ProPU

Project Report

Master 2 de Biologie et Informatique
Université Paris Diderot - Paris VII

Inka Leroy

January 2019

Table of contents

Summary

A protein structure can be partitioned into different levels of structures : secondary structures, supersecondary structures, domains, etc… As studying a protein structure helps predicting its function, knowing how to partition a protein into independent and interesting subunits can benefits the assignation of protein function. ProPU is a software written in Python3 and handled by a Shell script that scans a pdb file and suggests positions for protein units (PU). A protein unit is an independent portion of a protein where interactions between its atoms are in higher number than interactions with the rest of the protein. It can represensent a supersecondary structure, a secondary structure, or a bigger part of a protein. ProPU searches independent and compact PUs within a protein based on three criteria : the partition index, the separation criterion and the compactness criterion. It suggests several possibilities of PUs as well as the best ones, up to the limit of size set by the user. With two examples (2aak and 1atn), this report illustrates the capacity of ProPU to suggest interesting and coherent PUs. Results are compared with Protein Peeling 3D software.

Introduction

Knowing the structure of a protein helps predicting its function [1]. But once the structure is available (as a pdb file for example) we only have atoms position in space. It is then interesting to study this structure to identify domains or protein units (PU). Domains are defined as compact and independent parts of a protein. There is a software named SWORD [2] which is an automated method which produces multiple decompositions of protein structure. The method is based on PUs computations and uses other criteria so as to merge PUs and form domains.
PUs are intermediate structures between secondary structure and domains, that contain regular secondary structures and are conserved through evolution time [2]. Thus, delineating those PUs within a protein could benefit the search of structural domains. A PU is defined as a portion of a protein that has a high number of interactions between atoms of this portion, and a low number of interactions between atoms of this portion and atoms from the rest of the protein [3]. Protein Peeling 3D [4] is a web server that identifies PUs in a protein, analysing contacts matrix. First, it cuts the protein in two or three parts that are the most independent from each other, and then it continues to cut within those parts until it creates the best PUs for this protein. The possible issue that can emerge from this method is that after cutting the protein, the algorithm does not step back and thus does not try other possibilities. However, several protein units could be considered when studying a protein structure. In fact, PUs can have different sizes. It can represensent a supersecondary structure, which is the combination of specific and adjacent secondary structures [5]. It also can represent simply a secondary structure, or a bigger part of a protein.Protein Peeling 3D cuts hierarchically, beginning by large part of protein, then cutting smaller part, until getting secondary structures or supersecondary structures.

The aim of this project was to create a program that searches best PUs within a protein so as to suggest different possibilities of PUs and avoid cutting the protein with only one PU. ProPU allows the user to compute several positions for partitioning a protein. ProPU is based on Protein Peeling 3D method but also on SWORD method. Indeed, to combine PUs, SWORD calculates other criteria [2] to define whether two PUs are independent or not. ProPU cuts a PU inside the protein at each step and considers that the rest of the protein is an other PU. Then, it uses derived SWORD criteria between those two PUs.

Materials and methods

ProPU and the options

ProPU is a program written in Python3 [6] handled by a shell script. It can be downloaded at this link https://github.com/inka000/ProPU and works on Linux. It allows to delimitate protein units within a chain of a given protein structure. The user must download the program and follow the subsequent instructions :

Open a terminal and type : *cd protein_peeling*
Make the main script executable typing : *chmod +x ProPU*
Run the program with the command : *./ProPU -i directory_where_pdb_are/*

Options are available for ProPU and can be used in the command line :

-h, --help	*Displays help*
-i, --input	*Directory where pdb files are or path to a pdb file*
--min	*Minimum size for a PU (10 by default)*
--max	*Maximum size for a PU (40 by default)*
--delta	*Parameter of the logistic probability function (1.5 by default)*
--dist	*Distance cut-off for interactions (8.0 by default)*

Besides the pdb file directory, other options exist but are facultative. The user can choose the minimum size of a PU (10 amino acids by default) and the maximum size (40 by default) as long as max size is larger than min size. He/she can also choose delta, the parameter of the logistic probability function (1.5 by default), and d0, the distance cut-off for interactions (8.0 Å by default). When the user run the program, parameters are verified and displayed on the terminal.

The contacts matrix creation

Once the program starts, ProPU handles the creation of directories and copies the pdb files provided by the user into a directory named *Query.* Then it gets the chains available inside the pdb file and asks the user which of the chain he/she wants to analyse. To analyse several chains from a same pdb, the user have to run ProPU several times, as often as there are chains. Then, it reads the part in the pdb file corresponding to the chosen chain and gets the alpha carbon atoms that compose the chains.
Atoms are stored inside a list of *Atome* instances with all important information. This list of atoms is analysed and, based on distances, a distances matrix is computed calculating distance between pairs of atoms.

Thanks to this matrix, it is possible to assess if two atoms could probably interact or not using the formula (1)

$$p(i,j) = \frac{1}{1 + \exp\left[\frac{d(i,j) - d_0}{\Delta}\right]} \quad (1)$$

where *d(i,j)* is the distance between the atom *i* and the atom *j*, d_0 is the distance cut-off for interactions (set to 8.0Å), and Δ is a parameter for this logistic probability function (set to 1.5) (SWORD). Thus, a contacts matrix is created.

