Tutorial : Making the interactions graph of a transaminase, bound to its coFactor, reacting with oxoacetic acid.

We are study the following reaction :

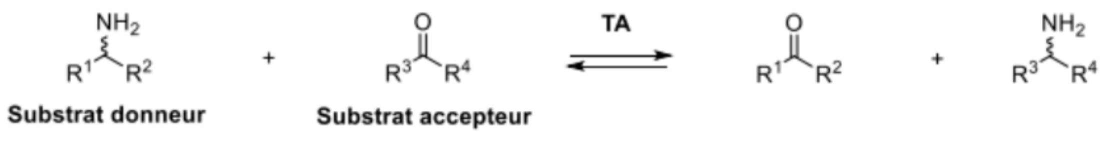


Schéma 25 : Réaction de transamination.

Here the “Ping pong” mechanism :

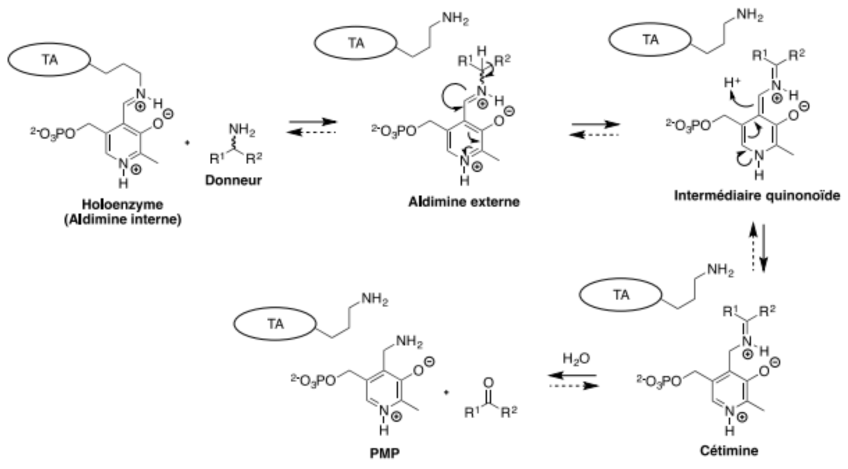
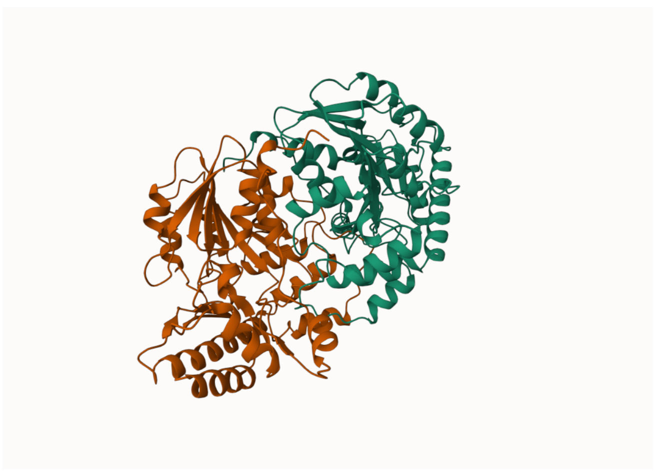


Schéma 26 : Mécanisme réactionnel de la transamination enzymatique.

taken from : Egon Heuson. Recherche de nouvelles transaminases pour la synthèse d'amines chirales. Chimie organique. Université Blaise Pascal - Clermont-Ferrand II, 2015. Français. ⟨NNT : 2015CLF22659⟩. ⟨tel-01912560⟩

