université PARIS DIDEROT PARIS 7

Meet-U

Paris Diderot University

Meet-U 2018-2019 competition

Fold U :
A Protein Structure Prediction Program

Authors
Gabriel Cretin, Hélène Kabbech, Tom Gutman,
Flora Mikaeloff & Franz-Arnold Ake

January 2, 2019

Contents

1 Introduction

Proteins are in the core of the central Dogma of Biology. Studying proteins and their folding mechanisms is necessary to have a general understanding of this Dogma. Structural biology is the science that focuses on those features. This science is fairly new and has begun during the 1960's with Max Perutz and John Kendrew who obtained the Nobel Prize in 1962 for their pioneer work in determining the structure of globular proteins. Structural biology is facing several major problems such as how is the 3D native structure of a protein determined by the physico-chemical properties that are encoded in its 1D amino-acid sequence [1].

To challenge those problems, structural biology teams are competing every second summer in the Critical Assessment of protein Structure Prediction (CASP). This competition aims at establishing the current state of the art in protein structure prediction by evaluating the current methods in protein modelling. Given sequence information, participants submit in turn protein structure models. CASP also aims at identifying what progress has been made, and highlighting where future effort may be most productively focused [2]. This subject is therefore still in great development, that is why this project, in the scope of Meet-U competition, will also focus on the problematic of protein folding. Meet-U is a collaborative pedagogical and bioinformatics research initiative between several Universities of Paris area [3]. This year, the objective of the project is to determine given a protein primary sequence, what is the most probable/stable 3D fold adopted by the protein in solution. This initiative is mimicing in a way the CASP experiment. To do so, the project unfolds in two steps: (i) domain annotation based on profile-profile comparison and (ii) identification of the most stable 3D fold by threading.

Our team is composed by master students in bioinformatics at Paris Diderot University. The objective of our team was to develop the second step (downstream) of this protein structure prediction project which mainly consists of first, threading a query sequence on different given templates and generates several other scoring functions to re-rank those templates in order to improve the models.

2 Material and Methods

2.1 Global strategy

The main idea of the project is to use profile-profile based-method alignments to rank templates for a given query (upstream team), followed by a threading technique (among others) in order to re-rank the templates (downstream team, us). It was indeed shown in the literature that profile-profile based-method alignments demonstrates dominant advantage and generates models with average TM-score 26.5% higher than sequence-profile methods and 49.8% higher than sequence-sequence alignment methods [4]. The accuracy of profile-profile alignments can be improved by 9.6% or 21.4% when predicted or native structure features are incorporated [4]. Also, 3D structure prediction of proteins can be greatly improved by using the power of evolutionary information found in patterns of correlated mutations in protein sequences [5, 6].

Therefore, as the downstream team we decided to develop several other scores in addition to threading score in order to take into account the evolutionary couplings between residues, structural information such as solvent accessibility and secondary structure, as well as the global accuracy of models conformations with a model energy score. The ultimate goal is to combine the information and specificity brought by each type of score into a global score, weighted by a simple machine learning approach, and the re-ranking being done according to this weighted combined score.

2.2 Fold-U Implementation strategy

Our program takes in input the protein sequence of the studied query (fasta format) and an uniref database. After running the SALUT program (Upstream) developed by the Team 1, for each alignments the query sequence is threaded on the template and a threading score is generated using an energy DOPE matrix. Then, the program MODELLER generates a new refined 3D model by homology using the aligned sequence as template and returns a high resolution DOPE energy score. This new model is then used to calculate the solvent accessibility and co-evolution scores. A secondary structure score is also calculated using the secondary structures assigned or predicted from the foldrec file. Each scores are normalized using the min-max scaling method (values between 0 and 1) in order to be able to simply sum them into a sum_score at first. We use machine learning (logistic regression) to calculate a weighted combined score and obtain a better ranking. Finally, the scores are stored in a comma separated file (scores.csv) and the top N models are generated (See the pipeline at Figure 1).

2.2.1 Protein Threading

Protein threading is based on two basic observations. The number of unique folds in nature is fairly small (1375) and almost 100% of the new structures submitted to the PDB since 2009 have similar structural folds to ones already in the PDB [7].

Threading is a method for protein modelling. Also known as **fold recognition**, the idea of this technique is to physically "thread" a sequence of amino acids (target sequence) onto a backbone structure (a fold), here the template selected by the alignment of the upstream team, and to evaluate this proposed 3-D structure using a set of pair potentials. The DOPE (Discrete Optimized Protein Energy) statistical potential was used. DOPE uses the statistical knowledge of the relationship between the structures deposited in the Protein Data Bank (PDB) and the sequence of the target protein. It is based on an improved reference state that corresponds to non-interacting atoms in a homogeneous sphere with the radius dependent on a sample native structure [8].

More specifically, threading was performed by calculating the distances between the pairs of the target's amino acids, based on the template's coordinates, forming a matrix of distances. Distances were then converted into energies using the DOPE matrix (bin (precision) of 0.5Å). A threading score was then generated.

