Faculté des Sciences et Ingénierie

Bioinformatique et Modélisation

Meet-U Project

Prediction of Protein Structure

Students:

Bénédicte Colnet

Gabriela Lobinska

Irène Mauricette Mendy

Yasser Mohseni Behbahani

Amandine Sandri

Supervisors:

Juliana Bernardes

Elodie Laine

January 2, 2019

Introduction

The goal of this project is to infer a 3D-structure of a protein from homology modeling, also known as comparative modeling. Comparative modeling relies on finding a homologous protein (a "template") whose structure is thought to be similar to the protein of interest (the "query"). This template-modeling approach is one of the main methods for protein structure prediction [1, 2]. Since protein structure is intimately related to the function of a protein, which is conserved along evolutionary time, we can expect high similarities in structure even though the primary sequences are diverged. The known structure can be then used to build a first model, which is then refined to match the query protein with its physicochemical specificities. The chosen template is therefore paramount for determining the structure, and often different models can be obtained with different templates. Hence, it is important to identify the best possible template to obtain a good three-dimensional model for a query amino acid sequence. Our project is focused on such identification and scoring.

In this methodology, a profile-profile comparison, instead of sequence-to-profile comparison, is used to mine databases and find an appropriate structural template [3]. Protein profiles, were introduced in 1987 [4], and are statistical tools synthesizing information about a given protein family. Each site is assigned a probability of containing a given amino acid.

Therefore the main challenges faced in this report are:

- **Part A** Gathering of homologous sequences
- **Part B** Building of profiles from multiple sequence alignment.
- **Part C** Comparison of two profiles and to assessment of their similarity with a score

1 Methods

With an incremental approach, we first decided to build a very simple pipeline serving as a minimum viable product. Then, we implemented additional features in order to improve our prediction for the best protein template, analyzing our results at each step. We present hereafter those improvements and how they affected the output of our tool. The final pipeline is presented in figure 1.

1.1 Part A : PSI-BLAST and database

To create a profile for a given query we need to find its homologous domains in protein sequences and create a family for it. In this project we use PSI-Blast algorithm (from BLAST+ software) with different e-values and various databases to find these homologous sequences. We use local databases in order to make the processes faster and more reliable and operate PSI-Blast in its multi-thread mode. Most of our experiments are done on UniProt databases, specially UniRef50.

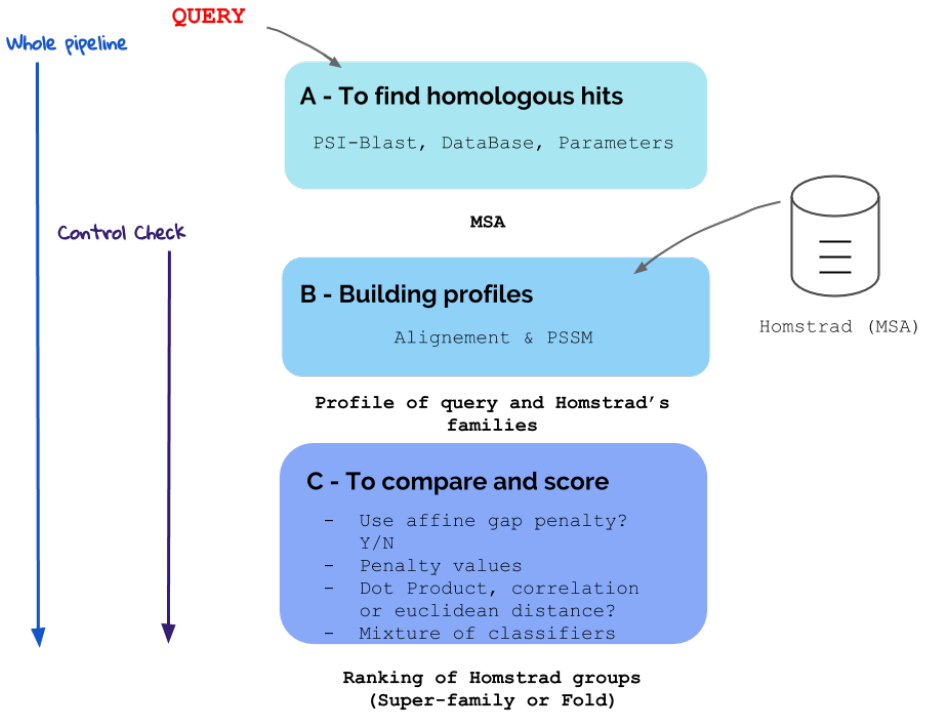


Figure 1: Whole pipeline's schematic.