Criteria calculation

ProPU scans this contacts matrix so as to find PUs. To this end, it tries to cut different size of PU based on min and max size. For each iteration, it verifies that the beginning and the ending of the PU are not in the middle of a continuous secondary structure. DSSP [7] assigns the secondary structure of the chain. To simplify this assignation and to avoid cutting between two similar structures, ProPU considers that the secondary structures are Helix, β-strands and coils. So, alpha-helix (H), 3/10-helix (G) and 5-helix (I) are set as helix (H), beta-bridge residue (B) and extended strand in beta ladder (E) are set as β-strands, and H-bonded turn (T), bend (S) and blank are set as coils (‘ ‘). Once ProPU verified that it can cut, it calculates the three criteria defined in the introduction : the Partition Index (PI) [4], the separation criterion (σ) [2] and the compactness criterion (κ) [2].
The PI allows to assess splitting quality quantifying the PUs independence based on contacts probability. It is calculated thanks to the formula (2)

$$PI_{i,j}(m) = \frac{AB - C^2}{(A + C)(B + C)} \quad (2)$$

where $PI_{i,j}(m)$ is the PI for a given slicing, A is the sum of contacts probabilities within the PU A, B is the sum of contacts probabilities within the rest of the protein and C is the sum of contacts probabilities between A and B. The more independent the PU A is, the more it has contacts between its atoms, and the less it has contacts with atoms from the rest of the protein (B). Thus, a PI close to 1 means that the PU A is independent.

The separation criterion is another criterion that assesses the independence between a PU and the rest of the protein, according to the formula (3)

$$\sigma_{i,j} = \frac{p_{i,j} / ((S_i)^{\alpha} \times (S_j)^{\alpha})}{p_{i+j} / (S_{tot})} \quad (3)$$

where $p_{i,j}$ is the sum of contacts probabilities between the PU and the rest of the protein, p_{i+j} is the sum of contacts probabilities in the whole protein, S_i is the size of the PU considered, S_j is the size of the rest of the protein and S_{tot} is the size of the whole protein. Thus, a σ close to 0 means that the PU is independent of the rest of the protein.

Finally, the compactness criterion measures the compactness of contacts in a PU according to the formula (4)

$$\kappa_{i,j} = \frac{\Sigma p_{pu}}{S_{tot}} \quad (4)$$

where Σp_{pu} is the sum of contacts probabilities in the PU considered, and S_{tot} is the size of the PU. Thus, a high κ means that there are a lot of interactions inside the PU and that the PU is compact.

PUs analysis

ProPU analyses all PUs. The program will keep only the best PUs based on thresholds of the three criteria. To define those thresholds, the program calculates the z-score associated with each value of PI, σ and κ, as well as the p-value of the z-score. Calculating z-scores provides a normalized distribution of scores and allows to calculates p-values with normalized distribution. p-values are computed according to a two-tails hypothesis (Figure 1). It is whether the z-score is positive or negative that determines if it is a protein unit well identified or on the contrary a protein unit that should not be disconnected from the rest of the protein.

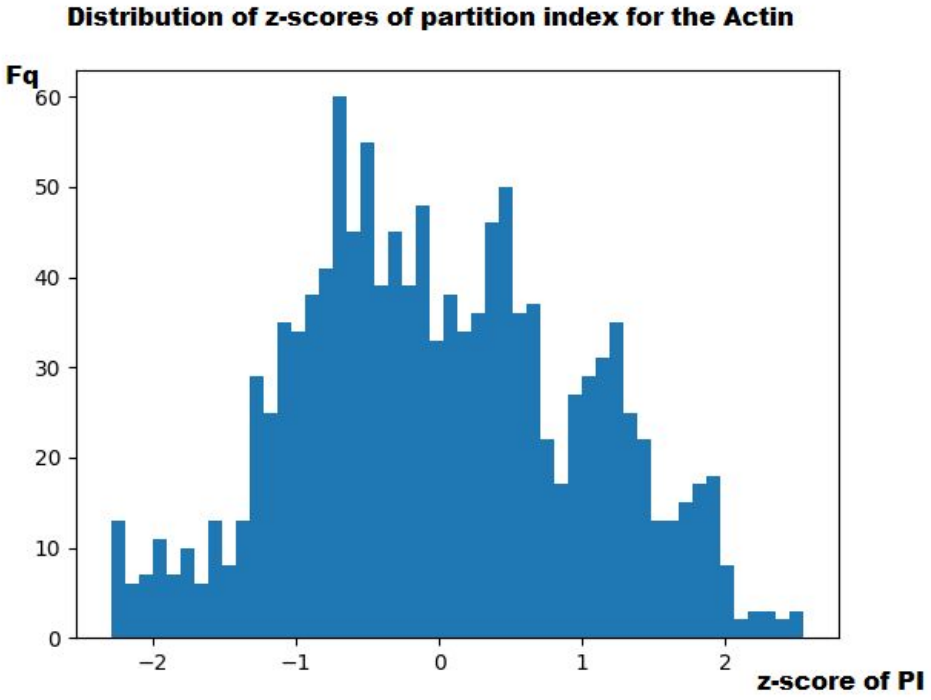

Figure 1. Distribution of z-scores corresponding to partition index values calculating by ProPU for the Actin

Even if there are a lot of p-values that are calculated, they were not corrected insofar as they are more informative than statistical. In fact, PU could have been chosen only based on raw criteria values but p-values allows to have scores relating to all calculated values. As the aim of the program is to find possible PUs to use before continuing to cut in the protein, it is consistent to scan the whole protein at once and to suggest some positions spread in the protein.

A PU is considered as “good” if it fulfils one of the conditions below :

- it has a p-value associated with PI value lower than 0.05 and a positive z-score (set as “P”)
- it has the first condition and a p-value associated with σ value lower than 0.05 and a negative z-score (set as “PS”)
- it has the first condition and a p-value associated with κ value lower than 0.05 and a positive z-score (set as”PK”)
- it has the three previous conditions (set as “PSK”)

Those “good” PUs are registered in a *.txt* file named after the studied chain and the name of the protein with the number “2”. This file allows the user to keep all possible PUs. For example, for purposes of using those PUs as first cutting iteration in a program that cuts within PUs. Then, ProPU analyses a second time the “good” PUs so as to extract the best ones that do not overlap each other. To do so, it finds the best PU from the list seeking for the PU that fulfils the most conditions and with the best PI value. Once the best PU is extracted, ProPU continue the search in a sublist that contains PUs that do not overlap the previously found as “best”.