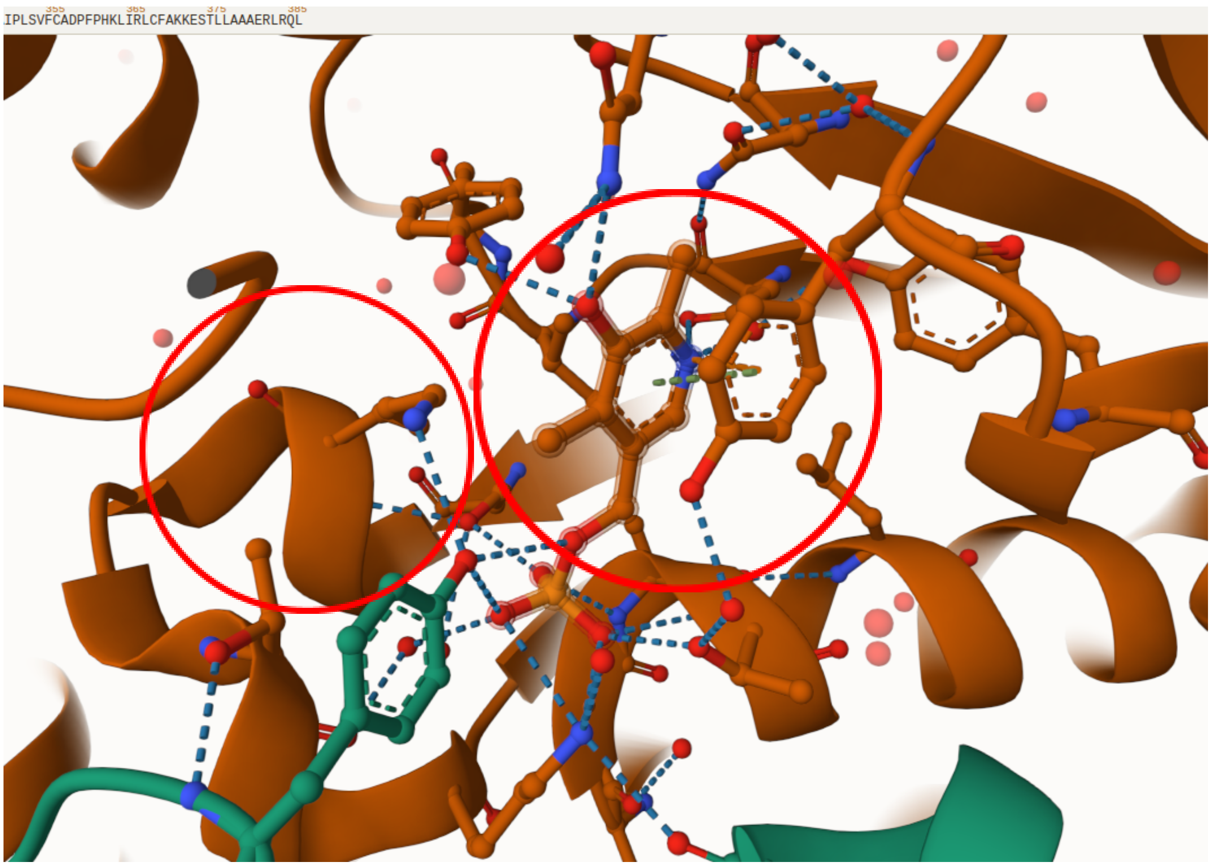
Let's take a transaminase, the fresh “A0AK37_relaxed.pdb” :



Here we have a dimere (it is a particularity of transaminase enzyme), obtained using colafold : https://github.com/sokrypton/ColabFold.

We know that this reaction needs the cofactor intervention to occur. Thus, if we find in which pocket the cofactor is, we find where the reaction takes place. Therefore : where we are going to dock.

First, let's understand how the cofactor (PLP) is bound to the enzyme. Let's get some experimental crystallized data from Uniprot :

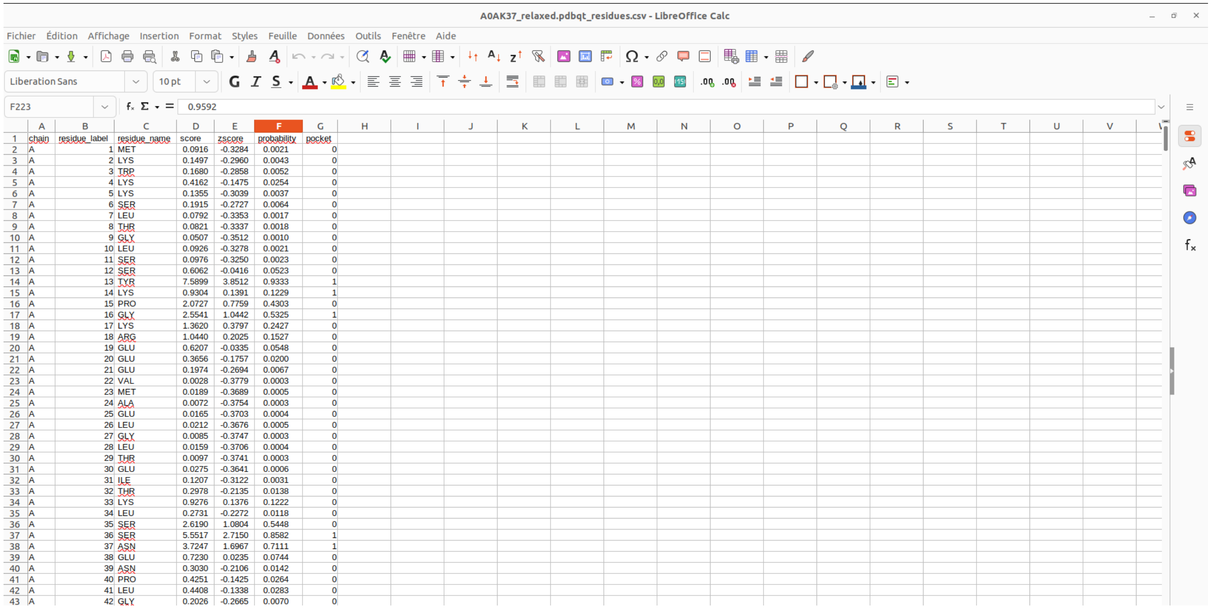


We observe the cofactor neighborhood on Mol* Viewer. Something very interesting can be observed, the cofactor : **Pyridoxal 5 Phosphate is always next to a precise Lysine** (on each dimere), covalently bonded or not, depending on the crystallized data.