1.2 Part B : Building profiles

Before building profiles, the MUSCLE algorithm is used to perform multiple sequence alignment. Two common ways to build a profile for a family of domain sequences is using Hidden Markov Model (HMM) and Position-Specific Scoring Matrix (PSSM). Due to its simplicity and performance we choose PSSM to build family profiles in both frequency and log-odd ratio formats. Thus PSSM profiles are built for families of queries and family of HOMSTRAD dataset.

1.3 Profile-profile comparison methods

The profile comparisons methods implemented are inspired from existing methods reported in different papers [5] [6] [7].

1.3.1 Dot Product

Dot product computes the product using frequencies. It is the scalar product of columns of two profiles. In other words, the higher the frequency of a same amino acid is, the higher the product is. Quantitatively, we combine column i of profile (A) and column j of profile (B) with the following formula :

$$\mathbf{D} = \sum_{\mathbf{i=1}}^{\mathbf{20}} \mathbf{A_i B_i}$$

Where A_i and B_i are the values of amino acid i in columns A and B of the profiles, respectively.

1.3.2 Pearson correlation

With the same notation we apply:

$$r_{AB} = \frac{\sum_{i=1}^{20} \left(A_i - \bar{A}\right)\left(B_i - \bar{B}\right)}{\sum_{i=1}^{20} \left(A_i - \bar{A}\right) \sum_{i=1}^{20} \left(B_i - \bar{B}\right)}$$

1.3.3 Euclidean distance

And also:

$$d(A, B) = \frac{\sum_{n=1}^{20}(A_i - B_i)}{20}$$

The distance scores range from 0, for identical columns, to $\frac{\sqrt{2S^2}}{20}$ where S is the sum of the values in each column. This largest possible distance occurs when only one different amino acid appears in each column.

1.3.4 Parameters

he parameters used for the gap penalty opening and extension are presented Table 1, these parameters were derived from bibliography relative to the subject [7] and had to be changed empirically to find better results on the benchmark.

Parameter	Value	Bibliography value [7]
Gap opening penalty Dot Product	12	0.07
Gap extension penalty Dot Product	1	0.005
Zero Shift Dot Product	-0.03	-0.05
Gap opening penalty Correlation	5	1.39
Gap extension penalty Correlation	0.5	0.07
Zero Shift Correlation	-0.2	-0.21

Table 1: Optimized parameters for scoring functions

1.4 Affine gap penalty principle

In the Smith-Waterman implementation, the gap penalty is added for each deleted/inserted amino acid. Therefore, an alignment with a deletion of several amino acids in length will have a markedly lower alignment score than an alignment with a deletion of one amino acid, despite their biological probability being similar. An extension of the Smith-Waterman algorithm allowing taking into account affine gap penalties (different gap penalties for gap opening and gap extension) was proposed by [8]. However, this method presented a huge disadvantage in terms of memory and computational speed. Bibliographical research showed that [9] suggested an algorithm which reduced the computational speed and memory requirements to be similar to the ones of the basic Smith-Waterman alignment. Therefore, we were able to implement a local alignment with gap opening and extension, without sacrificing in computational speed and memory.

1.5 Downstream programs benchmark

In order to evaluate the output files generated from our profile-profile comparison, we selected a downstream program allowing to test our results. The selection of the downstream team was led by considering six main criteria presented in the table 2.

Criteria	Details
Installation Process	It should be easy to make the program ready for execution (number of packages to be installed, size of the folder etc.).
Strategy	Computed scores to infer the highest similarity between families and query.
Program evaluation	Is the performance of program evaluated, and, if so, how?
Language	Is there any use of coding language allowing to speed up or being more suitable to run the downstream process?
Computer Processing Speed	Effectiveness and timeliness of the computer program.
Repository Organization	Ease of understanding and navigating into database documentary structure.

Table 2: Table of criteria to benchmark downstream programs.

After sorting all downstream teams, we chose to evaluate our results with team 5 and 8 because we were interested in their scoring method to rank all the models and the ease of installation of their package.

