (J. Li and Cheng 2016), and I-TASSER (Roy, Kucukural, and Zhang 2010); or template-free (ab initio) modeling such as QUARK (Roy, Kucukural, and Zhang 2010; Xu and Zhang 2012) and UniCon3D (Bhattacharya, Cao, and Cheng 2016).

Especially with ab initio modeling, the challenge is then to assess and rank the generated models to help improve prediction, and to know when an acceptable model has been generated (J. Li, Cao, and Cheng 2015).

Furthermore, these QA methods can be subdivided into two distinct approaches. The first is consensus, which considers many generated models and seeks out patterns to predict which one is the best. This has been shown to work very well with a good dataset generated by multiple different methods, but can be a bad predictor with a poor data-set or small pool and is computationally costly, often requiring $O(n^2)$ computations where n is the number of models (Cao et al. 2016) (Cao, Wang, and Cheng 2014). Such methods include Pcons5 (Wallner and Elofsson 2005) and ModFOLDClust2 (McGuffin, Buenavista, and Roche 2013).

The second form of QA, the focus of our research, is single protein assessment. Rather than attempting to score a protein relative to others, the goal is to consider it alone and predict how close it is to the unknown, native structure. Such methods of single-protein assessment include DeepQA which makes use of deep belief networks (Cao et al. 2016; Zou et al. 2019), ProQ3 which combines the results from Rosetta energy functions using full-atom and centroid models and the ProQ2 SVM (Uziela et al. 2016), SVMQA which uses a support vector machine to process 19 extracted features (Manavalan and Lee 2017), RFMQA (Manavalan, Lee, and Lee 2014) which ranks model based on random forest, and GMQ (Shin et al. 2017)evaluates local quality based on spatially neighboring residues using a graph representation.

What makes AngularQA especially interesting, is it bypasses most of the costs associated with using two or three dimensional data, reducing a complicated protein with thousands of atoms into a sequence of amino acids, angles, secondary structure, and proximity counts. In addition, only observable features are used in our method further cutting setup costs. The ability of AngularQA is that of its features, without reliance on other, unreliable predictions. This combined with its new methods and use of new features makes it not only a novel approach to single protein quality assessment, but also means it should be of high value to composite QA approaches.

2 Method and Implementation

The core of our machine learning model, is a LSTM network which processes each residue and its associated information as a time-step before finally generating an estimated score of the model accuracy. To the best of our knowledge, we are the first to use a recurrent neural network in protein QA.

2.1 Initial Data Preparation

All data used in training comes from 3DRobot decoys generated by The Yang Zhang Lab (Deng, Jia, and Zhang 2016) and from CASP 9, 10, and 11 (Moult et al. 1995). These have 92,535, 36,083, 15,901, and 14,193 models respectively from which we draw for training. Validation occurs on the CASP12, of which we use 6,790 models across 40 targets (Moult et al. 1995).

We begin by filtering all the models. During this process we verify the residue sequences in the predicted structures line up correctly with the native structure, and throw out any predicted models with gaps in the center. We also trim the beginnings and ends of the model to line up with the native. In addition, We throw out any models for which we do not have the native structure. After filtering, we are left with a total of 128,439 models with 121,875 training models and 6564 validation models.

We then calculate the global distance test (GDT) scores using the Local-Global Alignment program which superimposes two protein models and assesses the similarity between them (Zemla 2003). Presently, we do not use local alignment scores in training or validation of our method and focus on global quality prediction. All scores are calculated by comparing a given model to the native structure.

Next, we calculate the angles and bond lengths along the backbone and side-chain as was described by UniCon3D; the result is a sequence of angle and bond length information provided for each residue following along the carbon backbone (Bhattacharya, Cao, and Cheng 2016). This representation is central to our method, and has not been extensively studied for its usefulness in QA applications. Because the length of the angle sequence is the same as the length of the protein sequence, the angle information fits well with a recurrent neural network which can work well with inputs of varying lengths (Hochreiter and Schmidhuber 1997).