2.2.2 MODELLER to refine pdb structures

MODELLER [9] is a protein structure prediction program based on homology modelling (as threading technique it is template-based). Homology modelling is a method of protein modelling which is used to model proteins which have the same fold as proteins of known structures, but do not have homologous proteins with known structure. As a comparative protein modelling method, MODELLER is thus designed to find the most probable structure for a sequence given its alignment with the template's related structure. The three dimensional (3D) model is obtained by optimally satisfying spatial restraints derived from the alignment and expressed as probability density functions (pdfs) for the features restrained. Because of the nature of this project, no particularly deeply refined model was required. That is why MODELLER was executed to generate one model with the option "very fast" to get an approximate model quickly. This uses only a very limited amount of variable target function optimisation with conjugate gradients, and thus is roughly 3 times faster than the default procedure, which is sufficient for the project.

The only known structure that can be used as a template is the template which is aligned with the query sequence. In this particular project case, we seek to uncover the global folding of the query protein only. To do so, MODELLER was used to generate new 3D models by homology (alignment between query and template). A score, also based on the DOPE statistical potential was generated. It is the high resolution DOPE score returned by MODELLER.

2.2.3 CCMPred to calculate co-Evolution score

CCMPred is a tool predicting protein residue-residue contacts from correlated mutations[10]. It has been determined as one of the best tool to predict amino acids contacts [6]. First, a matrix of scores is calculated between each pair of amino acids according to an alignment file. If two amino acids are found correlated in a maximum of sequences the score will be higher. Then, best couples are selected based on score and distance : couples must be separated by at least 4 amino acids. According to literature, the number of top couplings must be L (length of the sequence), L/2, L/5 or L/10. We obtained the most efficient results with L/2. CCMPred is used on the query and new PDBs of the models obtained with MODELLER to check if the predicted structure of the query matches the evolutionary predictions. It determines which amino acids are evolving together and thus are most probably in contact and compare it to the template. The CCMPred score is therefore a score of adequacy between the predicted structure and evolutionary predictions made by CCMPred.

2.2.4 DSSP to calculate solvent accessibility score

Define Secondary Structure of Proteins (DSSP) is a program that assigns secondary structures to the amino acids of a protein, given the atomic-resolution coordinates of the protein from a PDB entry [11][12]. It also calculates the solvent accessibility that we used to generate a score. The solvent accessibility score is an adequacy score. The relative solvent accessibility (RSA) is compared between the modeller model and the template. Residues having an RSA $> 25\%$ were considered as exposed, the others as buried.

2.2.5 Implemented scores

In addition to the **alignment score** resulting from the profile-profile alignment step of the upstream team, several additional scores have been implemented in order to improve the ranking of models :

- **Threading Score**

  This score, based on the threading of the query on the template using the Discrete Optimized Protein Energy (DOPE, with a bin of 0.5 Å) matrix was deduced as the sum of energies for the structure. With an eye on calculation time optimization and knowing that the matrix is symmetric, only the upper half of the matrix was taken into account for calculations. The interactions (distances) outside the range of [5, 15] Å(interactions between an atom and either itself, the one next to him, or another one too far) were also ignored because not informative.

- **MODELLER Score**

  To assess the model generated by MODELLER the DOPE statistical potential (with a high resolution: bin of 0.125Å) was used in order to stay consistent with the scoring method of threading presented in the previous section. Globally, MODELLER is therefore expected to return better DOPE scores than threading.

- **Co-evolution Score**

  Co-evolution score measures co-occurrence of a pair of amino acids in ortholog sequences using CCMpred program. This program determines which amino acids are evolving together and are most probably in contact. It compares it to the template and generates a **score of adequacy** between the predicted structure and the target sequence it-self. Globally, the score of co-evolution has an important error rate. In some cases, it assesses the quality of the template made by Modeller.

- **Secondary structure score**

  Based on the predicted secondary structure and its associated confidence score generated by PSIPRED and the DSSP secondary structures assignation, templates and queries respectively are compared and a score is generated. We consider only PSIPRED predictions with a confidence ≥ 7 on the scale 0-9 because it was observed that such a threshold is attributed to 53% of the residues with more than 81% of the Q3 accuracy [13] and we want secondary structure predictions that have a Q3 accuracy of 80% at least [14].

- **Solvent Accessibility score**

  Solvent accessibility is a key property of amino acid residues, important for both the structure and function of proteins.
  In most cases, proteins are in an aqueous environment, which means that water atoms can touch residues at the surface of a protein. The area of an atom on the surface that can be touched by water is called the accessible molecular surface, or solvent-exposed area. The area covered by the "center" of a water molecule rolling over the surface of an exposed protein atom can also be calculated. This area is larger, of course, and is called the accessible surface. DSSP program can calculate the relative solvent accessibility (RSA) for each residues. this program was used on the template and the model to generate an adequacy score. This score was based on the number of residues with an RSA greater than 25% [15][16] in common between the two structures. The more residues in common the better.

- **Blosum based mutation score**

  Some mutations exist between the query and the aligned template sequence. To take advantage of this, a "blosum" score was calculated using the BLOSUM62 matrix [17]. This matrix contains substitution scores for each amino acid pair. A positive score is given to the more likely substitutions while a negative score is given to the less likely substitutions. The less mutations existing between the query and the template the closer the structures of those two proteins are. Early testing and results showed that this score was not informative and was always either random or worse at finding benchmarks, so it was not taken into account in the project. We used co-evolution instead for sequence-based information.

