

Autodock Vina tutorial

Master Bioinformática Universidad de Valencia
13-05-2013

Francisco Martínez Jiménez
fmartinez@pcb.ub.es

Hands on...

- We will :
 - **Predict** the binding site for a given protein structure.
 - Using Metapocket2 .
 - **Dock** a small molecule (inhibitor) into the predicted binding site.
 - Using Autodock Vina.
 - **Visualize** the ligand POSE. Creating a complex with the inhibitor and the protein.

https://www.dropbox.com/s/96xrk0vy6s0658e/valencia_docking.tar.gz

First steps...

- Visualize the protein structure with pymol.


```
>pymol /path_to_directory/receptor.pdb
```

- Show the structure as cartoons. (With secondary structure representation). In Pymol.

```
show as -> cartoon.
```

- Is there a putative binding pocket?

Predicting the binding



metaPocket 2.0

[Home](#) [Features](#) [Algorithm](#) [Benchmark](#) [Help](#) [About](#)

You might like our other meta servers:
[metaDBSite](#) [metaPPI 2.0](#)

Welcome to metaPocket 2.0 !

Program Arguments

You can enter a PDB ID and its chain ID:

PDB ID: (a four-letter PDB ID, e.g. 1dwd) ?

Chain ID: (separate many IDs by commas, e.g. E,I) ?

Or select a pdb file of your own to upload:

Select file to upload: No file chosen ?


How many pockets do you want:

Number of pockets: ?

Please enter your email address here, if you want us to send the result link to you:


E-mail address(optional):

Note:



- Before using metaPocket, please read our [help](#) page.
- The algorithm of metaPocket is illustrated [here](#).
- It will take up 400 seconds to find pockets depending on the size of your protein and the parameters you set.

Please be patient.



MetaPocket News

- 2011-06-04
- 2011-4-05
- 2011-4-01

Our metaPocket2.0 paper has been published in Bioinformatics! See the citation in the footnote!

The three datasets which were used to test metaPocket 2.0 are now available to public download, please refer to our [benchmark](#) page.

We submit the manuscript of metaPocket 2.0 to Bioinformatics journal !

Please be patient:

- It will take up 400 seconds to find pockets depending on the size of your protein and the parameters you set.
- The algorithm of metaPocket is illustrated [here](#).
- Before using metaPocket, please read our [help](#) page.

Retrieving the top predicted binding site...

4. Potential ligand binding sites

[HELP](#)

The potential 1 ligand binding sites in your protein:

HEADER binding site ID: 1

RESI	ILE_A^203^	PRO_A^205^	GLY_A^206^	LEU_A^207^	GLY_A^208^
RESI	TYR_A^202^	LEU_A^204^	ASP_A^209^	MET_A^142^	ASN_A^198^
RESI	ALA_A^210^	GLY_A^201^	ALA_A^144^	GLY_A^211^	LYS_A^200^
RESI	ALA_A^81^	PRO_A^141^	ARG_A^80^	THR_A^145^	ARG_A^105^
RESI	ASP_A^140^	ILE_A^143^	VAL_A^113^	MET_A^117^	LEU_A^79^
RESI	PHE_A^215^	ALA_A^146^	THR_A^148^	LYS_A^111^	GLU_A^112^
RESI	GLY_A^110^	SER_A^147^	GLU_A^107^	ASP_A^109^	ILE_A^78^
RESI	ALA_A^103^	PRO_A^114^	VAL_A^108^	MET_A^149^	TYR_A^122^
RESI	SER_A^104^	VAL_A^119^	ASP_A^116^	VAL_A^106^	ASP_A^118^
RESI	GLY_A^102^	ILE_A^101^	ASN_A^199^		

Download files of potential binding sites:

[A python script to visualize the protein structure and potential binding sites using PyMOL.](#)

(**Note: Please make sure to download all the following files to the same folder before to run this script.)