2 Experiments and Results

All the results presented were ran over the 21 sequences proposed in the benchmark list. The results are presented either as an enrichment curve with one single good result - considering or not the specificity of the fold or superfamily - or as a dispersion plot showing the rank for each query.

2.1 Control check of profile construction

The aim of this part is to check whether the profile generation (i.e the first degree of freedom from PSI-BLAST to profile construction) is accurate. Therefore, we implement a control check, meaning we run the profile-profile comparison part on the PSSM profiles generated from scratch with different e-values and on the PSSM profiles generated from the provided multiple-sequence alignment of each query. By doing this we bypass PSI-BLAST and MUSCLE modules and omit the influences of their parameters (such as e-value or databases) in our study of comparison methods. The enrichment curve is presented on figure 2. This figure shows that the enrichment is similar between the different PSSM profiles considered, all other parameters remain equal.
In addition to a single score of enrichment, a comparison of the dispersion of scores and specificity was computed for each query as an indicator of the specificity induced by the PSSM profiles on the final scores. Such dispersions are presented on figure 3. This figure 3 highlights the highest dispersion of the scores when using the Control Check PSSM profiles, and therefore highlights a huge difference in the PSSM profiles for each

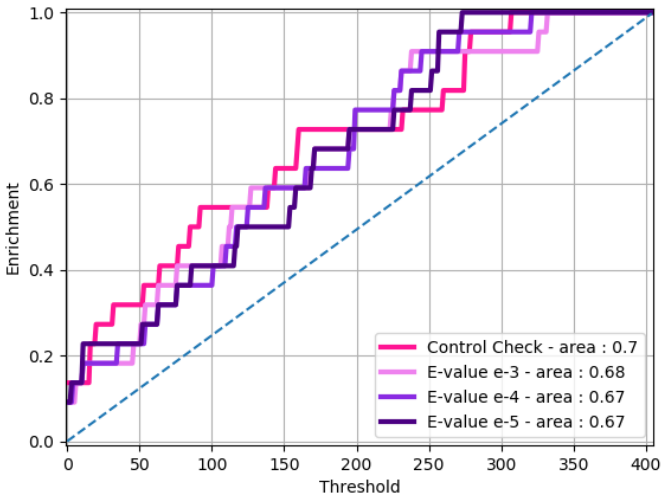


Figure 2: Enrichment curves for 4 different kinds of PSSM profiles using a profile-profile comparison with affine gap extension and dot product.

query, even if the output ranking of queries along the HOMSTRAD dataset shows the same enrichment.

2.2 Affine Gap penalties

We implemented the affine gap penalty approach for the Smith-Waterman algorithm, and beyond the biological explanation, this part checks whether this implementation brings pertinent information for your ranking. Therefore, we ran experiment with and without affine gap penalties on the benchmark[1], and observed the enrichment curve (figure 4 (a)). As expected, the enrichment curve is better with affine gap penalty. Moreover, including the affine gap penalty substantially increases the spread in scores. This is promising for scoring proteins and gives an estimation of the certainty and confidence we have on the score. The figure 4 (b) highlights the spread.
With the other scoring functions we defined above 1.2, we computed local alignment with Smith-Waterman algorithm with affine gap penalty, and varied the scoring of two columns extracted from profiles. We plotted the enrichment curves in order to compare dot product and correlation, results are presented in figure 5. The results are complicated to compare considering only the area under the curve, even if we notice that correlation comparison has better performances on the low threshold for HOMSTRAD dataset. In order to distinguish differences, the enrichment curves are then plotted separately for superfamily and fold results. These plots are shown in figure 5. We observe that the correlation comparison has better results when a superfamily is expected. A protein *superfamily* is the largest grouping (clade) of proteins for which common ancestry can be inferred, and superfamilies typically contain several protein families which show sequence similarity within each family. But proteins might also have considerable structural similarities even when no evolutionary relationship of their sequences can be detected. This

[1]This procedure was repeated for several e-values with no significant effect on the results, therefore only the e-value $1e-3$ is presented in this report.