- **Combined Score**

  The simple sum of all our scores (except Blosum score).

- **Machine learning score**

  This score is the sum of the previous scores (except combined and blosum score) to which were applied a weight determined by logistic regression.

2.2.6 Machine learning: Generalized Linear Models (GLM)

The first naive idea was to combine all the scores together with a simple summation to get a global score. However, the early findings suggested that in some cases some scores were better at ranking the benchmarks. Simply because scores are more or less specific to different kinds of proteins (all alpha, all beta, mixed etc.). This brought us to think of a way to **weight** the different scores in order to increase the quality of the ranking by reducing or increasing the impact of a score depending on its capacity to rank the benchmarks of different types of proteins.

A solution was to use logistic regression, a machine learning method [18], on the data generated by the 20 benchmarks to learn and calculate the "best weights" associated to each score. The models were validated with a leave-one-out cross-validation (CV) method (*cv.glm* from the R package *boot*[19]). Not a K-fold cross-validation because of the small size of training dataset. The cross-validation method returns two values called "delta". The first component of "delta" is the average mean-squared error (the raw cross-validation estimate of prediction error) . The second component is the adjusted cross-validation estimate. The adjustment is designed to compensate for the bias introduced by not using leave-one-out cross-validation, but since we do a leave-one-out cross-validation, we can omit this value.

Machine learning was applied on a benchmarking dataset of results (20 sequences) using a General Linear Model (GLM, logistic regression), in order to determine the weights of each score that optimize the discovery of benchmarking sequences. It was attended that each score would not necessarily bring the same amount of information, so this method can identify which score is "weak" and which one performs best and apply lower or higher weights respectively.

The weights were then integrated into the program to calculate a new global score: weighted_combined_score.

2.3 Benchmarking

We benchmarked our program using Enrichment style plots and Top N information to evaluate the power and the relevance of the different scores. Using the scores.csv file a dataframe was generated. The score results were generated for all queries. The benchmarking plot represents the cumulative sum of benchmarks encountered along the ranking (from rank 1 to rank 397) for each calculated scores. A top N result table is also generated showing the number of "Family", "Superfamily" and "Fold" benchmark found in the top N ranks. The benchmark type corresponds to a degree of similarity with the query.

2.4 Choice of the upstream team

The choice of the upstream team was based on several criteria upon which we paid special attention without them being absolute constraints:

- The team had to present a clear GitHub repository with a comprehensive README in order to facilitate the merge of the codes and the understanding of their strategy.
- A properly formatted foldrec file complying with the requested one.
- The global-local ("glo-loc") alignment approach is more relevant since the HOMSTRAD database, where the data is coming from, contains only protein domains and the query can have one or several domains [20].
- An essential element for our group is the use of a program to assign the secondary structure of templates (DSSP program) and another one to predict the secondary structure of queries (PSIPRED program [21]), because we developed a secondary structure-based score which demonstrated good results. If only PSIPRED is used for both queries and templates, it would lower the performance of this score because obviously reducing the precision.
- A properly formatted multiple alignment file for the co-evolution score (CCMPred tool).

After discussions and time comparing the different team programs, we decided to use the SALUT program implemented by the Team 1. This team was carefully selected according to the previously described criteria, excepted one, the "glo-loc" alignment technique. Only one team used glo-loc (Team 2) but did not used DSSP program to assign secondary structure of templates. Team 1 was one of the only team to use DSSP and to have a properly formatted multiple alignment file for our co-evolution score.