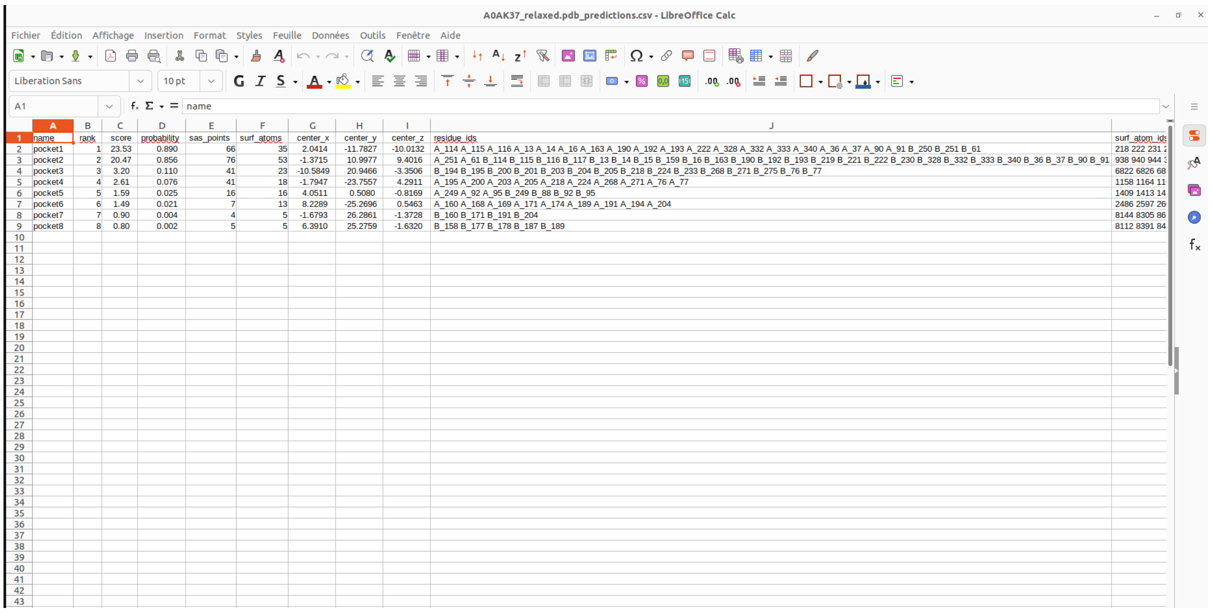
Thus, if we find the lysine where to dock the cofactor, we will know where to perform the docking !

To do so, we use P2rank : https://github.com/rdk/p2rank., a program predicting the best binding pockets for : protein. Be careful, it appears that it works pretty well on transaminase because the model has been trained on enzyme families. Yet, it is not a proof, therefore, I invite you to compare P2rank predictions to experimental Data before doing anything.

So, let's predict the best pockets of our enzyme using : get_boxes() functions. (lets documentation for further information on the functions). The output will be three files :


