Domain prediction from 3D structure by using CNN network

Master 2 Bio-informatique

Ragousandirane RADJASANDIRANE

2021-2022

Tutor : Jean-Christophe Gelly

Summary

I. Introduction

Proteins are building blocks of biology and understand them are a very challenging goal. Getting a better knowledge about them can leads to new treatment for some diseases or cancer. A way to have a better understanding of proteins is to cut them into pieces in order to analyse small part of it instead of the full protein, which can be a hard task if the protein is too large. The protein is naturally divided into domains which are the sub unit of the protein. These domains can also be divided into Protein Units (PU), a smaller unit of the protein and an intermediate level between secondary structures and domains. 3D structure of domains and PU can be enough to analyse certain properties of the protein.

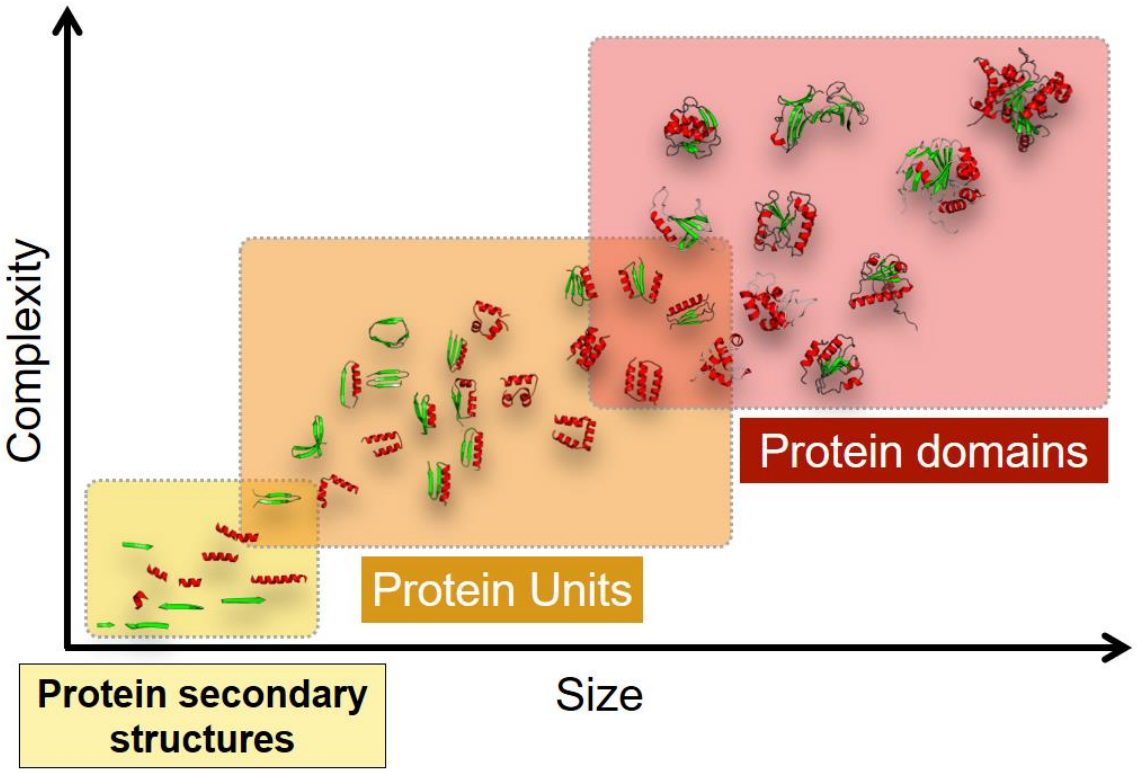


Figure 1: Graphical representation of different structure scale of a protein.
The complexity increase linearly with the structure size. Secondary structures are the first units of a protein structure linked to Domain by Protein units.

Some tools have been proposed to study the protein structure through PU and domain. Protein Peeling (PP) is an iterative algorithm used to determine all PUs for a given protein (J.-C. Gelly et al. 2006; Jean-Christophe Gelly, de Brevern, and Hazout 2006). The PP method cut the protein into small pieces and assesses the quality of these by using several criterions and by quantifying the subunits independence in terms of contact. Another tool named SWORD (Postic et al. 2017) has been developed which uses the PP algorithm to determine all PUs of a protein and then find its domains by merging generated PUs by using a bottom-up approach.

Here, we present KOPIS a deep learning program using a Mask R-CNN network (He et al. 2020) to predict PU and domain of a given protein with good accuracy and faster than conventional methods. KOPIS predictions and other method proposals are compared to expert annotation of domains of benchmarked proteins. For the following, we'll mainly focus on domains prediction rather than the PUs one.

II. Materials and methods

1. Benchmark data sets

a. Jones set

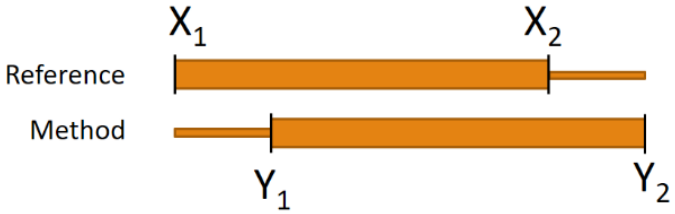
The first data set used to evaluate domain prediction of KOPIS is the Jones domain data set which contains 55 proteins with expert annotation by the authors of the structures (Jones et al. 1998).

b. Islam90 set

Islam90 data set contains 90 annotations of proteins that share under 30% identity to avoid overrepresented protein families (Islam, Luo, and Sternberg 1995).
In both datasets, proteins with discontinuous domains in their annotation are removed.