Those “best” PUs are registered in a *..txt* file named after the studied chain and the name of the protein without any number. In addition, it records graph of contacts matrix as *.png* files named after the studied chain, the name of the protein and the number of the first amino acid of the PU delineated by lines on the matrix. A letter “A” allows the user to locate the PU.

Example of 2aak

An example is available in the program in the directory *example* with the pdb 2aak.

Go to https://github.com/inka000/ProPU and download the program.

Extract the program.

Open a terminal and type :

cd ProPU-master

Make the main script executable typing :

chmod +x ProPU

Make dssp executable typing (if dssp version is not suited for your PC, please download the good version https://github.com/cmbi/xssp/releases and change the path to DSSP in ProPU file line 61) :

chmod +x bin/dssp-2.0.4-linux-amd64

Run the program with the command :

./ProPU -i example/

Type ‘A’ when the program asks which chain to analyse.

Results will be stored in *resultPU/2aak/A_2aak.txt* and *resultPU/2aak/A_2aak2.txt* with *.png* of best PUs on the contacts matrix.

Results and discussion

For this project, three protein structures were analysed with ProPU. For each protein, the structure was visualized thanks to PyMOL.

Results for 2aak

First, the structure of ubiquitin conjugating enzyme from arabidopsis thaliana was used as a test as it was used in the article of Protein Peeling 3D [3]. This protein participates to the second step of ubiquitination reaction that targets a protein for degradation. Its pdb structure was found at https://www.rcsb.org/structure/2aak and visualized on PyMOL (Figure 2). It has a unique chain (A) and contains 150 amino acids. The Figure 3 shows secondary structures of the protein.

Figure 2. Structure of 2aak on PyMOL

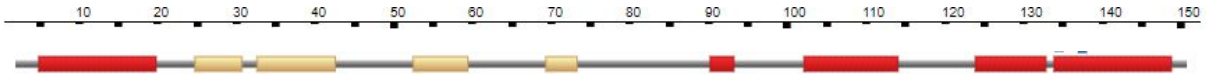

Figure 3. Secondary structure of 2aak, the helix are in red and the beta strands are in yellow

ProPU was used with min and max sizes by default (min size = 10 and max size = 40). ProPU found several significant PUs of different sizes (40 PUs). The program wrote those information inside the file *A_2aak2.txt*. ProPU defined that there were two best PUs :

Results for the chain A of the protein 2aak

begin	end	size	PI	sigma	k	significant
1	20	20	0.54	0.298	4.522	PS
124	150	27	0.523	0.375	4.934	S

The first PU is significant for PI and σ, the second one only for PI. Nevertheless, compactness criterion values of 4.522 and 4.934 show fairly compact PUs.

The Figure 4 shows the PU on the contacts matrix with the letter A in red. This first best PU correspond to the first alpha helix of the protein and the second one correspond to the two last alpha helix (Figure 5). If Protein Peeling 3D software is used with this protein and with

minimum size set to 10 amino acids, it cuts those PUs during the first and second iterations (Figure 6). ProPU prioritized those PUs the same way Protein Peeling 3D does. Indeed, the first helix seems to be quite independent from the rest of the protein and the second PU corresponds to a supersecondary structure helix-coil-helix.

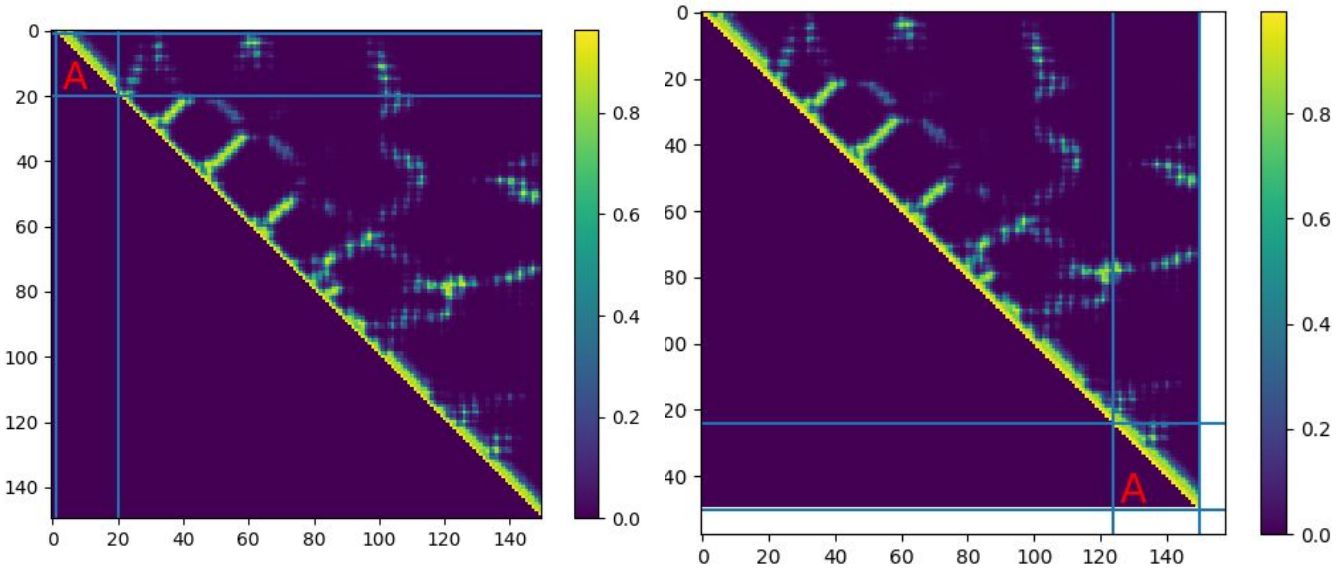

Figure 4. Boundaries of the two best PUs found in 2aak contacts matrix. The letter A in red indicates the PUs.