A0AK37_relaxed.pdbqt_residues.csv - LibreOffice Calc

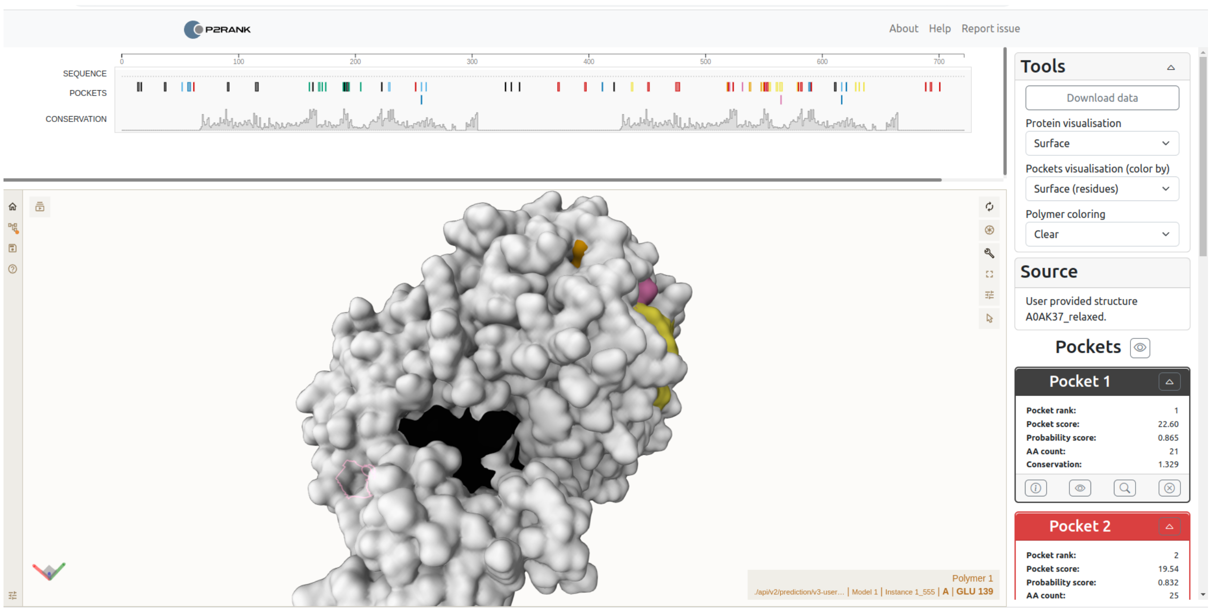
chain	residue_label	residue_name	score	zscore	probability	pocket
A	1	MET	0.0916	-0.3284	0.0021	0
A	2	LYS	0.1497	-0.2960	0.0043	0
A	3	TRP	0.1680	-0.2858	0.0052	0
A	4	LYS	0.4162	-0.1475	0.0254	0
A	5	LYS	0.1355	-0.3039	0.0037	0
A	6	SER	0.1915	-0.2727	0.0064	0
A	7	LEU	0.0792	-0.3353	0.0017	0
A	8	THR	0.0821	-0.3337	0.0018	0
A	9	GLY	0.0507	-0.3512	0.0010	0
A	10	LEU	0.0926	-0.3278	0.0021	0
A	11	SER	0.0976	-0.3250	0.0023	0
A	12	SER	0.6062	-0.0416	0.0523	0
A	13	TYR	7.5899	3.8512	0.9333	1
A	14	LYS	0.9304	0.1391	0.1229	1
A	15	PRO	2.0727	0.7759	0.4303	0
A	16	GLY	2.5541	1.0442	0.5325	1
A	17	LYS	1.3620	0.3797	0.2427	0
A	18	ARG	1.0440	0.2025	0.1527	0
A	19	GLU	0.6207	-0.0335	0.0548	0
A	20	GLU	0.3656	-0.1757	0.0200	0
A	21	GLU	0.1974	-0.2694	0.0067	0
A	22	VAL	0.0028	-0.3779	0.0003	0
A	23	MET	0.0189	-0.3689	0.0005	0
A	24	ALA	0.0072	-0.3754	0.0003	0
A	25	GLU	0.0165	-0.3703	0.0004	0
A	26	LEU	0.0212	-0.3676	0.0005	0
A	27	GLY	0.0085	-0.3747	0.0003	0
A	28	LEU	0.0159	-0.3706	0.0004	0
A	29	THR	0.0097	-0.3741	0.0003	0
A	30	GLU	0.0275	-0.3641	0.0006	0
A	31	ILE	0.1207	-0.3122	0.0031	0
A	32	THR	0.2978	-0.2135	0.0138	0
A	33	LYS	0.9276	0.1376	0.1222	0
A	34	LEU	0.2731	-0.2272	0.0118	0
A	35	SER	2.6190	1.0804	0.5448	0
A	36	SER	5.5517	2.7150	0.8582	1
A	37	ASN	3.7247	1.6967	0.7111	1
A	38	GLU	0.7230	0.0235	0.0744	0
A	39	ASN	0.3030	-0.2106	0.0142	0
A	40	PRO	0.4251	-0.1425	0.0264	0
A	41	LEU	0.4408	-0.1338	0.0283	0
A	42	GLY	0.2026	-0.2665	0.0070	0