2. Prediction evaluation

Using the two benchmark data sets above, 6 different references methods, widely used in domain assignment, are compared to KOPIS predictions. We used PDP (Alexandrov and Shindyalov 2003), DomainParser (Guo et al. 2003) and DDomain (Zhou, Xue, and Zhou 2007) methods. Like KOPIS, these methods provide domain delimitation for proteins in Jones and Islam90 data set. A coverage is computed between the proposal domain versus the reference annotation. This coverage consists of count the correct position matched over the area occupied by both domains as showed in Figure 2. If the coverage between the reference domain and the method domain is greater than or equal to 85%, we consider the method domain boundaries to be correct.



$$Cov_{XY} = 1 - \frac{abs(Y_2 - X_2) + abs(Y_1 - X_1)}{abs(max(Y_2, X_2) - min(Y_1, X_1))} * 100 \quad (1)$$

Figure 2: Coverage between two domains. The coverage by counting the mismatch position over the delimitation Y_2-X_1 *and then obtaining the correct matching position by the inverse.*

For a given protein annotation, if the reference gives 3 domains for this protein, method proposal must also have a length of 3, otherwise it is an incorrect global boundary. KOPIS predictions and other methods boundaries are treated is the same way.

A Wilcoxon signed-rank test is used to determine if KOPIS predictions is different or not from the other methods boundaries comparing to references benchmarks with an α error of 0.05.

3. Data

Approximatively 12500 representative PDB proteins are used as input into the Mask RCNN network. These proteins have under 30 percent of identity between them. Each protein is run through SWORD tool to get PUs and domains delimitations. These delimitations are visualized in a contact probability map of the protein obtained by applying the formula (1) for each pairwise distance residue.

$$p_{i,j} = \frac{1}{1+\exp\left[\frac{d_{i,j} - d_0}{\Delta}\right]} \qquad (2)$$

From this contact probability map, delimitations of PU or domains can be drawn (Figure 3). The association of map and delimitations forms the input of the network. The goal of the network is to predict the position of these delimitations which will be translate into 3D delimitation and so, domains delimitations. All images are padded with value -1 to the max size (1024). This method allows to keep the original image without modification, which gives better results in predictions. The padded values are ignored with a mask layer in the network.

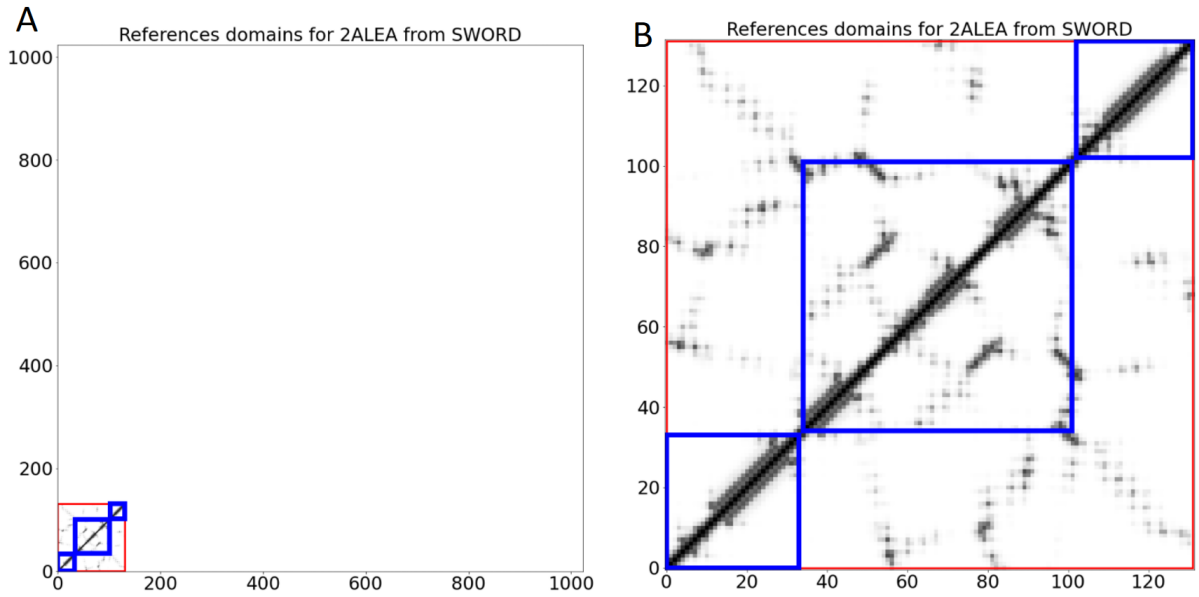


Figure 3: Contact probability map delimited with domains for 2ALEA PDB code. (A) Padded image with value -1. (B) The actual image of contact probability map. Domains from SWORD are delimited with square on the map. Red squares are the optimum solution which in this case takes all the protein size. Blue squares are the second best solution given by SWORD.

KOPIS will learn to predict the optimal solution but also the alternatives solution since there are often functionally or evolutionally correct domain that are not considered are optimal domains.

4. Mask RCNN

Mask RCNN (MRCNN) is a neural network composed of CNN initially implemented for object detection and object segmentation in an image.