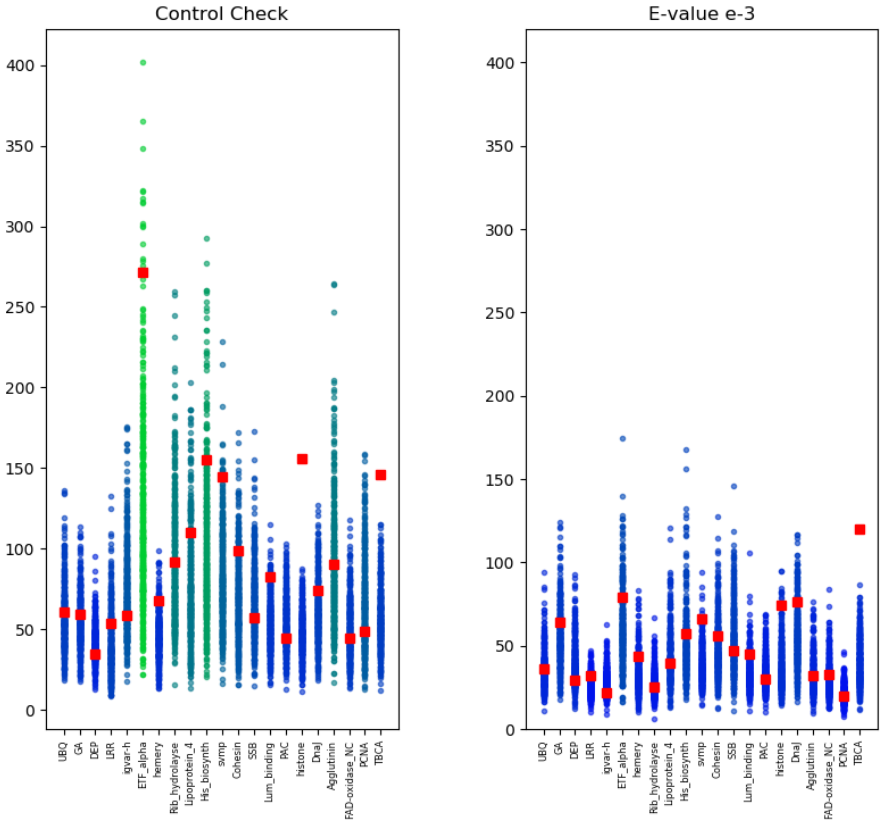
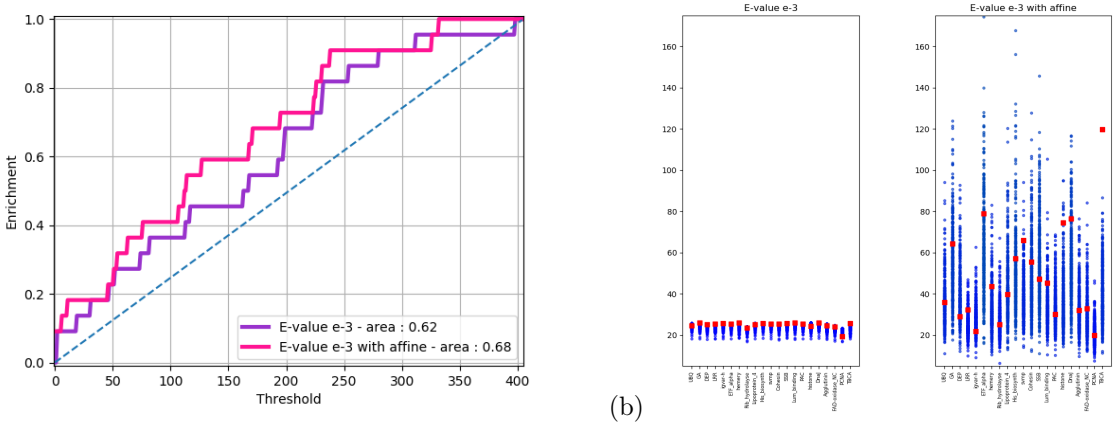
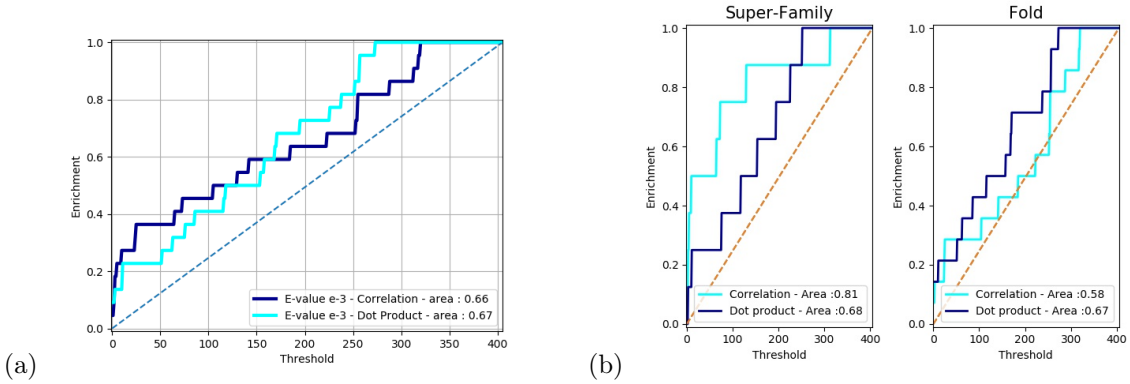