A0AK37_relaxed.pdb_predictions.csv - LibreOffice Calc

name	rank	score	probability	sas_points	surf_atoms	center_x	center_y	center_z	residue_ids	surf_atom_ids
pocket1	1	23.53	0.890	66	35	2.0414	-11.7827	-10.0132	A_114 A_115 A_116 A_13 A_14 A_16 A_163 A_190 A_192 A_193 A_222 A_328 A_332 A_333 A_340 A_36 A_37 A_90 A_91 B_250 B_251 B_61	218 222 231 2
pocket2	2	20.47	0.856	76	53	-1.3715	10.9977	9.4016	A_251 A_61 B_114 B_115 B_116 B_117 B_13 B_14 B_15 B_159 B_16 B_163 B_190 B_192 B_193 B_219 B_221 B_222 B_230 B_328 B_332 B_333 B_340 B_36 B_37 B_90 B_91	938 940 944 3
pocket3	3	3.20	0.110	41	23	-10.5849	20.9466	-3.3506	B_194 B_195 B_200 B_201 B_203 B_204 B_205 B_218 B_224 B_233 B_268 B_271 B_275 B_76 B_77	6822 6826 68
pocket4	4	2.61	0.076	41	18	-1.7947	-23.7557	4.2911	A_195 A_200 A_203 A_205 A_218 A_224 A_268 A_271 A_76 A_77	1158 1164 11
pocket5	5	1.59	0.025	16	16	4.0511	0.5080	-0.8169	A_249 A_92 A_95 B_249 B_88 B_92 B_95	1409 1413 14
pocket6	6	1.49	0.021	7	13	8.2289	-25.2696	0.5463	A_160 A_168 A_169 A_171 A_174 A_189 A_191 A_194 A_204	2486 2597 26
pocket7	7	0.90	0.004	4	5	-1.6793	26.2861	-1.3728	B_160 B_171 B_191 B_204	8144 8305 86
pocket8	8	0.80	0.002	5	5	6.3910	25.2759	-1.6320	B_158 B_177 B_178 B_187 B_189	8112 8391 84

The first gives you the list of all the residues, and the pocket they're in, with a probability score.
The second gives information on the predicted pockets (rank, probability : more like a binding score probability, residue in the pockets …)

To have a more graphical result let's use the web serveur of P2Rank output : (https://prankweb.cz/)



Tools

Protein visualisation: Surface

Pockets visualisation (color by): Surface (residues)

Polymer coloring: Clear

Source

User provided structure A0AK37_relaxed.

Pockets

Pocket 1

Pocket rank:	1
Pocket score:	22.60
Probability score:	0.865
AA count:	21
Conservation:	1.329

Pocket 2

Pocket rank:	2
Pocket score:	19.54
Probability score:	0.832
AA count:	25

Here we see our pockets !

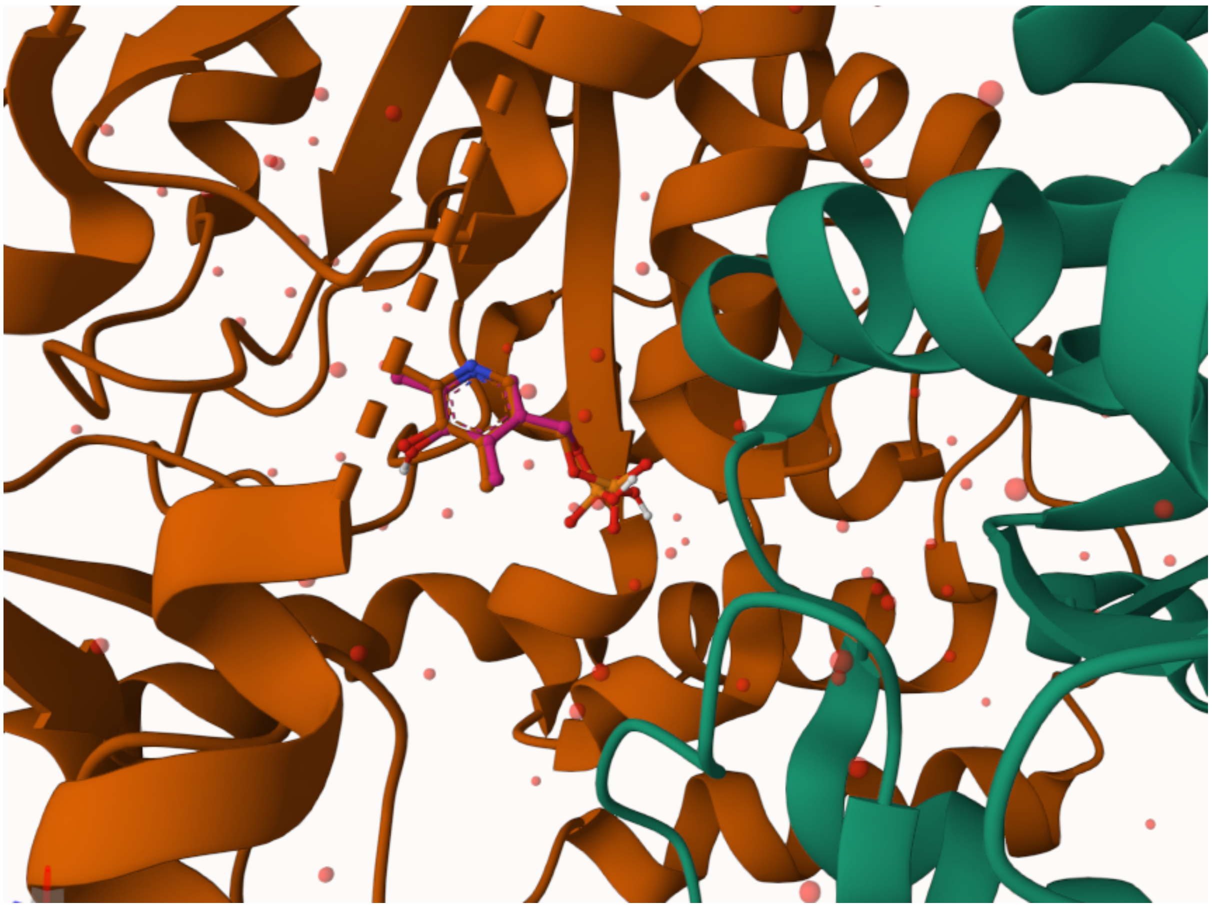
Now, we are going to make an assumption, the Lysine of interest is the Lysine with the best score, in the best pockets.
It works pretty well for tested transaminase structures (compared to crystallized data), but it is still an assumption.
So, We are going to make the docking of Cofactor pyridoxal 5 phosphate (surrounded in red hereinabove).