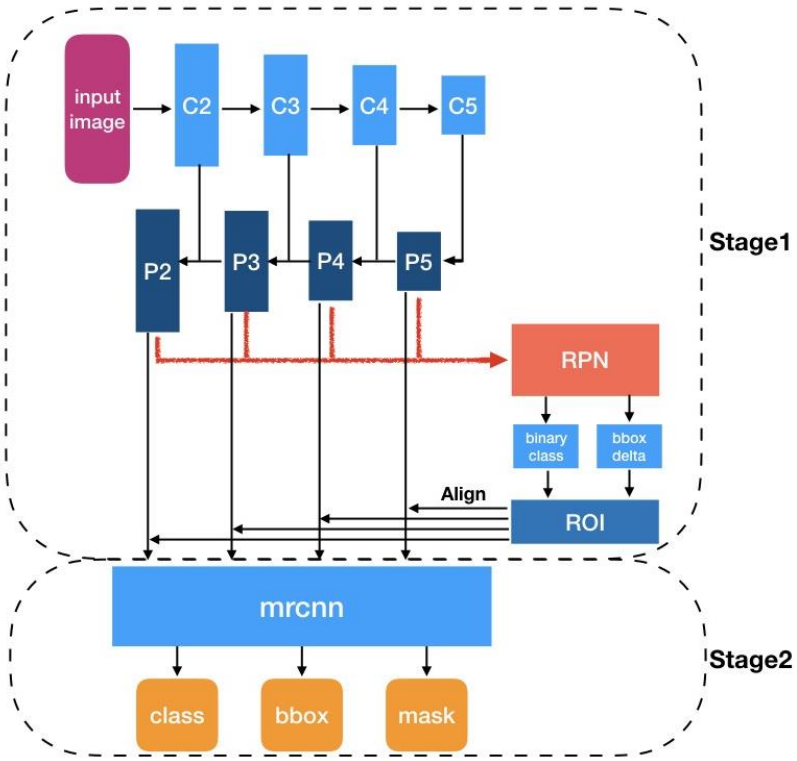


Figure 4 : Architecture of MRCNN

The architecture of MRCNN shown that it is composed of Convolutional backbone layer (C2 to C5) used to scan the whole image and detect all information on it. The backbone network can be either a Resnet50 of Resnet101 architecture for the backbone of MRCNN (For KOPIS, Resnet50 has been chosen for speed of the network). The backbone layers are followed by layers of FPN (Feature Map Network) which uses a top-down hierarchical structure to extract target objects from image. The RPN block give the Region Of Interest (ROI) boxes by scanning the image (Fig S1B). It represents where the object might be.

When MRCNN detects a ROI, it performs a RoIAlign on it in order to get the class label of the object (Fig S1C) and also a mask which will recover the surface of the object detected at pixel level using fully connected layers (Fig S1D). The RoIAlign consist of locating the initial position on the input image of the detected object by performing a specific pooling on the image. MRCNN has three output, a class object label, a bounding-box where the object might be and an object mask covering the whole object at pixel level.

There is some metrics that MRCNN gives as loss (Fig S2) for stage 1 (RPN train loss Fig S2, E and F) and for stage 2 (MRCNN train loss, Fig S2 B to D). There is the same loss for validation set. First loss is global loss for training set which represents all metric loss in one. The MRCNN_bbox (Fig S2B) loss concern the accuracy in the bounding box precision in stage 2, this loss is important in order to correctly detect the position of domains. The MRCNN_class loss (Fig S2C) concerns the classification of the detected object. In our case, it is not relevant since we have only one class (domain or PU at one time). MRCNN_mask loss (Fig S2D) concerns the accuracy of the mask produced in the bounding box at pixel level. It is not relevant for our case. RPN_bbox loss (Fig S2E) describe the capacity of the network to

correctly localize the position of the box in stage 1. It is different from MRCNN_bbox, which concerns stage 2, by using another method to relocate the ROI on the initial image. Indeed, in the stage 1 anchors are used to relocate the ROI on the image but on stage 2, it is the RoIAlign algorithm. Finally, the RPN_class loss represents the capacity of the network to detect either there is an object or not. This loss is also determinant to find the potential domain on the contact probability map.

5. Parameters

The training is on 50 epochs with a step per epoch of 1048 with a learning rate if 0.0001. All images are padded until reach the size of 1024. ResNet50 is used as backbone.

III. Results

1. Loss plots

MRCNN gives many loss metrics described above. These plots are showed with Tensorboard.

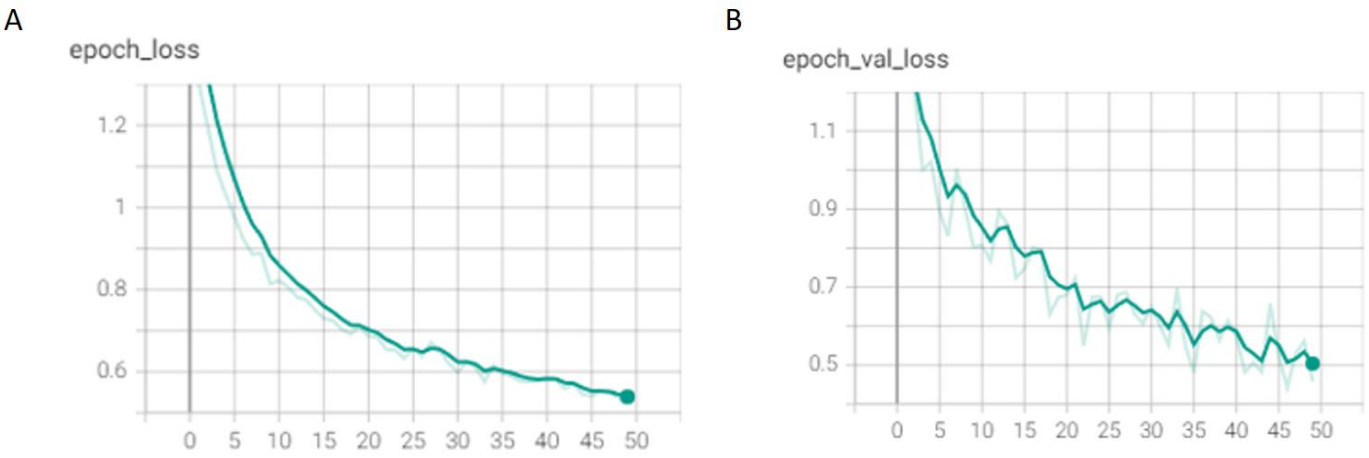


Figure 5: Train (A) and validation loss (B) on 50 epoch

The loss on both data set is gradually decreasing (Fig 5 A and B) and this is the case for all metrics (Fig S2) but there is some variation in the loss of RPN_bbox in validation set (Fig S2K). This might be due to a bit high learning rate.