The proximity counts are also calculated at this time by counting the number of Cα atoms within a set radius of each residue's Cα atom. We perform this calculation for all radii in the discrete range [5Å, 15Å].

Finally the secondary structures are calculated for all the models using the DSSP program which does not predict the structure, but interprets what is displayed in the predicted model [DSSP]. From its output, we extract only the secondary structure, one of Alpha Helix, Beta Bridge, Strand, Helix-3, Helix-5, Turn, Bend, or if no structure is assigned, Random Coil. The result, is for each residue in the sequence, there is assigned the secondary structure it forms.

2.2 Run-Time Data Preparation

With the great variance allowed for by the PDB format, sometimes one of the initial steps fails for a certain model or group of models, we have added error checking and handling when loading the data to catch inconsistencies and cases where one or more parts may be outright missing. These cases impact less than 1% of models, so we have chosen to identify them, and ignore them. The CASP12 dataset has proven a convenient exception to these challenges, and from our experience, has no such issues.

Before we use the data, we normalize all the angle values we calculate to be in the range [0, 1] and trim the first and last residues from each model. The values at both ends we found to have extreme values in many cases which are not related to the sequence.

Additionally, because proteins have different numbers of residues, we pad the data to make the lengths consistent for training; the model itself later masks zero values to counteract this. We choose to pad at a length of 500 as most proteins in our dataset are shorter. Of the 688 targets in our dataset (including CASP12), they have a distribution of N(180.0, 119.6) with only 13 longer than 500 (This average is based on the length of the observed structures which are sometimes missing a few residues at the front or end.). Those few which are longer drop the last residues.

A notable benefit to our representations, is their small space requirement; when loaded, all our datasets combined, roughly 160,000 models, use less than 4GB of memory even when padded to a length of 500 residues.

2.3 Features

Each time-step includes information about that residue. The amino acid type is considered one of the most fundamental and is included with all tests.

The core features we use are the angles between residues. To verify this new representation is of value, we calculated the correlation between the different angles—Tau, Theta, Phi, and Delta—and the related GDT score finding weak correlations for both Cα angles ($r_\tau = 0.373$, $r_\Theta = 0.427$) and lesser correlations for both side-chain angles ($r_\phi = 0.187$, $r_\delta = 0.299$); these results indicate the angles within and between residues could be a good feature for assessing predicted structures.

DSSP determines the secondary structures at each of the residues based on their 3D form and is then fed in along side the amino acid in addition to the results from Proxcalc, an in-house program to calculate and count the residues within a given radius (Joosten et al. 2011).

We have tried different combinations of these, and for now have settled on using the amino acid, theta, tau, the secondary structure, protein properties, and proximity counts for radii of 8 Å and 12 Å.

2.4 LSTM Network

The network begins by masking the zero values of padded proteins. Followed after, the data for each time-step is separated into three parts: the amino acid, the secondary structure, and physical protein properties (hydrophobic, polar, or charged) and the angles and bond lengths. The amino acid and secondary structure are then converted to dense-vector encodings, mapping the value to n-dimensional space—in both cases we use four dimensions.

With the vectorized values, they are then reunited with the other data, forming the true LSTM input for each time-step. The LSTM layers vary in breadth and depth, but a common configuration we test which achieves highly, is a [64, 32, 10] arrangement. The first two layers return a sequence of values, one for each time step, while the final layer outputs a single value at the end of the residue series. Each LSTM cells uses a hyperbolic tangent activation with a hard sigmoid recurrent activation.

The output of the final layer then gets run through a single layer of perceptrons with sigmoid activation and learned weights before being converted to a single value in the range (0, 1).

The network is trained using RMSprop with a learning rate of 0.0001. Stochastic gradient descent and Nadam were also tried, but neither proved as stable or effective. The final output value would be predicted GDT score for input protein structure.

3 Results and Discussion

Thus far, we have achieved an overall correlation of 0.684 on CASP12 with an average loss of 0.122. These results were found using three LSTM layers in the configuration [128, 64, 32] which was trained for less than 154 epochs on all models from 3DRobot and CASP 9, 10, and 11.