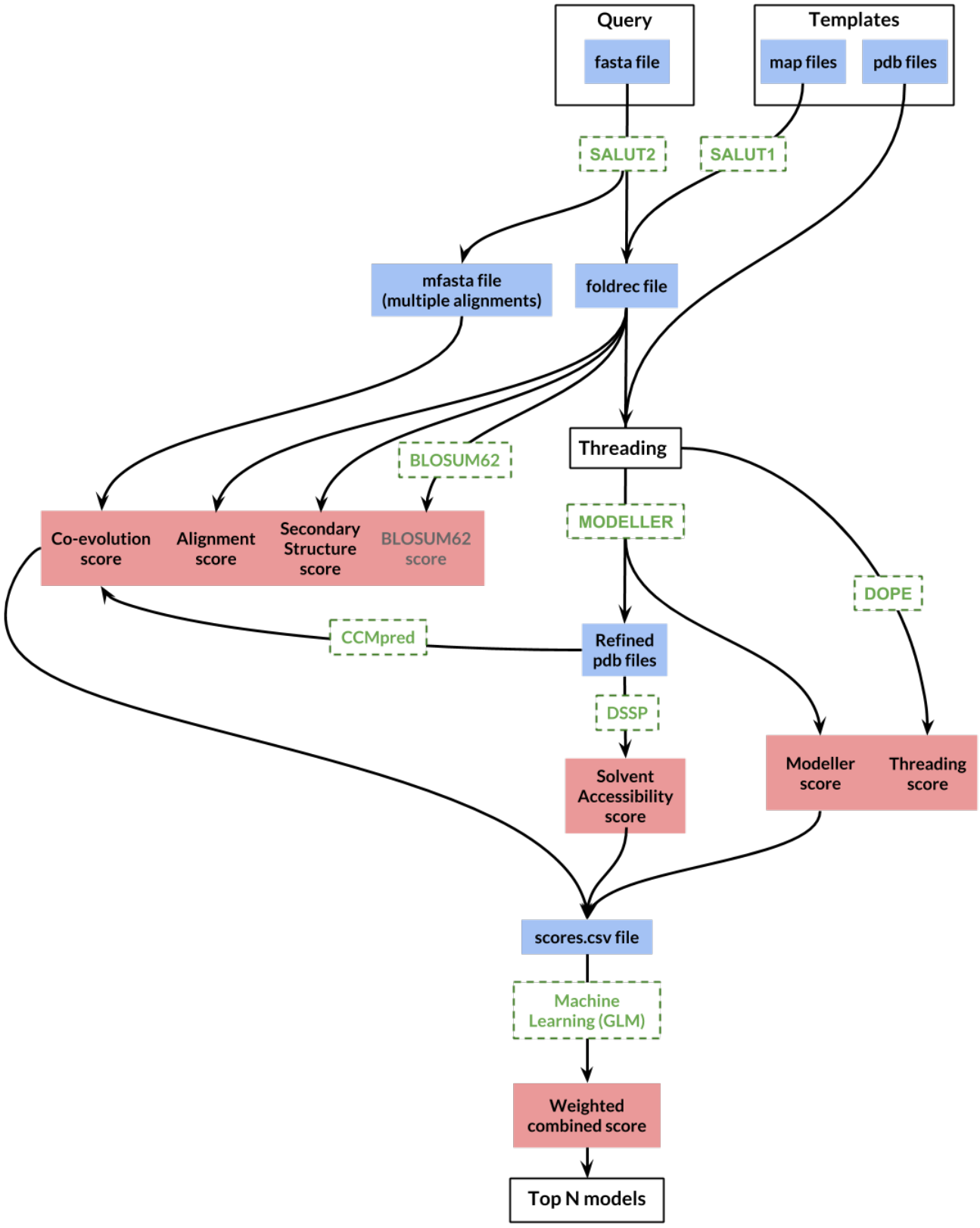

Figure 1: Pipeline summarizing the strategy of the Fold-U program

2.5 Programming tools

2.5.1 Programming language chosen

The Fold-U program is written in Python 3 with the use of following standard Python libraries : Pandas (v.0.23.4), Numpy (v.1.15.2), Biopython (v.1.72), Docopt (v.0.6.2), Schema (v.0.6.8), Tqdm (v.4.28.1), matplotlib (v.2.2.2). We decided to implement our program in oriented object (see Figure 2). The Machine Learning part is written in R.

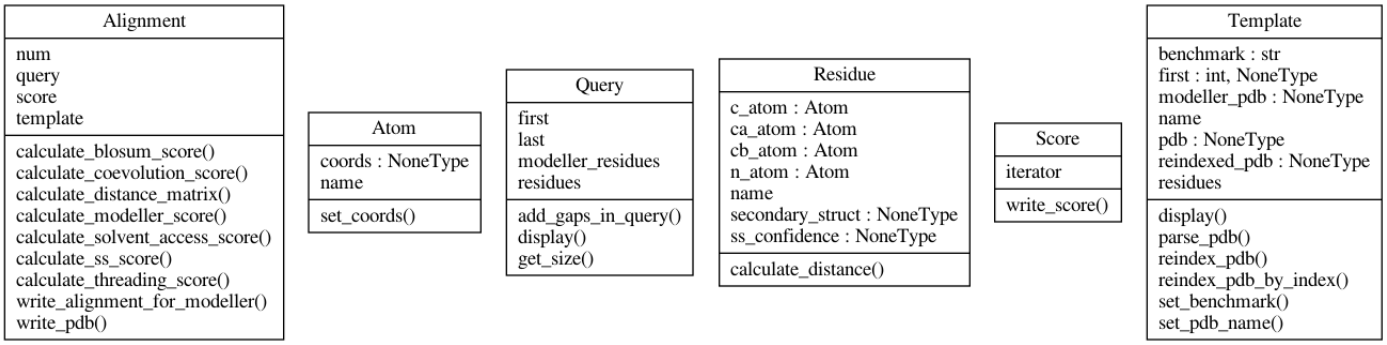

Figure 2: Implemented classes of the Fold-U program.

2.5.2 Online version control

GitHub has been used for the development of this program. It is a web-based hosting service for version control using Git. Therefore, this program (its documentation and source code) is available online at the following address : https://github.com/meetU-MasterStudents/Fold_U.

2.5.3 PEP8 Code Quality Checker

Pylint and PyCodeStyle for Python 3 was used to check the code quality of the Fold U scripts regarding the PEP8 Python style guide recommendations. The program was scored 9.27/10.

2.5.4 Documentation

The documentation of Fold U, available at the following address: https://fold-u.readthedocs.io, was generated with Sphinx and continuously built on Read The Docs.

2.5.5 Gain process time with a cluster

In order to gain process time we run our program on Google Cloud Platform, using a Virtual Machine instance composed of 8 SkyLake vCPUs, 15 GB of RAM and 30 GB of storage.