To do so, we are going to use best_pocket_docking() functions. In fact, It actually do :

1. Find the interest residue (the type of residue is an input) (with the method we have just seen) with get_residue() function
2. Create the best box to dock with centered_box() function. The best "best box", is the intersection of the residue neighborhood and the pocket of the residue. That's allow to dock in "free space", around the interest residue, with good affinities, and following experimental Data.
3. Dock with Gnina Flexible docking GPU (you can modify the flexible parameter, it is 0 in the program, because it can cause problem of convergence) : https://github.com/gnina/gnina

ADVICE, to do so : I highly recommend you to use the PDBQT files as input for the ligand (cofactor, donor, whatever), even if GNINA can have pdb or sdf input. I work pretty much better. You can use prepare_ligand() or prepare_rececptor() to do so.
Before showing the result, let's perform a test on the experimental data we observed before, (pdb1u08). We have just made a copy and cut off the cofactor and water molecules from the pdb files. Cut off the connection to, it can work only with ATOM mentioned.
Finally, you need to check what is center_atom_number : here the N at the end of the C chain for the Lysine. It depends on the file, so check how it has been written in your case.

So let's begin ! The result : (here are the docked cofactor, and the real cofactor) Interessant, isn't it ?

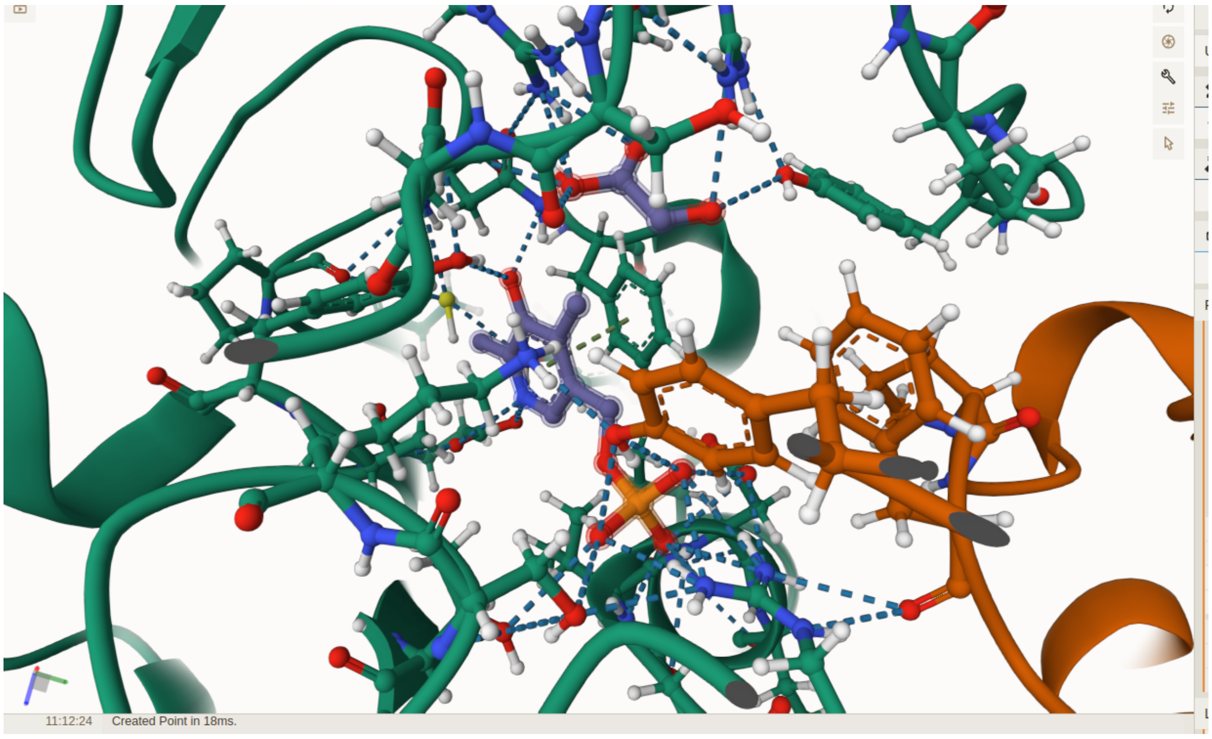


Now, let's perform docking on “A0AK37_relaxed.pdb” !