Table 1 and Table 2 demonstrate the Pearson Correlation and loss before and after trimming Stage 1 and Stage 2 datasets for CASP12. We found a surprising difference between running on the CASP12 data which had been trimmed to the native structure (If the native was missing part of the sequence at the beginning or end, it was removed for all trimmed tests.) versus our results with the raw predictions. Of course, using the trimmed data requires more information than is available in real-world and thus represents no more than an interesting comparison.

In addition, we compare performance of our method AngularQA with few selected top performing methods from CASP12. Table 3 describes the average per-target Pearson Correlation and loss for our method AngularQA and four selected top performing single-model QA methods on Stage 1. We could see that DeepQA method achieves the best performance among all methods on average correlation metric. It is not very surprising that DeepQA performs better than our method, because our method only uses the information from the model, such as angle information. DeepQA ustilized 16 features with the help of deep belief network to

Table 1: Comparison of correlations between datasets on Stage 1 and Stage 2 of CASP12. Overall correlation uses all data points while the average correlation is the mean of correlation scores.

Dataset	Overall Corr.	Avg. Corr.	Avg. Corr. on Stage 1	Avg. Corr. on Stage 2
CASP12 trimmed	0.684	0.469	0.545	0.393
CASP12 untrimmed	0.651	0.439	0.502	0.377

Table 2: Comparison of loss scores between datasets on Stage 1 and Stage 2 of CASP12.

Dataset	Ave. Loss	Loss on Stage 1	Loss on Stage 2
CASP12 trimmed	0.122	0.116	0.128
CASP12 untrimmed	0.138	0.148	0.128