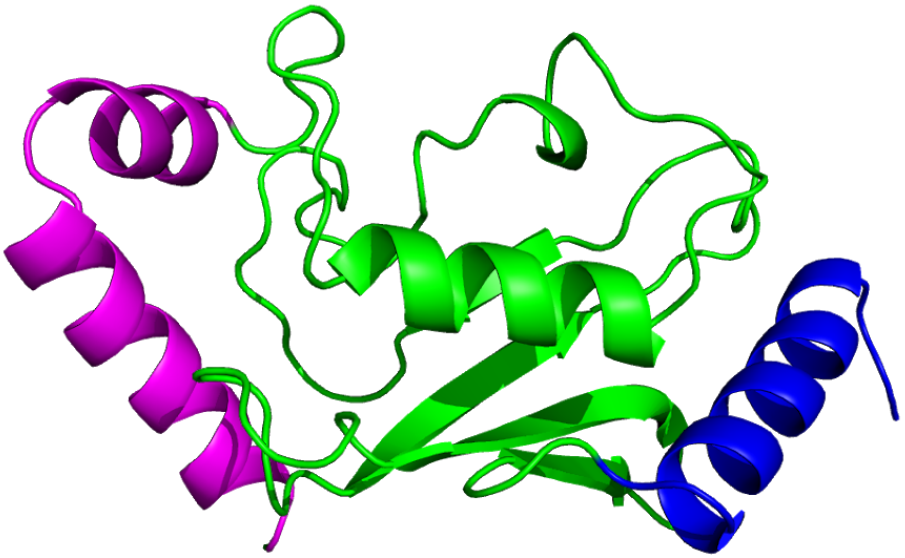

Figure 5. Structure of 2aak on PyMOL with the best PsU found by ProPU : residues 1 to 20 colored in sky blue, residues 124 to 150 colored in magenta

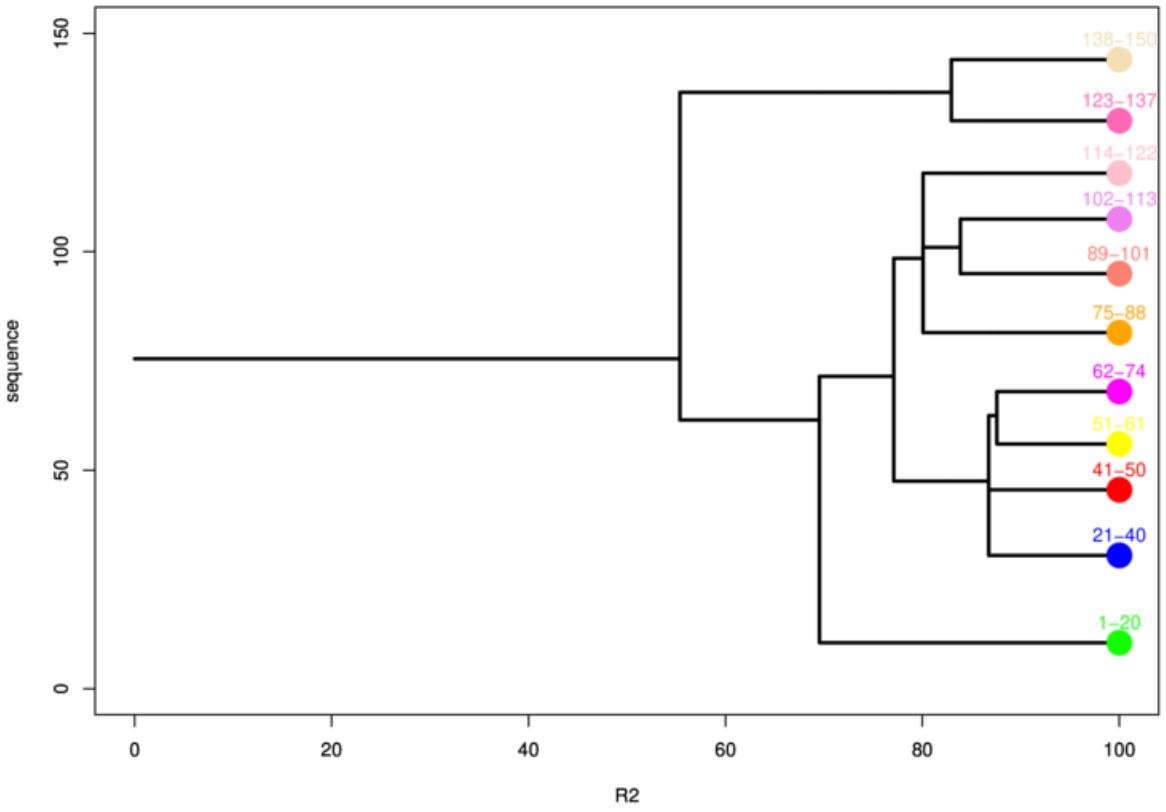

Figure 6. Final PUs proposed by Protein Peeling 3D for 2aak

In any case, ProPU suggest other PUs with slightly different intervals. For 2aak, ProPU suggests those subsequent possibilities :