3 Results

3.1 Results for the 21 query tests

In order to test our program, we had 21 queries for which the structure is already known and a benchmark template associated to each query. The benchmark has a SCOP id close to the query's one and the program is eventually supposed to rank the benchmark at top 1 as the best case.

3.1.1 Run of the Fold-U program on the 21 queries

Our results and the quality of scores highly depend on the alignment realised by the upstream team. After merging our part with the SALUT program implemented by the team 1, the full program was run with a total run-time of 14 hours 37 minutes and 37 seconds.

20 out of 21 queries were analysed. The query LRR had to be removed because of the lack of efficient sequences in the multiple alignment, which makes it impossible for CCMpred to run and thus no co-evolution score would have been available for this query. For each other templates, all scores were calculated.

Also, the database of templates had to be cleaned because eight of them had a pdb structure containing more than one chaine A, which caused issues when running MODELLER for example. The following templates were thus ignored: "bromodomain", "rhv", "Peptidase_A6", "ins", "Arg_repressor_C", "SAM_decarbox", "prc", "Chorismate_mut". This reduces automatically the amount of templates from 405 to 397. The program also skips empty alignments in the foldrec file, so depending on the alignments of a query, less templates can be actually used.

3.1.2 Scores comparison

An enrichment plot was generated (See Figure 3). The curves represent the cumulative sum of benchmarks encountered along the ranking (from rank 1 to rank 397) for each calculated score.

According to this plot, alignment and solvent accessibility are the best scores. Threading and modeller scores are less efficient and performing very similarly. This was expected because both scores are based on the same DOPE statistical potential. Secondary structure and co-evolution scores are less informative and sometimes are performing worse than randomly. We can make the hypothesis that either the algorithms used are not correct or parameters should be tweaked in order to improve these scores. Another possibility is that the quality of the alignment highly impairs the quality of these scores. The global score (sum_score, sum of all scores) is good and very close from the solvent accessibility score until it reaches around the 110^{th} rank with a total score of 12/20 benchmarks encountered.

The last score, the weighted combination of all other scores except sum_score, was generated according to the weights proposed by the model of the logistic regression: $weighted\_combined\_scores = -10.8256(intercept) + 4.5026{*}alignment - 0.0764{*}threading + 1.0713{*}modeller + 3.8989{*}secondary\ structure - 2.7788{*}solvent\ access - 1.2846 * co - evolution$. The weights show that some scores are clearly more significant than others (alignment and secondary structures) whereas solvent accessibility and co-evolution for example have small weights. Nevertheless it is interesting to notice that the solvent accessibility curve shows good results even though its weight is small. This can be explained by the fact that the solvent accessibility score compares RSAs of the residues of a template and a model which was generated from this template. So it is not informative enough for the machine learning model but it performs correctly at assessing good models. Surprisingly, all scores and more especially co-evolution score curves plummet at around 160^{th} rank. After some investigation it can be explained by the fact that this ranking area contains many proteins with SCOP ids of the class "g" such as kazal (g.68.1.1), planttoxin (g.3.7.5) or GAL4 (g.38.1.1), which stands for "small proteins". For contact predictions, amongst others, it has been suggested that the number of sequences required to produce reliable contact predictions should be of the order of 5xL, where L is the length of the protein[6].

The weighted combined score improves the ranking compared to the un-weighted combined score. Indeed, at first, combined and weighted combined scores curves are similar but at rank 110th, instead of getting closer to the random curve this new curve is way over the other curves. All benchmarks are encountered faster thus the re-ranking is enhanced. However for the top 20, less benchmarks are encountered than for the un-weighted combined scores. This machine learning method therefore makes the program more sensitive and less specific,

which is more relevant as the program is now performing better for diverse kinds of proteins.

A box plot describing the distribution of values for each scores for the 20 benchmarks was generated (See Figure 4). Each box represent the standard deviation above and below the mean or second quarter. Dote lines represents maximum and minimum values. We can see that the alignment score, which is the best score, the threading score and the modeller score have a high standard deviation. Modeller and solvent accessibility scores have a lot of extreme values represented as small circles. Co-evolution score has a low standard deviation but many templates have a score of 0 making a huge disequilibrium. Even if the mean is lower, we can mention that weighted combined scores have a lower standard deviation compared to the sum score which shows that the weights increased the efficiency of the program.

3.1.3 Analysis of ranking of family, superfamily and fold benchmarks

When running the benchmarking program, it generates a table presenting results for the Top N (top 5 to top 350) templates. The ranking was done according to the simple sum score (SS) and with the weighted combined score (WS) for comparison. With SS the program finds 15.8% of benchmarks in the top 10 (2 "Superfamily" and 1 "Fold") when WS needs to wait until top 15 to find only 5% of benchmarks (1 "Fold"). However, all benchmarks are encountered at top 250 thanks to machine learning, whereas simple scoring still lacks a "Fold" even at top 350. Knowing that there are in total only 1 "Family", 6 "Superfamily" and 13 "Fold" benchmarks, the results are decent. Moreover, the targets are particularly hard to find because of the very weak identity percentage between the queries and the benchmarks revealed by a Needlman & Wunsch global alignment, for example "UBQ" and its Fold benchmark "protg" have 8.93% identity only, "Agglutinin" and "intb" have 2.33% identity. Same between "Lipoprotein_4" and its Superfamily benchmark "mofe" with 8.46% identity. It is interesting to notice that the "Family" type of benchmark should be the easiest target to find, however it is not the case here. The reason is that the only "Family" benchmark available is for PCNA: DNA PPF. A simple Needlman & Wunsch alignment showed no more than 15% identity between PCNA and DNA PPF, even though they are classified by SCOP database as belonging to the same "Family". This explains why we have difficulties encountering it also.