Figure 3: Dispersion of scores using affine gap penalty and dot product on different PSSM profiles (From the left to the right: Control check and e-value $1e-3$). All the dispersion for our PSSM profiles construction - whatever the e-value- are similar and show the same difference with the Control Check's dispersion.



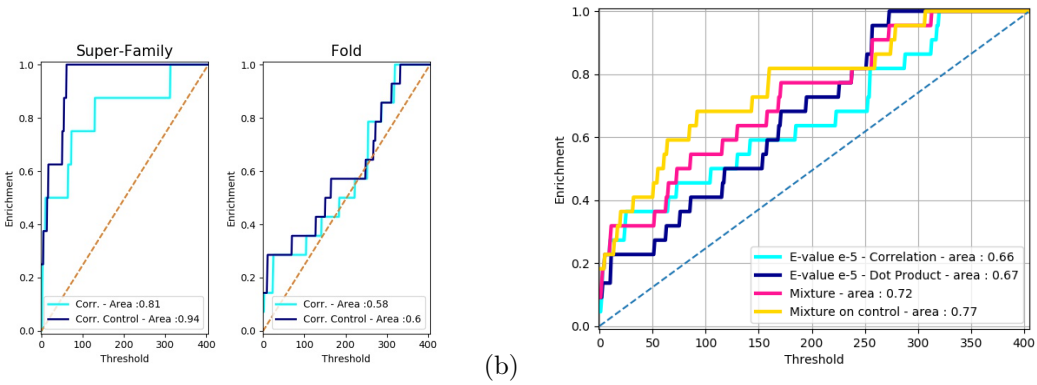
(a) (b)

Figure 4: (a) The enrichment curves for two tests: with and without affine gap penalties, all other parameters remaining equal. (b) Visualization of the distribution of scores over HOMSTRAD for the algorithm with and without affine gap penalties.



(a) (b)

Figure 5: (a) The enrichment curves for two tests with correlation comparison and with dot product comparison, and all other parameters remaining equal such as affine gap penalties and e-value equals $1e-5$. (b) Same representation but according to the type of the results expected.



(a) (b)

Figure 6: (a) Enrichment curves with correlation and dot product comparison methods, all other parameters remaining equal (e-value equals $1e-5$). (b) A mixture of the two classifiers.

property is often referred to as the proteins sharing only a *fold* [10]. Therefore, it seems logical that the correlation that is a more powerful tool to distinguish conserved positions on profiles is more performing on superfamily recognition as it is linked to the sequence homology. The dot product cares more about identical amino acids found at the same position, than similar distribution of frequencies of having amino acids, as having the same amino acid provides a highest score than having a distribution of frequency over the position. Furthermore, this result on correlation should warn us on the importance of the PSSM profiles construction to have the best signal possible. Therefore we ran the correlation comparison on the control check, and expected a better results on superfamily as we observed that the provided MSA for HOMSTRAD were slightly better than ours. Figure 6 confirms this trend with a very good performance of correlation comparison on PSSM profiles when superfamily results is expected.

Mixture of classifiers Since we observed and explained why the superfamily is better predicted than the fold using the correlation comparison, we can propose an original approach using a mixture of the two classification (the idea - even if it has nothing to do

in practice! - comes from the Boosting [11]). Then we proposed, from two methods having the same performance to build a mixture on the expected result: Fold or superfamily. As expected the mixture has the best performances as showed on figure 6 (right). We also notice that, we can estimate kind of a maximum performance using control check. It reveals the method to be sensitive to the used MSA method.