Results for the chain A of the protein 2aak

begin	end	size	PI	sigma	k	significant
1	18	18	0.512	0.308	4.43	PS
1	19	19	0.532	0.297	4.466	PS
1	20	20	0.54	0.298	4.522	PS
1	21	21	0.53	0.312	4.499	PS
1	22	22	0.523	0.323	4.445	PS
1	23	23	0.503	0.347	4.399	PS
2	18	17	0.487	0.329	4.472	PS
2	19	18	0.508	0.317	4.507	PS
2	20	19	0.517	0.317	4.564	PS
2	21	20	0.508	0.331	4.538	PS
2	22	21	0.502	0.341	4.479	PS
2	23	22	0.483	0.365	4.43	PS
2	**41**	**40**	**0.496**	**0.462**	**5.073**	**P**
3	20	18	0.485	0.34	4.503	PS
3	21	19	0.477	0.354	4.478	PS
3	**41**	**39**	**0.478**	**0.48**	**5.053**	**P**
114	150	37	0.489	0.456	4.955	P

115	148	34	0.482	0.456	4.986	P
115	149	35	0.487	0.453	4.986	P
115	150	36	0.501	0.438	4.98	P
116	148	33	0.488	0.446	5.025	P
116	149	34	0.494	0.443	5.024	P
116	150	35	0.508	0.428	5.017	P
117	148	32	0.491	0.437	5.003	P
117	149	33	0.497	0.435	5.003	P
117	150	34	0.512	0.42	4.996	P
118	148	31	0.487	0.437	4.985	P
118	149	32	0.493	0.434	4.986	P
118	150	33	0.508	0.419	4.98	P
119	150	32	0.491	0.433	4.907	P
120	150	31	0.486	0.435	4.934	P
121	149	29	0.477	0.44	4.965	P
121	150	30	0.493	0.424	4.958	P
122	150	29	0.493	0.419	4.921	P
123	148	26	0.487	0.413	4.953	P
123	149	27	0.494	0.409	4.955	P
123	150	28	0.512	0.393	4.949	P
124	148	25	0.497	0.395	4.938	P
124	149	26	0.504	0.391	4.941	P
124	150	27	0.523	0.375	4.934	P

For the first PU, ProPU selected another kind of possibility that combines two secondary structures (Figure 7) : the alpha helix from residues 2 or 3 to 20, and two beta-strands from residues 21 to 41. However, the separation criterion values for those PUs are to high to be considered as better PUs than smaller ones. Those PUs are in bold in the list of PUs. As the maximum size was defined by default at 40, ProPU stopped before the end of the beta-sheet. However, even if the max size were expanded, ProPU did not suggest to extend this first PU. The PI value was not enough.For the second PU, the main difference between possibilities is the beginning that varies along a coil.

Concerning the significativity of values, for this example ProPU kept PI values that were 11.7% lower than the best one. For σ and κ, as PI values prevail for selection, they are not necessarily the best ones but participate to the selection of “best” PUs.

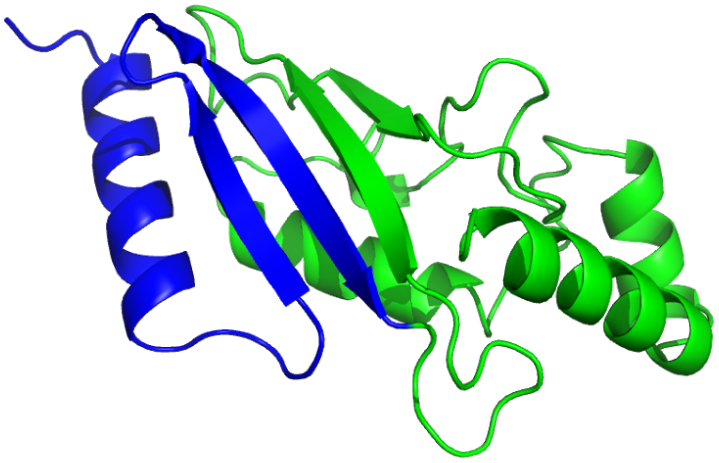

Figure 7. Structure of 2aak on PyMOL with another significant PU suggested by ProPU colored in blue (residues 2 to 41)

Results for 1atn

Then, the structure of rabbit skeletal muscle actin was studied. The pdb file 1atn is the structure of the complex between actin and the bovine pancreatic deoxyribonuclease I, with the actin as the chain A. For the rest of the analysis, actin refers to the chain A of 1atn. Actins participate to several types of cell mobility, they are highly conserved proteins and are expressed in all eukaryotic cells [8]. Its pdb structure was found at https://www.rcsb.org/structure/1atn and visualized on PyMOL (Figure 8).

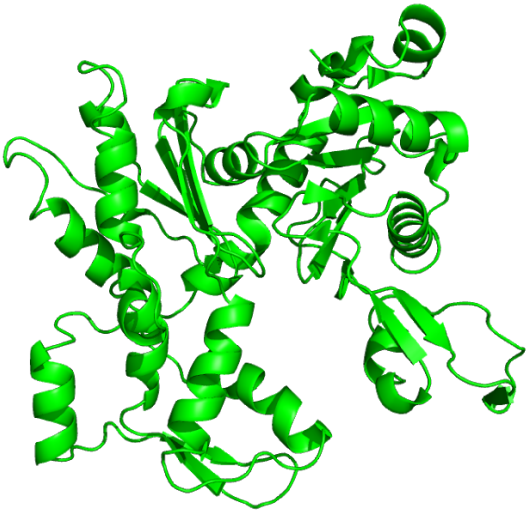

Figure 8. Structure of Actin (1atn chain A) visualized on PyMOL

Actin is composed of 372 amino acids. Its secondary structure is described in the figure 9.