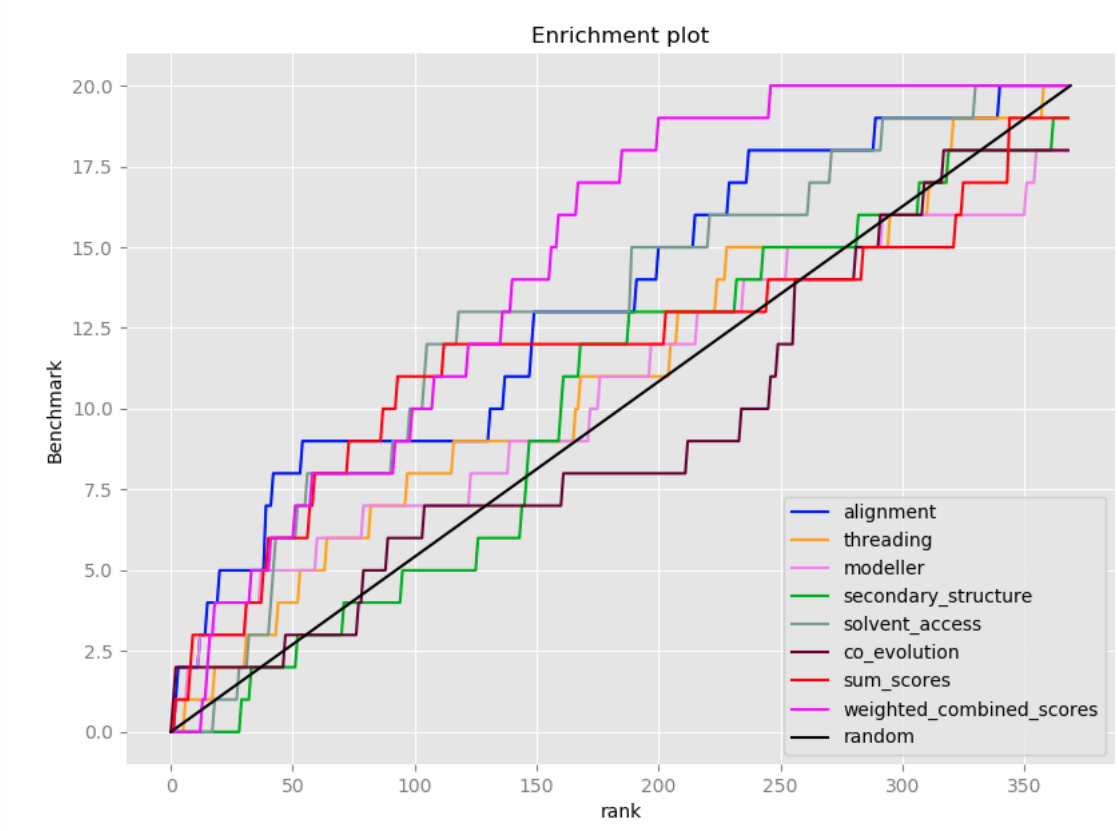

Figure 3: Enrichment plot : Cumulative sum of benchmarks encountered along the ranking (from rank 1 to rank 397).

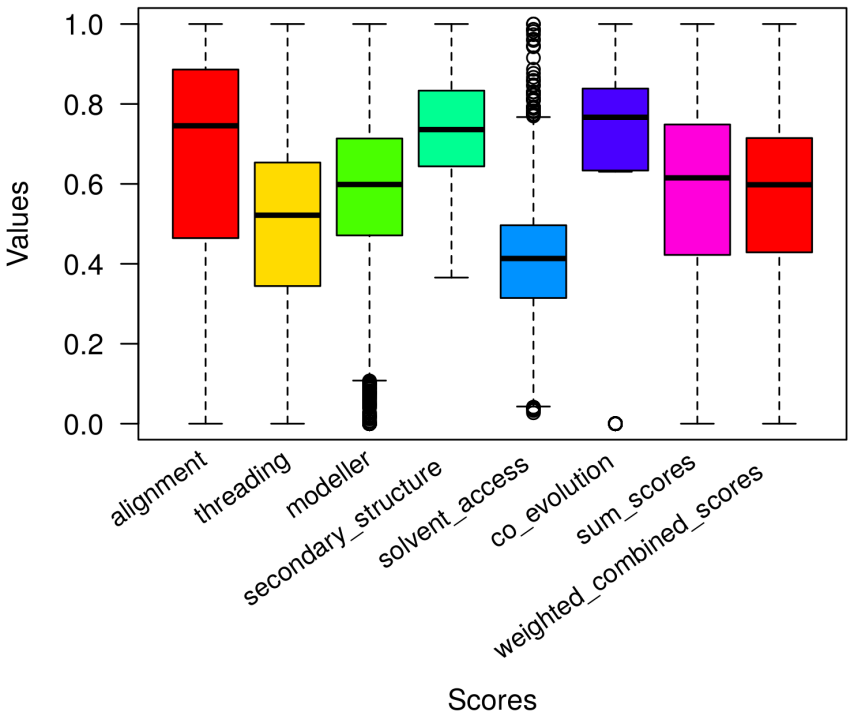

Figure 4: Distribution of values for each scores for the 20 benchmarks.