Going further on Fold recognition Protein folds are generally more conserved than their primary structures. By definition, similar folds translate in similar secondary structures. This means that secondary structure should be conserved within a fold even if the amino acid sequences have diverged. Secondary structure can be predicted with an accuracy higher than 70% [12]. Therefore, using secondary structure as a major component of the ranking score when the output is a fold, would improve the current pipeline. Several authors have found that including the secondary structure in the score calculations improves fold recognition [13]. Another improvement would be to use such a score as a mixture with one more suited to homology detection, such as the correlation comparison method.

2.3 Downstream Benchmark

For the downstream team we initially chose to work with Team 5 because they have an interesting strategy for us to evaluate our output which is the combination of C-beta score used in addition to the C-alpha score allowing to better evaluate the threading score. Unfortunately, after installing and running its package we realized that the secondary structures are mandatory in its implementation - which are not part of our calculations in pipeline. Therefore, we chose to work with the Team 8. The Team 8 uses our profile-profile alignments as well as the Discrete Optimized Protein Energy (DOPE) and the Hydrophobe-Polaire (H-P) scores in their ranking. These scores are then computed together using *Machine Learning* approach on an extended HOMSTRAD dataset. Prediction of HOMSTRAD family from DOPE and H-P scores using machine learning techniques can be considered as an original approach.
Therefore the idea is to compare our results on the benchmark queries with and without the downstream part, the results are presented[2] on figure 7. Our pipeline was used with the dot product and affine gap extension and e-value of $1e-3$ not to over-rank the superfamilies (and therefore being more *neutral*). We realized that the combined results are somehow better than ours when the threshold is set to the 50 first ranked proteins (over 405). But in general, the combination of our two pipelines give bad performances. We then analyze the superfamily and fold effect in this enrichment curvre. The enrichments depending on superfamily and fold are represented on figure 8. We observe that the error in the combination of the two pipelines seems to come from the superfamilies that are badly ranked, when folds does not show such a bad ranking. An hypothesis can be related to the mixture of three scores in the downstream part that could have been adapted from the Orion's one (maybe overfitted) and that have less meaning when it comes to use our score.

[2]The threshold is set to the 300 first ranked proteins (over 405) because the output of the downstream pipeline stops to that threshold. Therefore the representations and area are calculated based on this threshold.

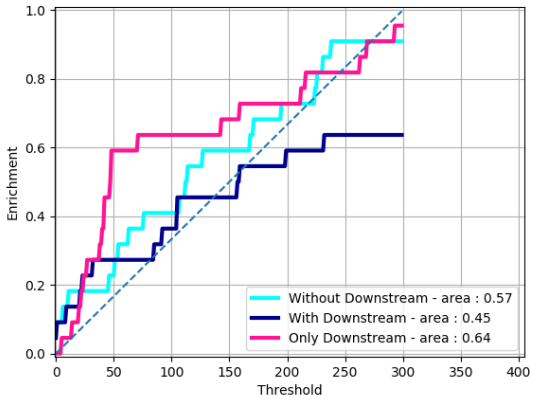


Figure 7: Results on the benchmark using or not the downstream part on our output generated using the dot product and affine gap extension and e-value of $1e-3$, and also in pink the results of the Downstream part using Orion [3] output.

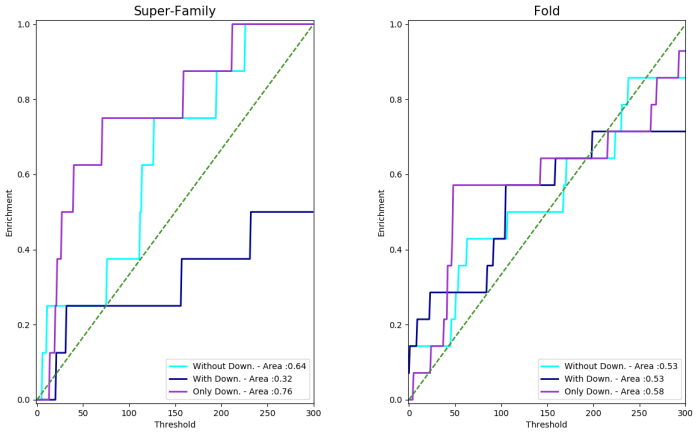


Figure 8: Same legend as figure 7, but distinguishing superfamily and fold.