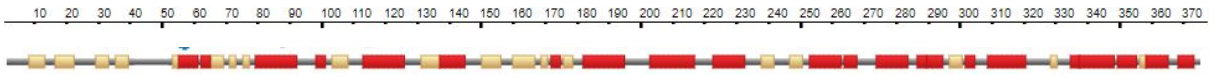

Figure 9. Secondary structure of actin, the helix are in red and the beta strands are in yellow

ProPU was used with 1atn, min and max sizes were left at default values (min size = 10 and max size = 40). ProPU found several significant PUs of different sizes (96 PUs). The program wrote those information inside the file *A_1atn2.txt*. ProPU defined that there were three best PUs :

Results for the chain A of the protein 1atn

begin	end	size	PI	sigma	k	significant
34	68	35	0.766	0.126	5.473	PSK
222	252	31	0.691	0.148	4.702	PS
182	216	35	0.611	0.241	5.31	P

The first PU is significant for PI, σ and κ values, the second one only for PI and σ values, and the last one just for PI value. Nevertheless, compactness criterion values are high for the three PUs which demonstrate compact PUs, and separation criterion values for the two firsts PUs imply independent PUs.

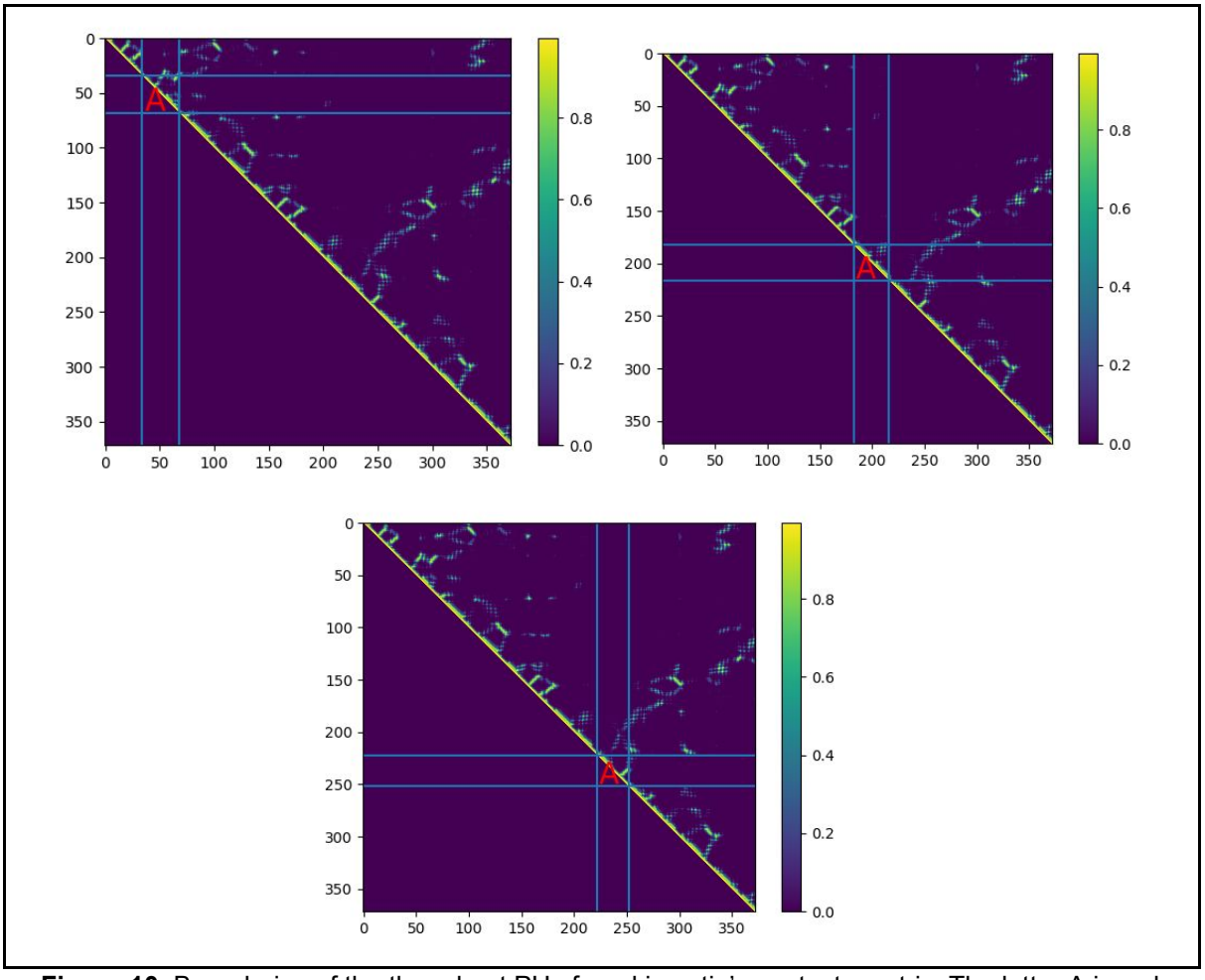

Figure 10. Boundaries of the three best PUs found in actin's contacts matrix. The letter A in red indicates the PUs.

The Figure 10 shows the PUs on the contacts matrix with the letter A in red. The first PU (34-68) correspond to several beta strands and a little alpha helix, the second PU (222-252) correspond to an alpha helix followed-up by two beta strands, and the last PU (182-216) correspond to a supersecondary structure helix-coil-helix (Figure 11). If Protein Peeling 3D

software is used with this structure and with minimum size of PU set to 10 amino acids, it cuts the first and the last PUs at the end of the process, but does not cut the second PU the same way (Figure 12). Protein Peeling 3D separates the last PU found by ProPU in two. However, as ProPU is only used to find other possibilities of partitioning, at the end this whole PU could be cut in two as Protein Peeling 3D did it. Indeed, the PUs suggested by Protein Peeling 3D are the helix on one side and the two beta-strands on the other side which seems more consistent than merging those two secondary structures.