2. Example of prediction

KOPIS predictions can be visualized by plotting true and predicted domain with matplotlib. Like in Fig 2B, the SWORD domain are colored in red and blue in Fig 6A.

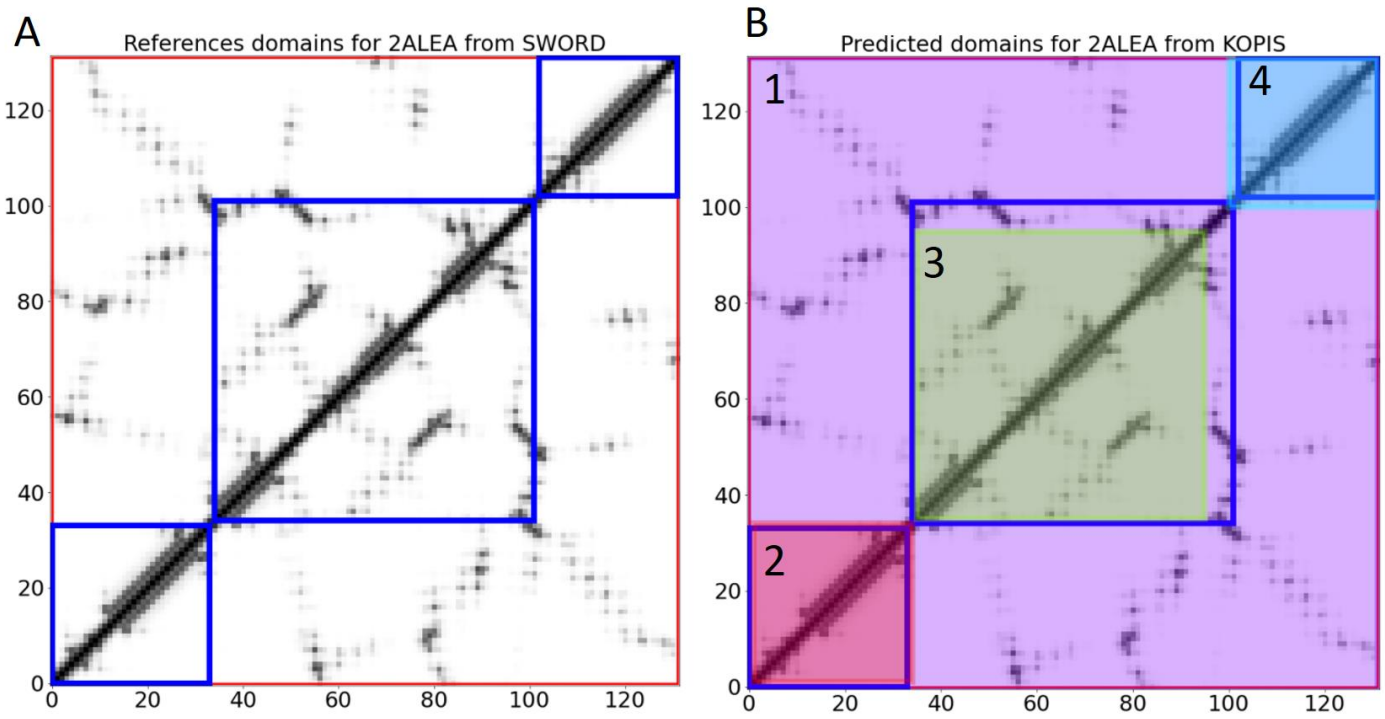


Figure 6: Visualization of KOPIS predictions for protein 2ALEA

In the predicted image (Fig 6B), the entire map is coloured with purple colour (domain 1). This means that KOPIS gives a solution that takes all the protein size, as in Fig 6A with the optimal solution from SWORD in red. This solution is an optimal solution with a probability score of 0.999 according to KOPIS output (Fig S6). KOPIS gives then the alternatives solutions which are the domains 2, 3 and 4 coloured with red, green and blue respectively in Fig 6B. These three domains constitute the alternative solution from KOPIS. It is consistent with the alternative solution from SWORD. KOPIS did a good prediction for 2ALEA protein. If we take a look at the 3D structure of the alternative solution (Fig S6B), it seems like the domain given by KOPIS are actually three PU. Since the protein size if inferior than 60 residues (size for a domain), KOPIS finds PU instead but put it as a second best solution and not the best one because KOPIS is trained to find domain instead of PU but can find it if no more domain can be predicted. The best domain we can find with this small protein is the protein itself which is the optimal solution (Fig S6A).

3. Benchmarks

KOPIS predictions are compared to other method proposals against Jones and Islam benchmarks (Fig 5).