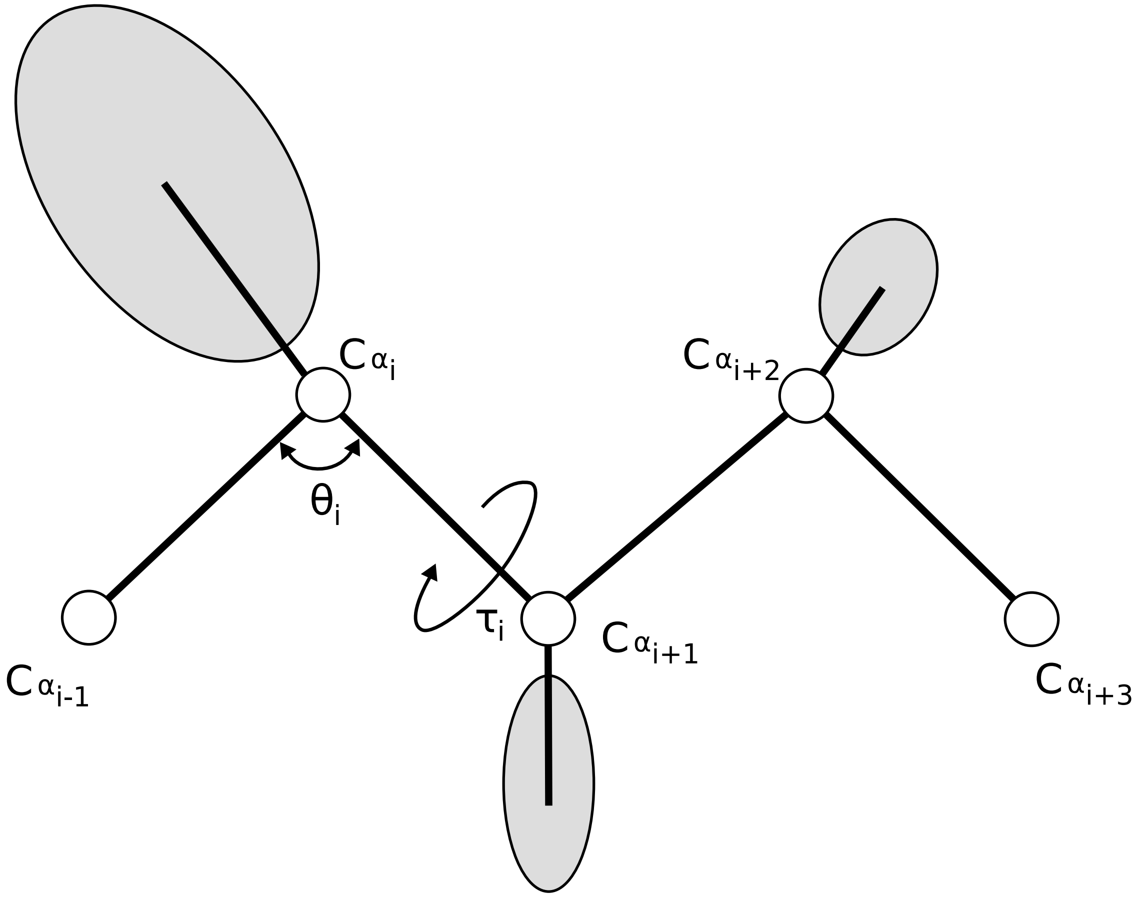

Figure 1: Diagram of the LSTM network components and data flow.

achieve the best performance. We did not even use the secondary structure prediction from the sequence, which is usually a very useful feature for model quality assessment. We did that so that our method runs much faster than any other methods, and our method could be used to generate new features for future new tools. At the same time, we also find out that AngularQA performs better than other two methods Wang1 and QMEAN. The similar pattern is found in Stage 2 datasets, which is shown in Table 4. Overall, based on our experiment, our AngularQA method demonstrated a great potential of using angle information for model quality assessment, and our method could be even furtherly used as a feature for top performing QA method and improve the accuracy of these QA method.

We would like to mention the importance of secondary structure feature in our model contributes more in the output. For example, without using secondary structure feature, the average correlation score on our testing CASP dataset is 0.24 with a loss 0.11, and the performance significantly improves with the help of secondary structure feature. At the same time, we would like to highlight that the secondary structure feature we used is different than all other methods. Our secondary structure feature is only based on protein structure