3.1.4 Example with the query TBCA

As an example, we decided to take a closer look at the query "TBCA". TBCA is a tubulin binding protein only composed of alpha-helices. Its SCOP ID is a.7.5.1. The associated benchmark is "BAG" and is in the same "Fold" category. This template also has an all alpha helices structure and its SCOP ID is a.7.7.1. When aligned, the identity between the two sequences is only 9.82 %. A top 20 table was generated for this query (See **Table 1**). "BAG" was Top13 which is the best rank for this data with our program. In the top 20, 65 % of the templates were all alpha proteins, 20 % were alpha and beta (a + b) and 15 % were all beta. There is a clear enrichment for proteins containing alpha helices. Maybe this category of proteins is easier to identify explaining the high enrichment.

To go further with the analysis of the results for "TBCA" query, we decided to superpose TBCA predicted structure with BAG template. (See **Figure 5**). As we can see, Fold-U predicts 3 alpha helices corresponding well to the real structure of the template BAG although one of the helices is longer. This cannot be explained by the length of the sequence of the query which is 102 amino acids while BAG is 114 amino acids long.

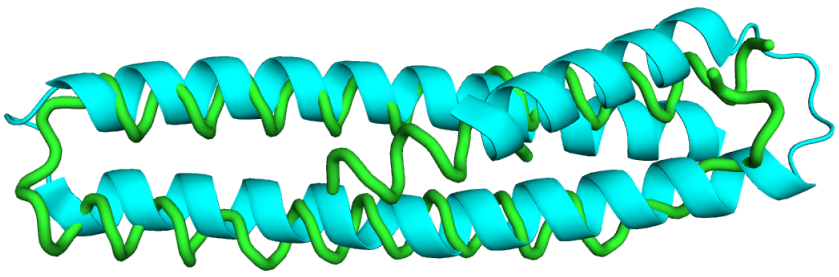

Figure 5: Alignment in PyMol between **TBCA** (in blue, pdb code: 1qsda) and its "Fold" benchmark **BAG** found as the top 13 model (in green) by Fold-U.

Top	Template name	SCOP id	First class SCOP id	Percent in top 20
Top 1	Fapy_DNA_glyco	a.156.1.2	**a**: all alpha	65%
Top 2	Band_41_M	a.11.2.1	**d**: alpha and beta (a+b)	20%
Top 3	hemopexin	b.66.1.1	**b**: all beta	15%
Top 4	bv	b.121.4.7	**Total**	100%
Top 5	START	d.129.3.2		
Top 6	RGS	a.91.1.1		
Top 7	utg	a.101.1.1		
Top 8	gag_p17	a.61.1.1		
Top 9	Syntaxin	a.47.2.1		
Top 10	parv	a.39.1.4		
Top 11	capsid_n	a.73.1.1		
Top 12	cytc	a.3.1.1		
Top 13	BAG	a.7.7.1		
Top 14	NusB	a.79.1.1		
Top 15	intb	b.42.1.1		
Top 16	ghf22	d.2.1.2		
Top 17	Sua5_yciO_yrdC	d.115.1.1		
Top 18	PTS_EIIA_2	d.112.1.1		
Top 19	Ribosomal_S7	a.75.1.1		
Top 20	Ald_Xan_dh_1	a.56.1.1		
Query	**TBCA**	**a.7.5.1**		

Table 1: Top 20 results of the Fold-U program for TBCA. The benchmark associated (a Fold) is BAG.

3.2 Comparing our results with the other team for the unknown sequences

Eleven FASTA sequences with unknown structures were given by the students from the Bio Santé Master. Among them, recN fasta sequence contained 5 "X" unknown residues which were removed in order to be able to launch the program. The consequence is only few additional gaps in alignments. The mfd sequence could however not be run with our program because of its length (1178 residues). Using uniref90 for multiple sequence alignments, the alignments were much too long and unusable by CCMPRED (co-evolution score). According to poor results of CCMPred scoring discussed earlier, Fold-U was thus run without the CCMPred scoring function, for 10/11 sequences and the results compared with other teams. For some sequences like G1G14-3311(BT9727_3234) by Mathieu Legras or hmp (BT9727_1331) by SELVAM Marie-Jeanne Lucie, COX1 template was found for at least 3 out of 4 teams. For other sequences, all team had different template results.

4 Discussion & Conclusion

In this project, we developed a program able to generate several 3D models of proteins for a given FASTA sequence, based on protein threading and additional scoring methods which take into account more protein sequence- and structure-based features.