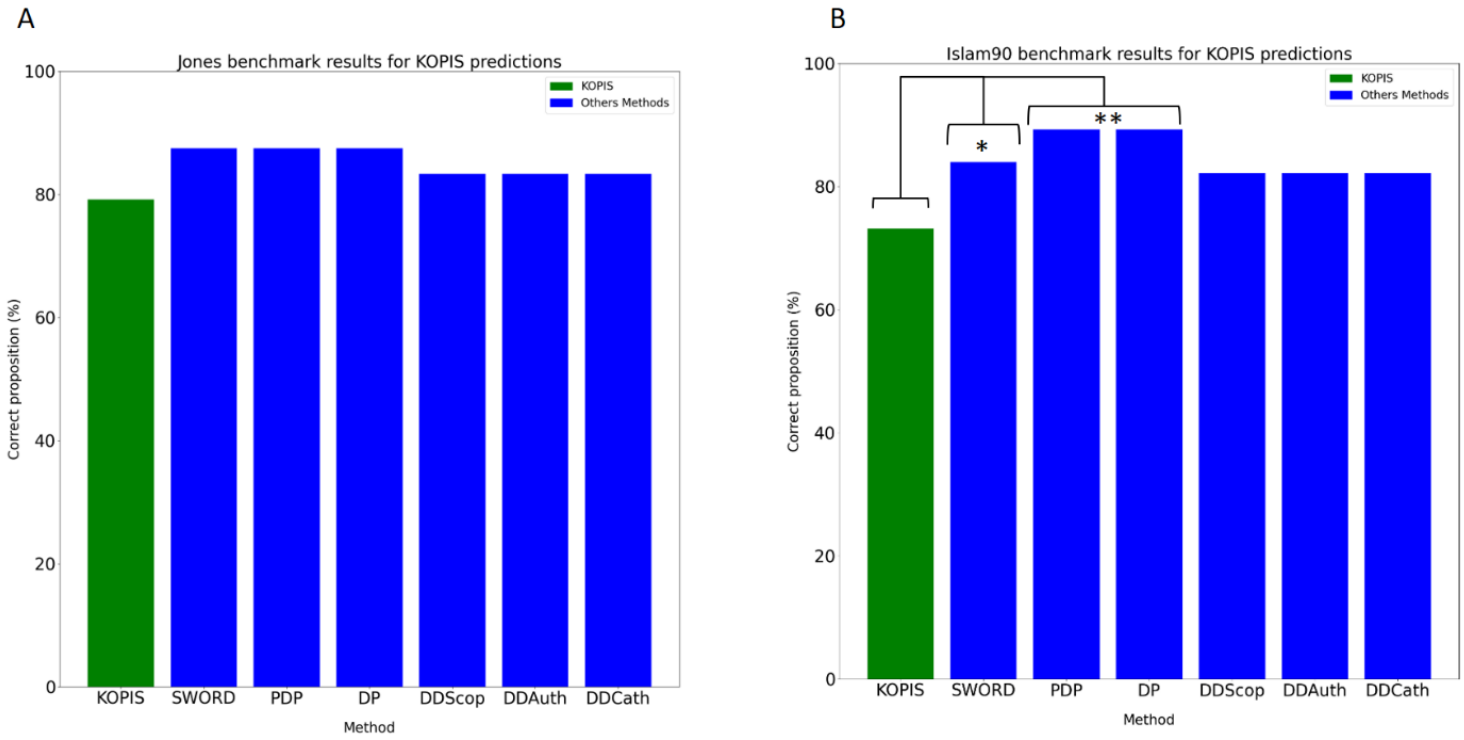


Figure 7: Precision of each method against Jones (A) and Islam90 benchmarks (B). In green are the accuracy of KOPIS predictions. In blue the method proposals for SWORD, PDP, DP and DD (3 versions of it) algorithm. A Wilcoxon signed-rank test is used to know if there are accuracy that are significant different.

For the Jones benchmarks, all methods used are near 80%-85% of accuracy (Fig 7A). KOPIS gives 79.17% of correct boundaries (Fig S4). SWORD, PDP and DP gives 87.5% of correct boundaries. There is no significant difference between KOPIS and other method according to the Wilcoxon test (see p-value in Fig S3). KOPIS gives predictions as good as other method for this first benchmark data set. KOPIS gives less good results for Islam90 benchmark data set with 73.21% of good boundaries predictions. SWORD, PDP and DP gives respectively 83.93%, 89.29% and 89.23% of good proposals. The distribution of predictions against their three distribution (pair wised) is significantly different. The Wilcoxon test between KOPIS and SWORD gives a p-value of 0.033 (*). Against PDP and DP, it gives 0.002 (**) for both methods.