Figure 11. Colored best PUs found by ProPU in actin, residues 34-68 in blue, residues 182-216 in magenta, residues 222-252 in red

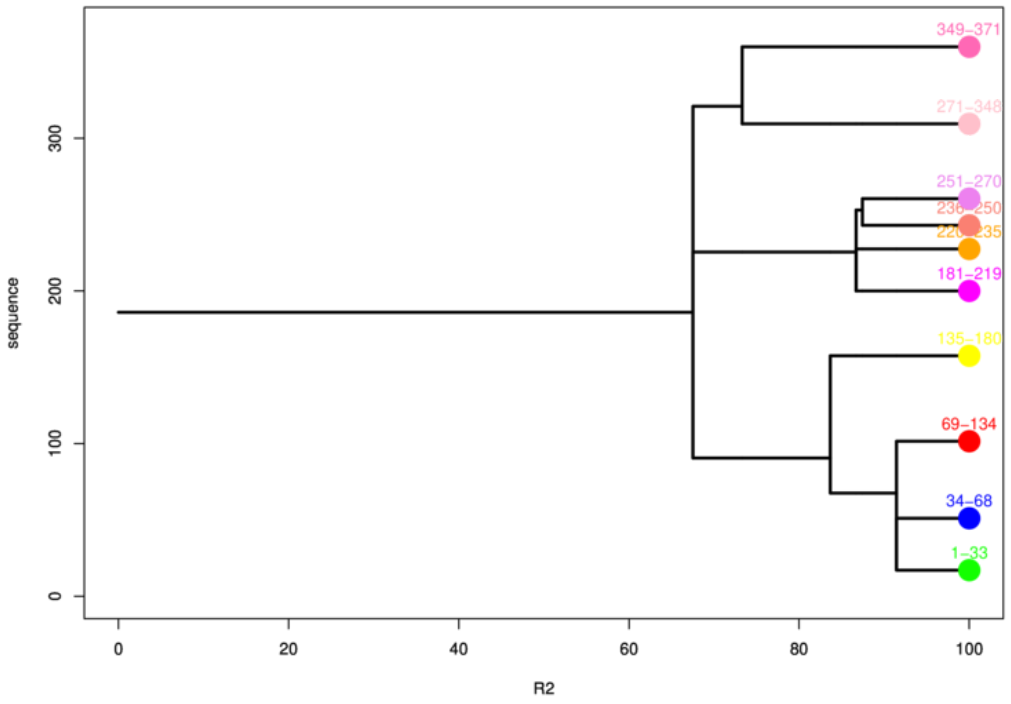

Figure 12. Final PUs proposed by Protein Peeling 3D for actin

Best PUs suggested by ProPU are focused on independent portions of the protein, which is relevant with what should be expected.
In any case, ProPU suggests other PUs that could be interesting :

Results for the chain A of the protein 1atn

begin	end	size	PI	sigma	k	significant
29	64	36	0.589	0.249	4.922	P
29	68	40	0.704	0.18	5.449	PK
33	64	32	0.618	0.211	4.915	P
33	**68**	**36**	**0.746**	**0.141**	**5.492**	**PSK**
33	**69**	**37**	**0.743**	**0.146**	**5.516**	**PSK**
33	70	38	0.726	0.161	5.509	PK
33	72	40	0.689	0.192	5.438	PK
34	64	31	0.637	0.192	4.898	P
34	**68**	**35**	**0.766**	**0.126**	**5.473**	**PSK**
34	**69**	**36**	**0.756**	**0.134**	**5.477**	**PSK**
34	70	37	0.733	0.152	5.454	PK
34	72	39	0.69	0.187	5.365	PK
34	73	40	0.669	0.205	5.312	P
35	64	30	0.638	0.185	4.801	P
35	68	34	0.756	0.128	5.348	PS
35	69	35	0.739	0.141	5.33	PS
35	70	36	0.711	0.162	5.292	P
35	72	38	0.666	0.198	5.198	P
35	73	39	0.645	0.217	5.146	P
35	74	40	0.624	0.236	5.12	P
39	49	11	0.627	0.079	3.244	PS
39	50	12	0.655	0.077	3.396	PS
39	51	13	0.658	0.081	3.442	PS
39	52	14	0.621	0.099	3.46	PS
39	68	30	0.594	0.207	4.475	P
40	49	10	0.662	0.065	3.273	PS
40	50	11	0.687	0.064	3.428	PS
40	51	12	0.683	0.069	3.46	PS
40	52	13	0.635	0.089	3.465	PS
41	50	10	0.694	0.059	3.436	PS
41	51	11	0.683	0.066	3.455	PS
41	52	12	0.63	0.087	3.454	PS
42	51	10	0.66	0.069	3.406	PS
42	52	11	0.605	0.091	3.403	PS
181	216	36	0.596	0.259	5.309	P
181	217	37	0.593	0.266	5.343	P
182	216	35	0.611	0.241	5.31	P
182	217	36	0.608	0.249	5.345	P
182	218	37	0.592	0.266	5.314	P
217	252	36	0.603	0.226	4.674	P
217	256	40	0.64	0.22	5.051	P
218	251	34	0.6	0.217	4.55	P
218	252	35	0.622	0.208	4.683	P
218	256	39	0.642	0.214	5.003	P