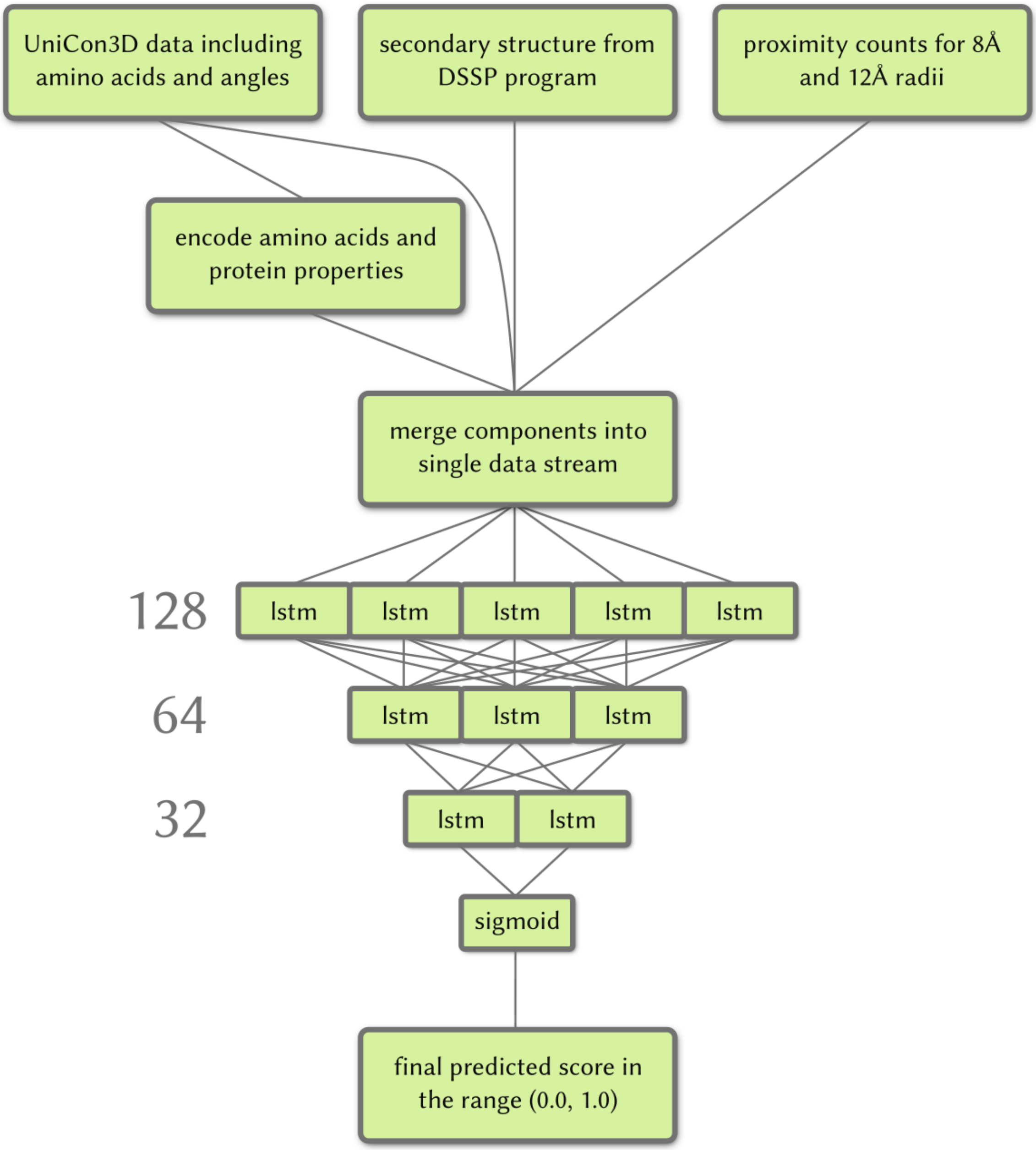
UniCon3D data including amino acids and angles
secondary structure from DSSP program
proximity counts for 8Å and 12Å radii
encode amino acids and protein properties
merge components into single data stream
128
64
32
lstm
sigmoid
final predicted score in the range (0.0, 1.0)

Figure 2: Representation of angles and bond-lengths. Each Cα is interpreted as an individual time-step for the network.

model without using any protein secondary structure prediction from protein sequences, so that our method is much faster than all other methods, which makes it possible to rank a large number of protein structure models. In addition, the output score generated by our method could be considered as a new feature for future QA method development.

Table 3: The performance of the global quality predictions of our AngularQA method and four selected CSP12 methods in terms of average correlation, and average loss of top 1 models ranked by each method, evaluated on CASP12 Stage 1 targets.

	Avg. Correlation on Stage 1	Avg. Loss on Stage 1
AngularQA	0.545	0.116
ProQ3	0.638	0.048
DeepQA	0.654	0.078
Wang1	0.462	0.170
QMEAN	0.342	0.174

Table 4: The performance of the global quality predictions of our AngularQA method and four selected CSP12 methods in terms of average correlation, and average loss of top 1 models ranked by each method, evaluated on CASP12 Stage 2 targets.

	Avg. Correlation on Stage 2	Avg. Loss on Stage 2
AngularQA	0.393	0.128
ProQ3	0.616	0.068
DeepQA	0.578	0.100
Wang1	0.256	0.144
QMEAN	0.292	0.125

4 Conclusions

Our results show great promise for the use of both angle information in QA, as well as recurrent neural networks. The angle correlations we calculated show they can be a useful metric in protein quality assessment. Our work demonstrates these values are generalizable between models of the same target, and more importantly to new, unseen targets.

Interestingly, before we added the secondary structure information, the overall correlations between the true and predicted scores for all attempted networks were below 0.3 before adding the secondary structure information and were often closer to 0.15. This indicates the secondary structure helps the model assess the validity of angles and determine if the overall is coherent.