4. Plot time

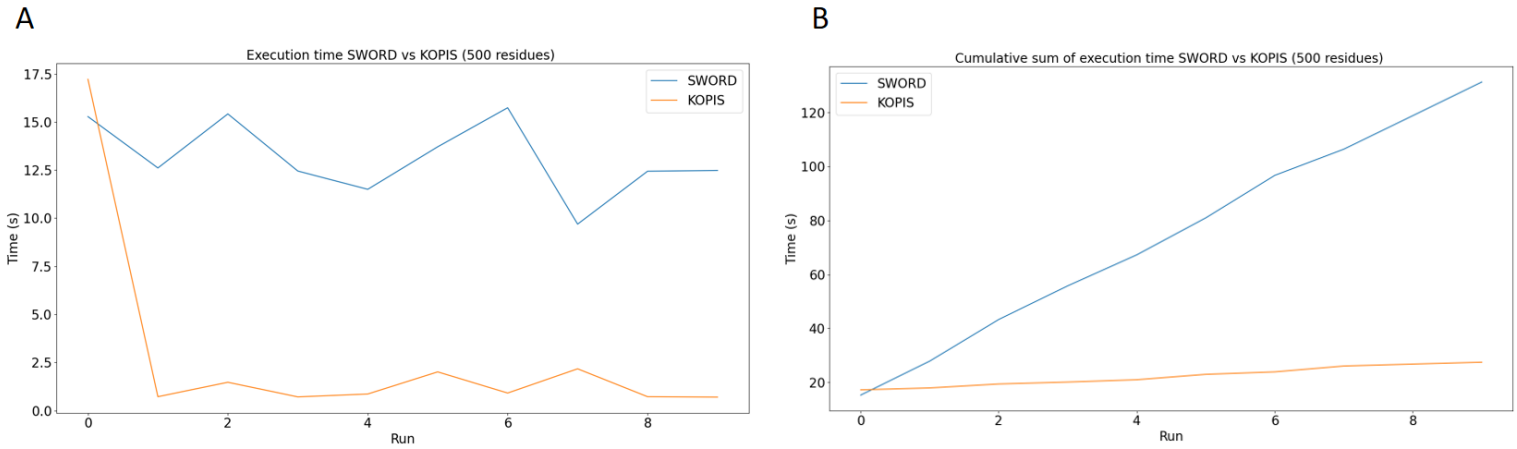


Figure 8: Time execution of KOPIS versus SWORD

KOPIS is a bit slow for the very first prediction. This can be explained by the fact that all the packages and the network itself needs to be fully loaded before make a first prediction. But when this is loaded, KOPIS make prediction much faster than SWORD (Fig 8A) with less than 2.5 seconds per prediction although SWORD execution time is always greater than 10 seconds per prediction. In Fig 8B with the cumulative sum of execution time, KOPIS is clearly better in the case of multiple predictions with less than 30 seconds to predict 10 proteins although SWORD takes more than 2 minutes.

5. Prediction of proteins function and evolution

KOPIS can also predict the domain with function and evolution history. An example of the DNA Polymerase with PDB Code 1JX4. Some database annotates this protein with 3 domains like CATH 3.4, but the version 3.5 and 4.0 annotates it with 4 domains. These two proposition are valid in term of evolution and function and KOPIS find them.

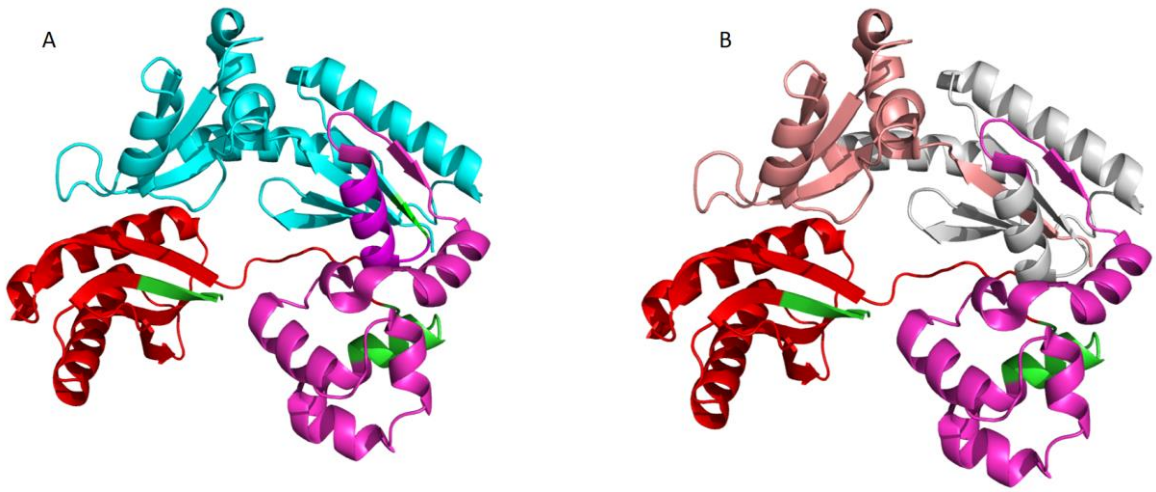


Figure 9: KOPIS prediction for DNA Polymerase (1JX4)

KOPIS give an optimal solution (Fig 9A) containing 3 domains, the three domain from CATH 3.4, but KOPS gives also several alternatives solutions including the solution with 4 domains (Fig 9B).

6. PU predictions

As we seen in Fig S6B, KOPIS can also predict PU if no more domain can be predicted.

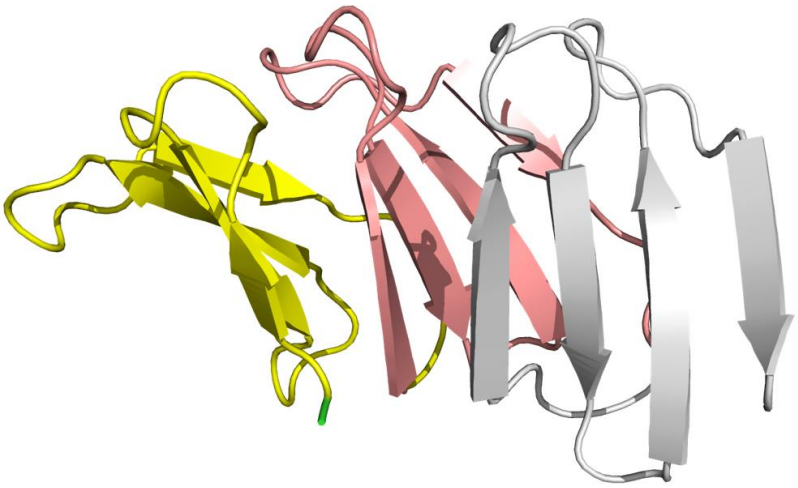


Figure 10: Prediction of PU for a beta propeller (6R5ZA)

Here, KOPIS give a prediction for 6R5ZA and cut it into 3 PU with 4 beta strands for each. The prediction on PU is not yet as accurate as domain prediction.