218	257	40	0.631	0.227	5.033	P
219	250	32	0.597	0.209	4.433	P
219	251	33	0.621	0.199	4.566	P
219	252	34	0.642	0.19	4.697	P
219	256	38	0.644	0.208	4.959	P
219	257	39	0.629	0.224	4.979	P
219	258	40	0.619	0.237	5.011	P
220	236	17	0.6	0.147	4.309	PS
220	237	18	0.6	0.152	4.341	P
220	250	31	0.624	0.186	4.448	P
220	251	32	0.649	0.176	4.584	P
220	252	33	0.67	0.168	4.715	P
220	256	37	0.655	0.196	4.932	P
220	257	38	0.636	0.214	4.941	P
220	258	39	0.62	0.231	4.953	P
221	235	15	0.592	0.137	4.155	PS
221	236	16	0.612	0.134	4.245	PS
221	237	17	0.611	0.139	4.282	PS
221	250	30	0.633	0.175	4.418	P
221	251	31	0.658	0.165	4.558	P
221	252	32	0.679	0.158	4.69	P
221	256	36	0.652	0.194	4.876	P
221	257	37	0.632	0.213	4.882	P
221	258	38	0.612	0.232	4.884	P
222	234	13	0.596	0.126	4.182	PS
222	235	14	0.607	0.124	4.147	PS
222	236	15	0.627	0.121	4.243	PS
222	237	16	0.626	0.127	4.282	PS
222	250	29	0.644	0.165	4.422	P
222	251	30	0.67	0.155	4.567	P
222	252	31	0.691	0.148	4.702	PS
222	256	35	0.659	0.186	4.882	P
222	257	36	0.638	0.206	4.887	P
222	258	37	0.617	0.226	4.886	P
223	235	13	0.589	0.126	4.072	PS
223	236	14	0.611	0.123	4.179	PS
223	237	15	0.611	0.129	4.225	PS
223	250	28	0.637	0.166	4.396	P
223	251	29	0.663	0.156	4.547	P
223	252	30	0.685	0.149	4.686	P
223	256	34	0.652	0.189	4.867	P
223	257	35	0.631	0.208	4.872	P
223	258	36	0.609	0.229	4.87	P
224	236	13	0.611	0.116	4.095	PS
224	237	14	0.61	0.122	4.15	PS
224	250	27	0.639	0.161	4.364	P
224	251	28	0.666	0.151	4.52	P
224	252	29	0.687	0.144	4.66	PS
224	256	33	0.644	0.19	4.813	P
224	257	34	0.622	0.21	4.817	P

224	258	35	0.599	0.232	4.811	P
233	251	19	0.59	0.157	4.192	P

Several PUs between 33 and 73 could be interesting candidates as they have significant criteria values (in bold in the list of PUs). Even if the best one is the one presented before, it could be interesting to study other partitions.
This time, ProPU selected PI values that were 23.1% lower than the best one, which is less restrictive than for 2aak. However, as PI values are higher than PI values of 2aak, selected PUs are still good candidates.

Conclusion

ProPU allows the user to scan a protein structure and to find the best protein units based on three criteria : partition index, separation criterion and compactness criterion. It combines the search of PU of Protein Peeling 3D with the partition index, but also the evaluation of PU's quality with criteria from SWORD. At the end, several PUs are suggested as potential first PUs to cut for Protein Peeling 3D's first iteration. Other criteria are used by Protein Peeling 3D, however it seems that criteria used by ProPU are sufficient to defined PUs of good quality.
ProPU based it search mainly on the partition index values. Compactness and separation criteria participate to the first selection but PI prevails for the second selection. Moreover, the "good" PUs are selected only if the PI values are significant. Indeed, it was more relevant to rely on PI as it assesses the units independence in terms of contacts. It selects PUs based on PI value and this selection depends on every PIs of the protein. Thus, it considers that all PIs depend on each other. Maybe it would have been better to select PIs within an interval so that for a large protein, the selection of a PI on one end does not depend on the PI from the other end.
In any case, ProPU only finds the most independent PU within a protein and does not reiterate PU selection inside those PUs. Furthermore, this selection depends highly on the max and min size defined by the user. For better results, it is preferable to try different sizes of PU.
The next step could be to reiterate inside part of the protein that were not considered as PUs, e.i. the "rest" of the protein. Indeed, as PU selection depends on the size put in option, it is possible that the large portions left contain PUs that do not fulfil selection criteria as PIs selection depends on all PIs values. Furthermore, the program could have a PyMOL script generator in which residues are selected and colored according to found PU to facilitate visualisation. Another information could also be the main secondary or supersecondary structure for each PU specified in the *.txt* file. Lastly, other criteria from Protein Peeling 3D software could be used to improve ProPU.

References

[1] Najmanovich, R.J., Torrance, J.W., and Thornton, J.M. (2005). Prediction of protein function from structure: Insights from methods for the detection of local structural similarities. BioTechniques 38, 847–851.

[2] Postic, Guillaume, Yassine Ghouzam, Romain Chebrek, and Jean Christophe Gelly. 2017. “An Ambiguity Principle for Assigning Protein Structural Domains.” Science Advances 3 (1). American Association for the Advancement of Science. doi:10.1126/sciadv.1600552.

[3] 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. SERV. ISS.). doi:10.1093/nar/gkl292.

[4] 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 22 (2): 129–33. doi:10.1093/bioinformatics/bti773.

[5] Bhagavan, N.V., and Chung-Eun Ha. 2011. “74-Three-Dimensional Structure of Proteins.” Essentials of Medical Biochemistry, 29–38. doi:10.1007/978-3-642-03022-2_1.

[6] Python Core Team (2015). Python: A dynamic, open source programming language. Python Software Foundation. URL https://www.python.org/.

[7] Wouter G Touw, Coos Baakman, Jon Black, Tim AH te Beek, E Krieger, Robbie P Joosten, Gert Vriend. A series of PDB related databases for everyday needs. Nucleic Acids Research 2015 January; 43(Database issue): D364-D368.

[8] https://www.uniprot.org/uniprot/P68133