While we have tested different combinations and ablations of both features and layer setups, much work remains to optimize the system as a whole. The thing we found to have had the largest impact beyond the features and data, was the learning rate which we ended up reducing by a two factors of ten. Before doing so, we found the models trained in very few epochs, somewhere between 10 and 40 in most cases. Since reducing the learning rate we have found the network to perform less well on training data, but to perform much better on testing data, and begin overfitting after a couple hundred epochs.

To help reduce the rate of drift in training, we wanted to increase the number of models we were using for validation, however, when we tried blindly splitting the training set for validation, we found the model scored very highly while training it, but when we went to test it against only CASP12, its performance was far worse than without the extra testing data points. This indicates it was able to recognize similar structures to what it was trained on and adequately assess them. To reduce this issue, we could make sure data splits take entire targets at a time rather than individual models preventing it from training on models from some of the same targets it will later be validated on.

In future, new features could be added to further help the system, such as the physical properties of the amino acids or contact information. We have also considered changing the way it runs, possibly adding a bidirectional LSTM system to consider the sequence from both ends. In addition, it is also interesting to train a model for different type of protein in future.

Overall, there are many possibilities left to try, and our work shows QA based on angles and using recurrent neural networks has great promise.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

RC conceived the project. MC, RC designed the project. MC and RC implemented and tested the tool. MC, DS, MS, RC wrote the manuscript. All the authors read and approved the manuscript.

Acknowledgements

The authors would like to thank support from students and faculties at Pacific Lutheran University. Also, we thank the anonymous referees for their valuable comments and helpful suggestions. This work is supported by Natural Sciences Undergraduate Research Program at Pacific Lutheran University, Karen Hille Phillips Regency Advancement Award to Prof. Cao, and the Division of Natural Science at Pacific Lutheran University.

References

Basith, Shaherin, Balachandran Manavalan, Tae Hwan Shin, and Gwang Lee. 2018. "iGHBP: Computational Identification of Growth Hormone Binding Proteins from Sequences Using Extremely Randomised Tree." Computational and Structural Biotechnology Journal 16 (October): 412–20.

Bhattacharya, Debswapna, Renzhi Cao, and Jianlin Cheng. 2016. "UniCon3D: De Novo Protein Structure Prediction Using United-Residue Conformational Search via Stepwise, Probabilistic Sampling." Bioinformatics 32 (18): 2791–99.

Cao, Renzhi, Debswapna Bhattacharya, Jie Hou, and Jianlin Cheng. 2016. "DeepQA: Improving the Estimation of Single Protein Model Quality with Deep Belief Networks." BMC Bioinformatics 17 (1): 495.

Cao, Renzhi, Zheng Wang, and Jianlin Cheng. 2014. "Designing and Evaluating the MULTICOM Protein Local and Global Model Quality Prediction Methods in the CASP10 Experiment." BMC Structural Biology 14 (April): 13.

Chen, Wei, Hao Lv, Fulei Nie, and Hao Lin. 2019. "i6mA-Pred: Identifying DNA N6-Methyladenine Sites in the Rice Genome." Bioinformatics, January. https://doi.org/10.1093/bioinformatics/btz015.

Chen, Wei, Hui Yang, Pengmian Feng, Hui Ding, and Hao Lin. 2017. "iDNA4mC: Identifying DNA N4-Methylcytosine Sites Based on Nucleotide Chemical Properties." Bioinformatics 33 (22): 3518–23.

Dao, Fu-Ying, Hao Lv, Fang Wang, Chao-Qin Feng, Hui Ding, Wei Chen, and Hao Lin. 2018. "Identify Origin of Replication in Saccharomyces Cerevisiae Using Two-Step Feature Selection Technique." Bioinformatics. https://doi.org/10.1093/bioinformatics/bty943.

Deng, Haiyou, Ya Jia, and Yang Zhang. 2016. "3DRobot: Automated Generation of Diverse and Well-Packed Protein Structure Decoys." Bioinformatics 32(3):378–87.

Feng, Chao-Qin, Zhao-Yue Zhang, Xiao-Juan Zhu, Yan Lin, Wei Chen, Hua Tang, and Hao Lin. 2018. "iTerm-PseKNC: A Sequence-Based Tool for Predicting Bacterial Transcriptional Terminators." Bioinformatics, September. https://doi.org/10.1093/bioinformatics/bty827.