IV. Conclusions and discussions

As SWORD does, KOPIS can give an optimal solution and also alternative solution for a given protein which can be a solution against the ambiguity of protein structure. KOPIS are much faster than SWORD for multiple predictions (Fig 8), but got a lack of accuracy for some proteins especially for the protein of the Islam90 benchmark where KOPIS prediction are not good enough Fig 7B). It gives good predictions for Jones benchmark.

There is some step to take to improve KOPIS. Better data can be selected for training for example. An optimisation of parameters can be done.

V. References

Alexandrov, Nickolai, and Ilya Shindyalov. 2003. 'PDP: Protein Domain Parser'. *Bioinformatics* 19(3):429–30. doi: 10.1093/bioinformatics/btg006.

Gelly, J.-C., C. Etchebest, S. Hazout, and A. G. de Brevern. 2006. 'Protein Peeling 2: A Web Server to Convert Protein Structures into Series of Protein Units'. *Nucleic Acids Research* 34(Web Server issue):W75-78. doi: 10.1093/nar/gkl292.

Gelly, Jean-Christophe, Alexandre G. de Brevern, and Serge Hazout. 2006. '"Protein Peeling": An Approach for Splitting a 3D Protein Structure into Compact Fragments'. *Bioinformatics (Oxford, England)* 22(2):129–33. doi: 10.1093/bioinformatics/bti773.

Guo, Jun-tao, Dong Xu, Dongsup Kim, and Ying Xu. 2003. 'Improving the Performance of DomainParser for Structural Domain Partition Using Neural Network'. *Nucleic Acids Research* 31(3):944–52. doi: 10.1093/nar/gkg189.

He, Kaiming, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2020. 'Mask R-CNN'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42(2):386–97. doi: 10.1109/TPAMI.2018.2844175.

Islam, Suhail A., Jingchu Luo, and Michael J. E. Sternberg. 1995. 'Identification and Analysis of Domains in Proteins'. *Protein Engineering, Design and Selection* 8(6):513–26. doi: 10.1093/protein/8.6.513.

Jones, Susan, Michael Stewart, Alex Michie, Mark B. Swindells, Chirstine Orengo, and Janet M. Thornton. 1998. 'Domain Assignment for Protein Structures Using a Consensus Approach: Characterization and Analysis'. *Protein Science* 7(2):233–42. doi: 10.1002/pro.5560070202.

Postic, Guillaume, Yassine Ghouzam, Romain Chebrek, and Jean-Christophe Gelly. 2017. 'An Ambiguity Principle for Assigning Protein Structural Domains'. *Science Advances*. doi: 10.1126/sciadv.1600552.

Zhou, Hongyi, Bin Xue, and Yaoqi Zhou. 2007. 'DDOMAIN: Dividing Structures into Domains Using a Normalized Domain–Domain Interaction Profile'. *Protein Science* 16(5):947–55. doi: 10.1110/ps.062597307.

VI. Acknowledgements

I would like to thank Jean-Christophe Gelly for his help and advices during this project. I would also like to thank Alexandre G. de Brevern for giving me access to his computer with GPU at lab to run my program.

VII. Supplementary materials

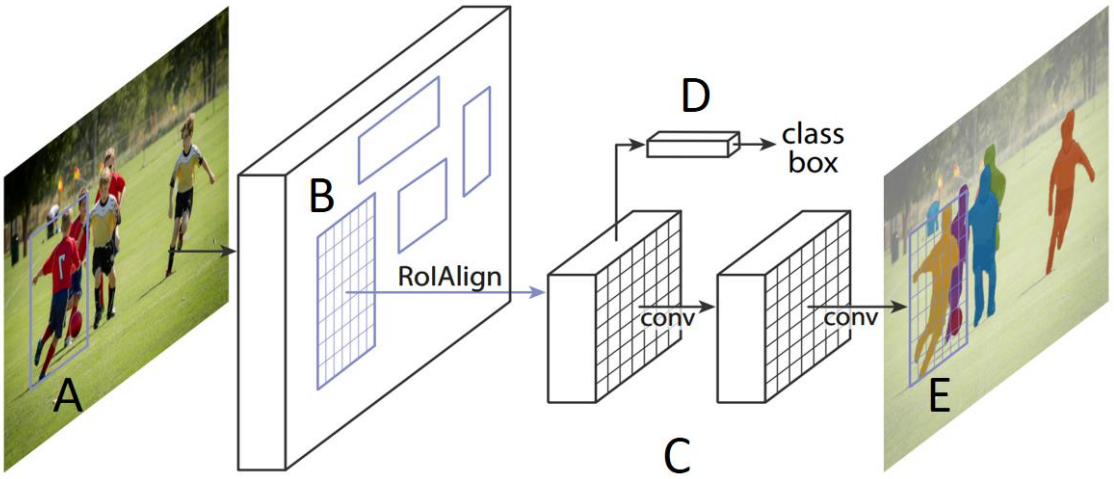


Figure S1 : The MRCNN framework from (He et al. 2020)