2.4 Mysterious queries

Several students from Master 2 of Biology Health have provided mysterious sequences for which they need to know the 3D structure. We therefore run our upstream program and the one from downstream (Team 8) that is previously selected. Results are provided in table 3 .

Mysterious query ID	Best hit template
trQ6HKV8 (BT9727_1481)	BRCT
trQ6HLB4 (BT9727_1323)	END
feuA (BT9727_3483)	neur_chan_memb
G1G14-4353	pot
recN (BT9727_3916)	COX1
G1G14-3311 (BT9727_3234)	PCI1
scdA (BT9727_1971)	PABP
hmp (BT9727_1331)	zf-CCHC
mfd (BT9727_0048)	COX1
celB (BT9727_4888)	Ribosomal_L12
arcD (BT9727_0754)	hormone5

Table 3: Table of best HOMSTRAD sequence hit for each mysterious query.

Conclusion

In this project we designed and implemented a package to predict the 3D structure of proteins.

Analysis With respect to the two degrees of freedom we developed an incremental approach in which we conducted different experiments and new features gradually added to the main pipeline. All these features were tested with various parameters until we reached optimum conditions for e.g concerning e-values, database, and more specifically for Pearson correlation and dot product methods. Beyond enrichment curves, our analysis focuses on the dispersion of score to assess the quality of the method for each query. We also analyzed the results for superfamily and fold which showed us that the results are highly dependent on them. Last but not least, we found a way to assess the influence of the profile building - *part of the so called first degree of freedom* - using a Control Check method.

To go forward Even though in these experiments optimization of parameters is performed, improvements are still required:

1. The efficiency of profile construction is assessed using a Control Check procedure. It reveals the profiles to be of good quality, but improvements are still required. Furthermore we observed that the correlation comparison is highly sensitive to the profiles construction.

2. Our pipeline underscores the intrinsic differences when ranking a superfamily or a fold, which leads us to the idea of a mixture to be the most efficient classifier. A first test of this algorithm was performed on the benchmark, and it appears to be promising.

3. For a mixture to be even more promising, the fold detection has to be improved, and it seems that information deduced from the sequence has to be used such as secondary structure prediction.

References

[1] Moult J., Pedersen J. T., Judson R., and Fidelis K. A large-scale experiment to assess protein structure prediction methods. *Proteins*, 23, 1995.

[2] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp) — round x. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):1–6, 2014.

[3] Ghouzam Y., Postic G., Guerin PE., de Brevern AG., and Gelly JC. Orion: a web server for protein fold recognition and structure prediction using evolutionary hybrid profiles. *Sci Rep.*, 2016.

[4] Gribskov, McLachlan, and Eisenberg. Profile analysis: detection of distantly related proteins. *Proc Natl Acad Sci U S A.*, 13:4355–8, 1987.

[5] Chuan Wang, Ren-Xiang Yan, Xiao-Feng Wang, Jing-Na Si, and Ziding Zhang. Comparison of linear gap penalties and profile-based variable gap penalties in profile-profile alignments. *Computational biology and chemistry*, 35:308–18, 10 2011.

[6] Shmuel Pietrokovski. Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic acids research*, 24 19:3836–45, 1996.

[7] Guoli Wang and Roland Dunbrack. Scoring profile-to-profile sequence alignments. *Protein science : a publication of the Protein Society*, 13:1612–26, 07 2004.

[8] O Gotoh. An improved algorithm for matching biological sequences. *J Mol Biol*, 162:705–708, 1982.

[9] Gianvito Urgese and Giulia Paciello. Dynamic gap selector: A smith waterman sequence alignment algorithm with affine gap model optimisation. *Proceedings IWBBIO*, pages 1347–1358, 2014.

[10] Lindahl and Elofsson. Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.*, 3:613–25, 2000.

[11] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[12] B Rost and C Sander. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proceedings of the National Academy of Sciences*, 90(16):7558–7562, 1993.

[13] Hargbo and Elofsson. Hidden markov models that use predicted secondary structures for fold recognition. *Proteins*, 1:68–76, 1999.