Feng, Peng-Mian, Wei Chen, Hao Lin, and Kuo-Chen Chou. 2013. "iHSP-PseRAAAC: Identifying the Heat Shock Protein Families Using Pseudo Reduced Amino Acid Alphabet Composition." Analytical Biochemistry 442 (1): 118–25.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." Neural Computation 9 (8): 1735–80.

Huang, Qiuyuan, Paul Smolensky, Xiaodong He, Li Deng, and Dapeng Wu. 2018. "Tensor Product Generation Networks for Deep NLP Modeling." In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). https://doi.org/10.18653/v1/n18-1114.

Huang, Qiuyuan, Pengchuan Zhang, Dapeng Wu, and Lei Zhang. 2018. "Turbo Learning for CaptionBot and DrawingBot." In Advances in Neural Information Processing Systems 31, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, 6456–66. Curran Associates, Inc.

Jacobson, Matthew, and Andrej Sali. 2004. "Comparative Protein Structure Modeling and Its Applications to Drug Discovery." In Annual Reports in Medicinal Chemistry, 259–76.

Joosten, Robbie P., Tim A. H. te Beek, Elmar Krieger, Maarten L. Hekkelman, Rob W. W. Hooft, Reinhard Schneider, Chris Sander, and Gert Vriend. 2011. "A Series of PDB Related Databases for Everyday Needs." Nucleic Acids Research 39 (Database issue): D411–19.

Lai, Hong-Yan, Xin-Xin Chen, Wei Chen, Hua Tang, and Hao Lin. 2017. "Sequence-Based Predictive Modeling to Identify Cancerlectins." Oncotarget 8 (17): 28169–75.

Li, Dapeng, Ying Ju, and Quan Zou. 2016. "Protein Folds Prediction with Hierarchical Structured SVM." Current Proteomics 13 (2): 79–85.

Li, Jilong, Renzhi Cao, and Jianlin Cheng. 2015. "A Large-Scale Conformation Sampling and Evaluation Server for Protein Tertiary Structure Prediction and Its Assessment in CASP11." BMC Bioinformatics 16 (October): 337.

Li, Jilong, and Jianlin Cheng. 2016. "A Stochastic Point Cloud Sampling Method for Multi-Template Protein Comparative Modeling." Scientific Reports 6 (May): 25687.

Liu, Yang, Qing Ye, Liwei Wang, and Jian Peng. 2017. “Learning Structural Motif Representations For Efficient Protein Structure Search.” https://doi.org/10.1101/137828.

Manavalan, Balachandran, Shaherin Basith, Tae Hwan Shin, Sun Choi, Myeong Ok Kim, and Gwang Lee. 2017. “MLACP: Machine-Learning-Based Prediction of Anticancer Peptides.” Oncotarget 8 (44): 77121–36.

Manavalan, Balachandran, Shaherin Basith, Tae Hwan Shin, Leyi Wei, and Gwang Lee. 2018. “mAHTPred: A Sequence-Based Meta-Predictor for Improving the Prediction of Anti-Hypertensive Peptides Using Effective Feature Representation.” Bioinformatics, December. https://doi.org/10.1093/bioinformatics/bty1047.

Manavalan, Balachandran, and Jooyoung Lee. 2017. “SVMQA: Support–vector-Machine-Based Protein Single-Model Quality Assessment.” Bioinformatics 33 (16): 2496–2503.

Manavalan, Balachandran, Juyong Lee, and Jooyoung Lee. 2014. “Random Forest-Based Protein Model Quality Assessment (RFMQA) Using Structural Features and Potential Energy Terms.” PloS One 9 (9): e106542.

Manavalan, Balachandran, Tae Hwan Shin, Myeong Ok Kim, and Gwang Lee. 2018. “PIP-EL: A New Ensemble Learning Method for Improved Proinflammatory Peptide Predictions.” Frontiers in Immunology 9 (July): 1783.