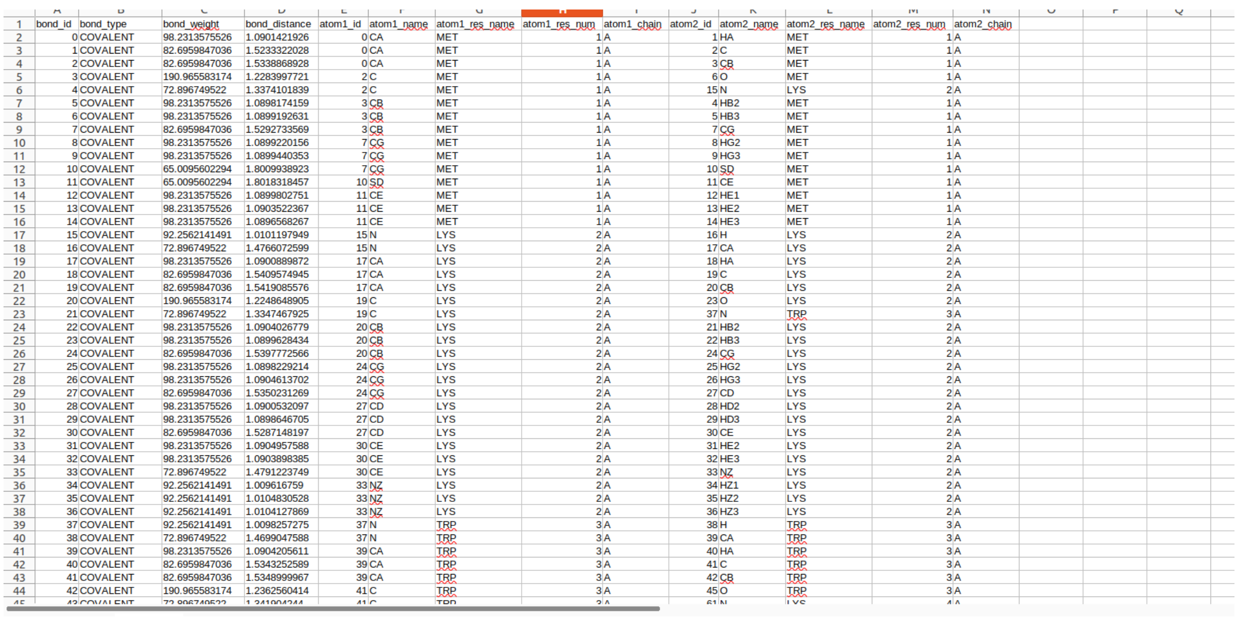
Now that we have docked the cofactor, we need to dock : the “donor” in the first complex then the “acceptor”.
First, use : Cofactor_Tag() to write the COF name in the pdb file (important is the future)
So, let’s just use the same functions : best_pocket_docking() on our ligand file. Here, we will not do the docking on both, but it is the same principle.

(to obtain the complex : enzyme + cofactor, use : create complex function).

Here the result : the PLP, at the right, the ligand at the left, in a very tiny pocket.



Here you can see the Lysine chain, the cofactor, and the ligand docked, and covalent interactions.
Finally lets use : Use : Cofactor_Tag() to write the LIG name in the pdb file. If not, PDB TO GRAPH will erase all the atoms with ‘UNK', 'UNL', 'UNX'
Now, let's model those interactions with PDB_to_GRAPH(complex_file,complex_name). It gave the interaction matrix of the complex, using BagPype :
https://github.com/yalirakilab/BagPype