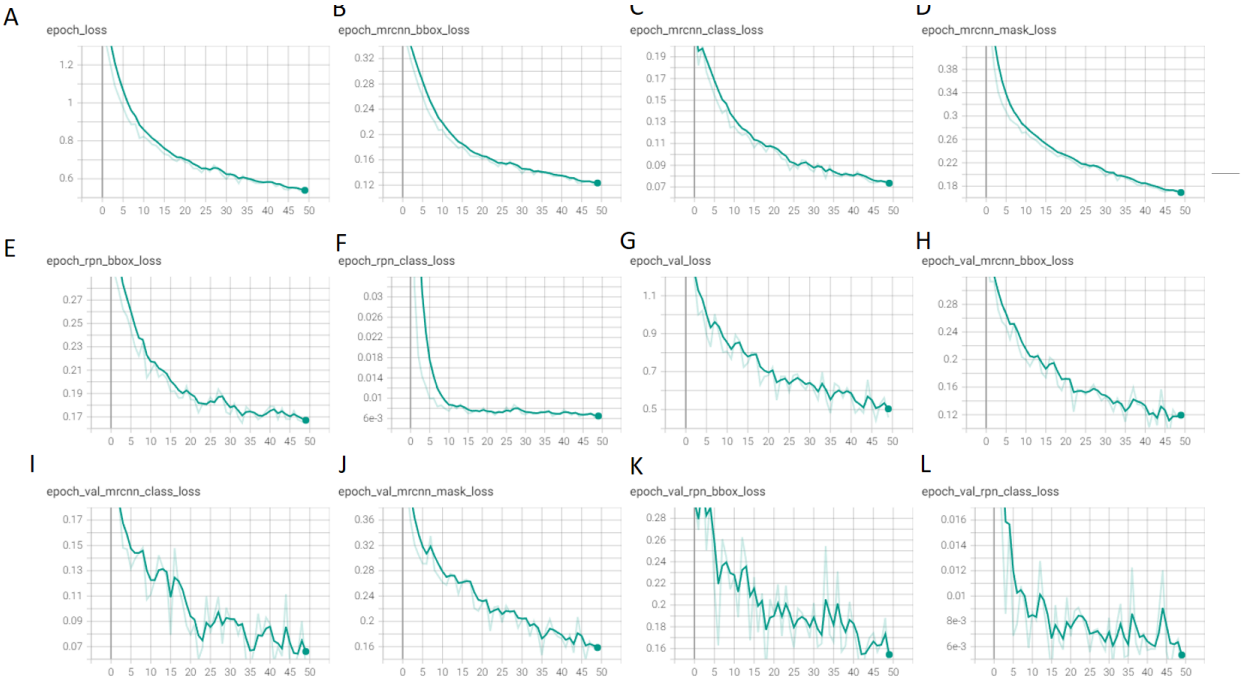


Figure S2: All loss metric from MRCNN

	SWORD	PDP	DP	DDSCOP	DDAuth	DDCath
Jones	0.157	0.157	0.317	0.563	0.157	0.563
Islam90	**0.033**	**0.002**	**0.002**	0.058	0.058	0.058

Figure S3: P-value from the Wilcoxon signed-rank test

	KOPIS	SWORD	PDP	DP	DDSCOP	DDAuth	DDCath
Jones	79.17%	87.5%	87.5%	87.5%	83.33%	83.33%	83.33%
Islam90	73.21%	83.93%	89.29%	89.29%	82.14%	82.14%	82.14%

Figure S4: Accuracy for each method on Jones and Islam90 benchmark data set

```
Predictions for 2ALEA

Prediction      Score   Coverage Prot
(0, 135)        0.9992924       97.04
(97, 132)       0.8866485       26.72
(1, 32)         0.57705724      23.66
(33, 91)        0.5568766       44.27

Best solution  :
---------------
0-135

Alternative #2 :
---------------
1-32 33-91 97-132
```

Figure S5: Example of output prediction from KOPIS for 2ALEA protein

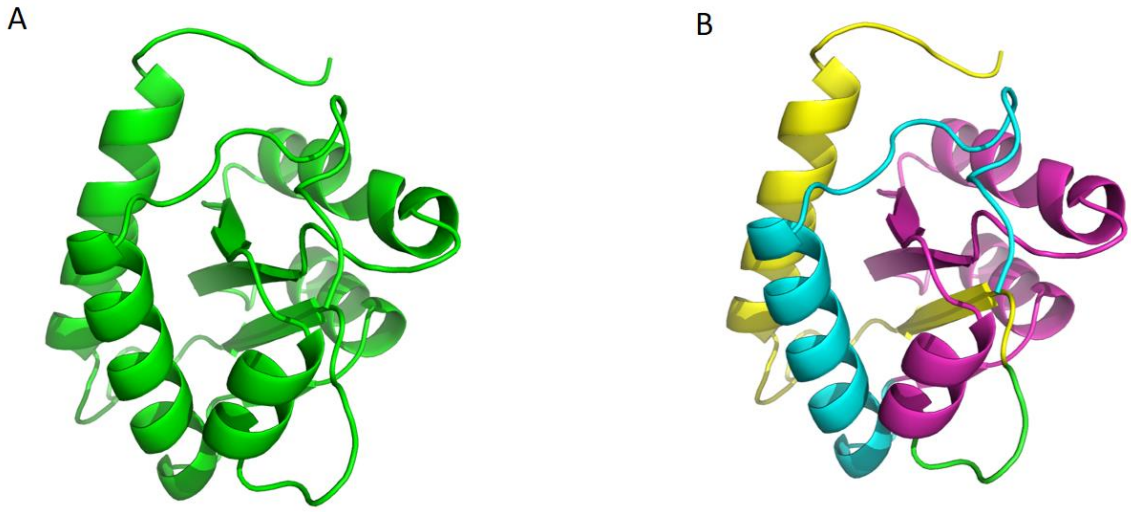


Figure S6 : 3D structure of optimal solution (A) and alternative solution (B). These two solutions are also the solution given by SWORD. KOPIS did a correct prediction.