McGuffin, Liam J., Maria T. Buenavista, and Daniel B. Roche. 2013. “The ModFOLD4 Server for the Quality Assessment of 3D Protein Models.” Nucleic Acids Research 41 (Web Server issue): W368–72.

Moult, J., J. T. Pedersen, R. Judson, and K. Fidelis. 1995. “A Large-Scale Experiment to Assess Protein Structure Prediction Methods.” Proteins 23 (3): ii – v.

Peterson, Lenna X., Woong-Hee Shin, Hyungrae Kim, and Daisuke Kihara. 2017. “Improved Performance in CAPRI Round 37 Using LZerD Docking and Template-Based Modeling with Combined Scoring Functions.” Proteins, August. https://doi.org/10.1002/prot.25376.

Roy, Ambrish, Alper Kucukural, and Yang Zhang. 2010. “I-TASSER: A Unified Platform for Automated Protein Structure and Function Prediction.” Nature Protocols 5 (4): 725–38.

Shin, Woong-Hee, Charles W. Christoffer, and Daisuke Kihara. 2017. “In Silico Structure-Based Approaches to Discover Protein-Protein Interaction-Targeting Drugs.” Methods 131 (December): 22–32.

Shin, Woong-Hee, Xuejiao Kang, Jian Zhang, and Daisuke Kihara. 2017. “Prediction of Local Quality of Protein Structure Models Considering Spatial Neighbors in Graphical Models.” Scientific Reports 7: 40629.

Tang, Hua, Ya-Wei Zhao, Ping Zou, Chun-Mei Zhang, Rong Chen, Po Huang, and Hao Lin. 2018. “HBPred: A Tool to Identify Growth Hormone-Binding Proteins.” International Journal of Biological Sciences 14 (8): 957–64.

Uziela, Karolis, Nanjiang Shu, Björn Wallner, and Arne Elofsson. 2016. “ProQ3: Improved Model Quality Assessments Using Rosetta Energy Terms.” Scientific Reports 6 (October): 33509.

Wallner, Björn, and Arne Elofsson. 2005. “Pcons5: Combining Consensus, Structural Evaluation and Fold Recognition Scores.” Bioinformatics 21 (23): 4248–54.

Wang, Chao, Haicang Zhang, Wei-Mou Zheng, Dong Xu, Jianwei Zhu, Bing Wang, Kang Ning, Shiwei Sun, Shuai Cheng Li, and Dongbo Bu. 2015. “FALCON@home: A High-Throughput Protein Structure Prediction Server Based on Remote Homologue Recognition.” Bioinformatics 32 (3): 462–64.

Wei, Leyi, Minghong Liao, Xing Gao, and Quan Zou. 2015. “Enhanced Protein Fold Prediction Method Through a Novel Feature Extraction Technique.” IEEE Transactions on Nanobioscience 14 (6): 649–59.

Wei, Leyi, and Quan Zou. 2016. “Recent Progress in Machine Learning-Based Methods for Protein Fold Recognition.” International Journal of Molecular Sciences 17 (12): 2118.

Xu, Dong, and Yang Zhang. 2012. “Ab Initio Protein Structure Assembly Using Continuous Structure Fragments and Optimized Knowledge-Based Force Field.” Proteins 80 (7): 1715–35.

Yang, Hui, Hao Lv, Hui Ding, Wei Chen, and Hao Lin. 2018. “iRNA-2OM: A Sequence-Based Predictor for Identifying 2’-O-Methylation Sites in Homo Sapiens.” Journal of Computational Biology: A Journal of Computational Molecular Cell Biology 25 (11): 1266–77.

Zemla, Adam. 2003. “LGA: A Method for Finding 3D Similarities in Protein Structures.” Nucleic Acids Research 31 (13): 3370–74.

Zou, Quan, Pengwei Xing, Leyi Wei, and Bin Liu. 2019. “Gene2vec: Gene Subsequence Embedding for Prediction of Mammalian -Methyladenosine Sites from mRNA.” RNA 25 (2): 205–18.