	bond_id	bond_type	bond_weight	bond_distance	atom1_id	atom1_name	atom1_res_name	atom1_res_num	atom1_chain	atom2_id	atom2_name	atom2_res_name	atom2_res_num	atom2_chain
2	0	COVALENT	98.2313575526	1.0901421926	0	CA	MET	1	A	1	HA	MET	1	A
3	1	COVALENT	82.6959847036	1.5233322028	0	CA	MET	1	A	2	C	MET	1	A
4	2	COVALENT	82.6959847036	1.5338868928	0	CA	MET	1	A	3	CB	MET	1	A
5	3	COVALENT	190.965583174	1.2283997721	2	C	MET	1	A	6	O	MET	1	A
6	4	COVALENT	72.896749522	1.3374101839	2	C	MET	1	A	15	N	LYS	2	A
7	5	COVALENT	98.2313575526	1.0898174159	3	CB	MET	1	A	4	HB2	MET	1	A
8	6	COVALENT	98.2313575526	1.0899192631	3	CB	MET	1	A	5	HB3	MET	1	A
9	7	COVALENT	82.6959847036	1.5292733569	3	CB	MET	1	A	7	CG	MET	1	A
10	8	COVALENT	98.2313575526	1.0899220156	7	CG	MET	1	A	8	HG2	MET	1	A
11	9	COVALENT	98.2313575526	1.0899440353	7	CG	MET	1	A	9	HG3	MET	1	A
12	10	COVALENT	65.0095602294	1.8009938923	7	CG	MET	1	A	10	SD	MET	1	A
13	11	COVALENT	65.0095602294	1.8018318457	10	SD	MET	1	A	11	CE	MET	1	A
14	12	COVALENT	98.2313575526	1.0899802751	11	CE	MET	1	A	12	HE1	MET	1	A
15	13	COVALENT	98.2313575526	1.0903522367	11	CE	MET	1	A	13	HE2	MET	1	A
16	14	COVALENT	98.2313575526	1.0896568267	11	CE	MET	1	A	14	HE3	MET	1	A
17	15	COVALENT	92.2562141491	1.0101197949	15	N	LYS	2	A	16	H	LYS	2	A
18	16	COVALENT	72.896749522	1.4766072599	15	N	LYS	2	A	17	CA	LYS	2	A
19	17	COVALENT	98.2313575526	1.0900889872	17	CA	LYS	2	A	18	HA	LYS	2	A
20	18	COVALENT	82.6959847036	1.5409574945	17	CA	LYS	2	A	19	C	LYS	2	A
21	19	COVALENT	82.6959847036	1.5419085576	17	CA	LYS	2	A	20	CB	LYS	2	A
22	20	COVALENT	190.965583174	1.2248648905	19	C	LYS	2	A	23	O	LYS	2	A
23	21	COVALENT	72.896749522	1.3347467925	19	C	LYS	2	A	37	N	TRP	3	A
24	22	COVALENT	98.2313575526	1.0904026779	20	CB	LYS	2	A	21	HB2	LYS	2	A
25	23	COVALENT	98.2313575526	1.0899628434	20	CB	LYS	2	A	22	HB3	LYS	2	A
26	24	COVALENT	82.6959847036	1.5397772566	20	CB	LYS	2	A	24	CG	LYS	2	A
27	25	COVALENT	98.2313575526	1.0898229214	24	CG	LYS	2	A	25	HG2	LYS	2	A
28	26	COVALENT	98.2313575526	1.0904613702	24	CG	LYS	2	A	26	HG3	LYS	2	A
29	27	COVALENT	82.6959847036	1.5350231269	24	CG	LYS	2	A	27	CD	LYS	2	A
30	28	COVALENT	98.2313575526	1.0900532097	27	CD	LYS	2	A	28	HD2	LYS	2	A
31	29	COVALENT	98.2313575526	1.0898646705	27	CD	LYS	2	A	29	HD3	LYS	2	A
32	30	COVALENT	82.6959847036	1.5287148197	27	CD	LYS	2	A	30	CE	LYS	2	A
33	31	COVALENT	98.2313575526	1.0904957588	30	CE	LYS	2	A	31	HE2	LYS	2	A
34	32	COVALENT	98.2313575526	1.0903898385	30	CE	LYS	2	A	32	HE3	LYS	2	A
35	33	COVALENT	72.896749522	1.4791223749	30	CE	LYS	2	A	33	NZ	LYS	2	A
36	34	COVALENT	92.2562141491	1.009616759	33	NZ	LYS	2	A	34	HZ1	LYS	2	A
37	35	COVALENT	92.2562141491	1.0104830528	33	NZ	LYS	2	A	35	HZ2	LYS	2	A
38	36	COVALENT	92.2562141491	1.0104127869	33	NZ	LYS	2	A	36	HZ3	LYS	2	A
39	37	COVALENT	92.2562141491	1.0098257275	37	N	TRP	3	A	38	H	TRP	3	A
40	38	COVALENT	72.896749522	1.4699047588	37	N	TRP	3	A	39	CA	TRP	3	A
41	39	COVALENT	98.2313575526	1.0904205611	39	CA	TRP	3	A	40	HA	TRP	3	A
42	40	COVALENT	82.6959847036	1.5343252589	39	CA	TRP	3	A	41	C	TRP	3	A
43	41	COVALENT	82.6959847036	1.5348999967	39	CA	TRP	3	A	42	CB	TRP	3	A
44	42	COVALENT	190.965583174	1.2362560414	41	C	TRP	3	A	45	O	TRP	3	A

We can convert it to Pytorch Geometric / Networkx X graph with : Reaction_GRAPH()
There is still work to do on this function, but it works (look at the program)...

Finally, we succeed to medelise the interaction graph of a transaminase reacting on a Ping Pong reaction !

Congratulation !