The results are interesting and decent even though of course far from being comparable to other methods implemented in CASP for instance, knowing the difficulty of the targets to find regarding the lack of identity between queries and benchmark sequences.
It was shown in this project that some scoring techniques such as solvent accessibility or co-evolution were either not enough informative, too specific or could be tweaked according to new parameters.

Further improvements could be to improve the scoring functions by changing the algorithms of solvent accessibility for example, or else find new scoring methods that were shown to be successful in the literature. Last but not least, machine learning could be greatly improved. Because of lack of time, only the benchmarking dataset of 20 proteins was used. Machine learning requires a lot of data to perform best. With this much data, other machine learning could even be tested such as Support Vector Machine or Random Forest, maybe even deep learning if enough data is available. Also, instead of learning the whole data naively, it could be interesting to classify and learn the training data according to the classes of SCOP ids ("a", "b", "g", etc.) to better learn the specificity of each class of protein.

References

[1] Ken A Dill and Justin L Maccallum. The Protein-Folding Problem, 50 Years On. Technical report.

[2] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (CASP)—Round XII. *Proteins: Structure, Function and Bioinformatics*, 2018.

[3] Nika Abdollahi, Alexandre Albani, Eric Anthony, Agnes Baud, Mélissa Cardon, Robert Clerc, Dariusz Czernecki, Romain Conte, Laurent David, Agathe Delaune, Samia Djerroud, Pauline Fourgoux, Nadège Guiglielmoni, Jeanne Laurentie, Nathalie Lehmann, Camille Lochard, Rémi Montagne, Vasiliki Myrodia, Vaitea Opuu, Elise Parey, Lélia Polit, Sylvain Privé, Chloé Quignot, Maria Ruiz-Cuevas, Mariam Sissoko, Nicolas Sompairac, Audrey Vallerix, Violaine Verrecchia, Marc Delarue, Raphael Guérois, Yann Ponty, Sophie Sacquin-Mora, Alessandra Carbone, Christine Froidevaux, Stéphane Le Crom, Olivier Lespinet, Martin Weigt, Samer Abboud, Juliana Bernardes, Guillaume Bouvier, Chloé Dequeker, Arnaud Ferré, Patrick Fuchs, Gaëlle Lelandais, Pierre Poulain, Hugues Richard, Hugo Schweke, Elodie Laine, and Anne Lopes. Meet-U: Educating through research immersion. *PLOS Computational Biology*, 14(3):e1005992, 3 2018.

[4] Renxiang Yan, Dong Xu, Jianyi Yang, Sara Walker, and Yang Zhang. A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Scientific Reports*, 3(1):2619, 12 2013.

[5] Debora S Marks, Thomas A Hopf, and Chris Sander. Protein structure prediction from sequence variation. *Nature Biotechnology*, 30(11):1072–1080, 11 2012.

[6] Saulo Henrique Pires De Oliveira, Jiye Shi, and Charlotte M. Deane. Comparing co-evolution methods and their application to template-free protein structure prediction. *Bioinformatics*, 33(3):373–381, 2017.

[7] Natalie L. Dawson, Tony E. Lewis, Sayoni Das, Jonathan G. Lees, David Lee, Paul Ashford, Christine A. Orengo, and Ian Sillitoe. CATH: An expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research*, 2017.

[8] Min-Yi Shen and Andrej Sali. Statistical potential for assessment and prediction of protein structures. *Protein science : a publication of the Protein Society*, 15(11):2507–24, 11 2006.

[9] Andrej Šali and Tom L. Blundell. Comparative Protein Modelling by Satisfaction of Spatial Restraints. *Journal of Molecular Biology*, 234(3):779–815, 12 1993.

[10] Stefan Seemayer, Markus Gruber, and Johannes Söding. CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130, 11 2014.

[11] Wouter G. Touw, Coos Baakman, Jon Black, Tim A.H. Te Beek, E. Krieger, Robbie P. Joosten, and Gert Vriend. A series of PDB-related databanks for everyday needs. *Nucleic Acids Research*, 2015.

[12] Wolfgang Kabsch and Christian Sander. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Technical report.

[13] Hsin-Nan Lin, Ting-Yi Sung, Shinn-Ying Ho, and Wen-Lian Hsu. Improving protein secondary structure prediction based on short subsequences with local structure similarity. *BMC genomics*, 11 Suppl 4(Suppl 4):S4, 12 2010.

[14] Peter Y. Chou and Gerald D. Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, 1 1974.

[15] Gianluca Pollastri, Pierre Baldi, Pietro Fariselli, and Rita Casadio. Prediction of Coordination Numberand Relative Solvent Accessibility in Proteins. Technical report.

[16] Huiling Chen and Huan Xiang Zhou. Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. *Nucleic Acids Research*, 2005.

[17] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 1992.

[18] Charles E. McCulloch. Generalized Linear Models. *Journal of the American Statistical Association*, 2000.

[19] Package 'boot'. Technical report, 2017.

[20] Yassine Ghouzam, Guillaume Postic, Pierre Edouard Guerin, Alexandre G. De Brevern, and Jean Christophe Gelly. ORION: A web server for protein fold recognition and structure prediction using evolutionary hybrid profiles. *Scientific Reports*, 2016.

[21] David T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 1999.