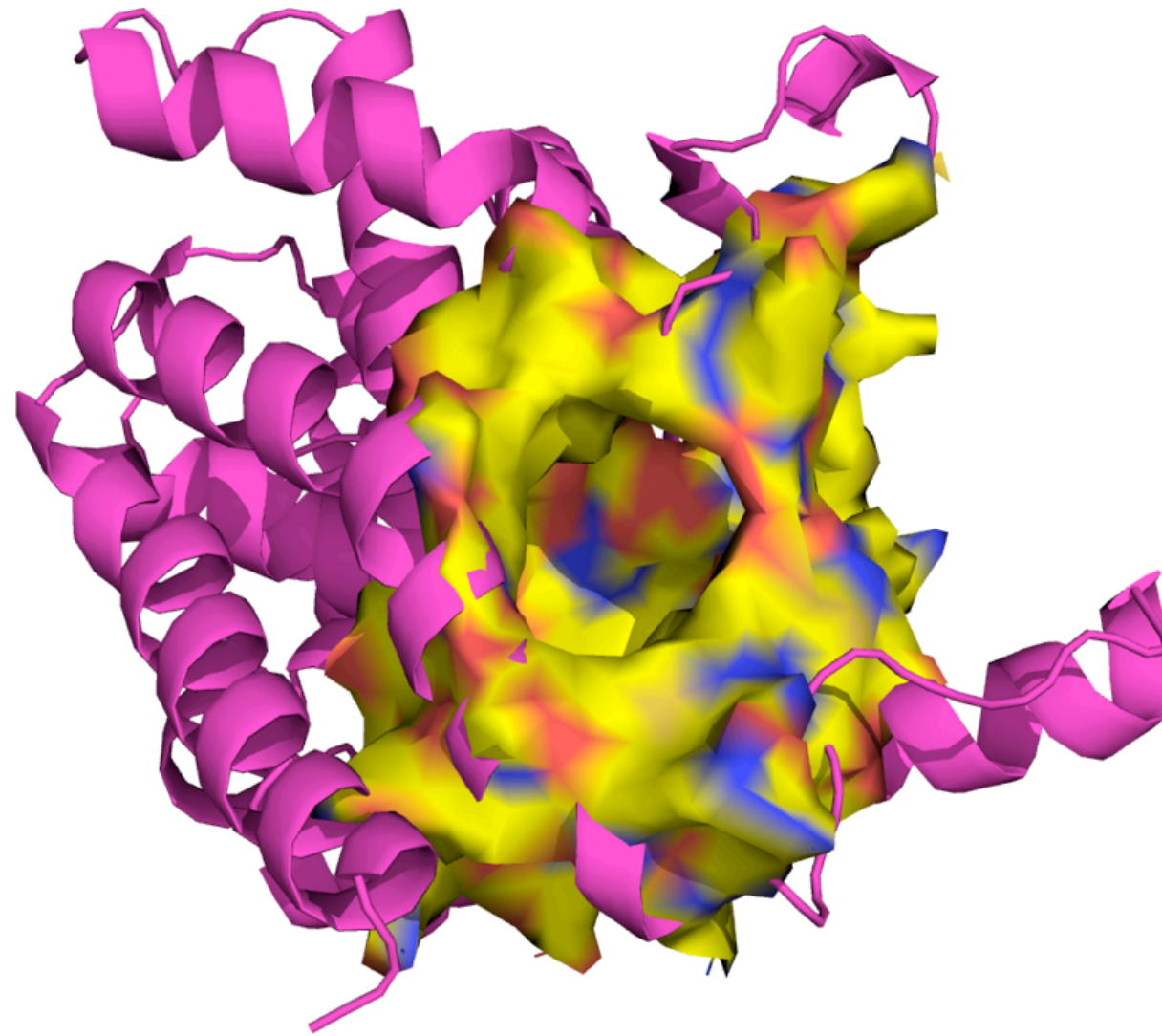
[The potential binding atoms of top 1 binding sites \(PDB format\).](#)

[The potential binding atoms of all the binding sites \(PDB format\).](#)

[The potential binding residues of top 1 binding sites \(PDB format\).](#)

[The potential binding residues of all the binding sites \(PDB format\).](#)

Visualizing predicted binding site + protein structure



Getting the binding site Center of Mass

- In pymol there is a plugin that calculates the CoM of an certain region.

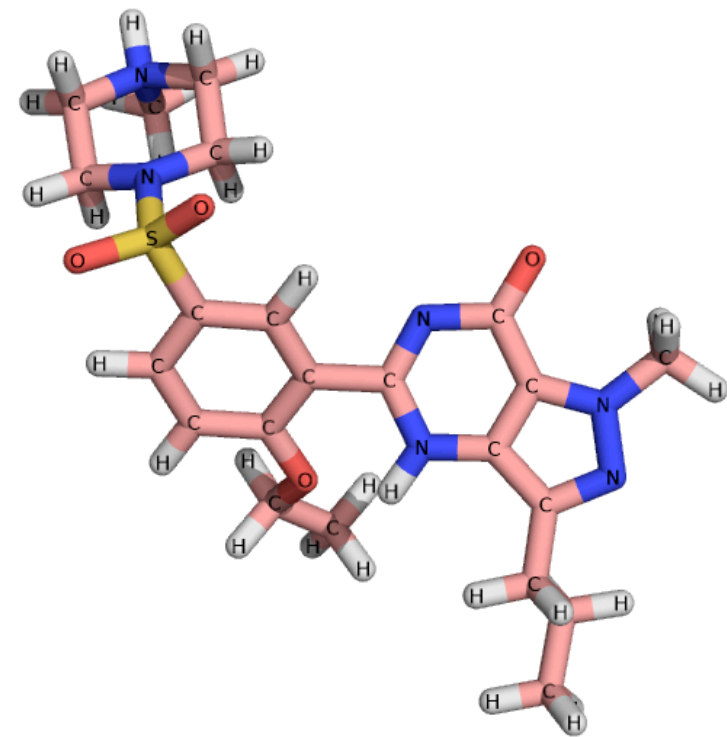
[http://www.pymolwiki.org/index.php/Center of mass](http://www.pymolwiki.org/index.php/Center_of_mass)

- In our case, the plugin is not installed but the point is :

X: 28.035454 Y:119.400581 Z:9.578382

Exploring the ligand...

- Visualize the ligand in pymol.
- File -> open -> ligand.mol2
- Show as -> sticks.
- Label -> Atom name.



Preparing the ligand for docking...

- Open Autodock Tools.
- Ligand -> Input -> Open -> ligand.mol2
- Ligand -> Torsion Tree -> Choose Torsion (Default).
- Save the *.pdbqt file in your working directory.
- **Remember!**, we have to measure the ligand size.
- Display -> Measure -> Distance. ($\sim 15 \text{ \AA}$).

Preparing the receptor...

- Grid -> macromolecule -> Open -> receptor.pdb
- Save the *.pdbqt file in your working directory.
- Visualize as ribbons.
- Grid center: Center of Mass.
- Grid -> Grid Box.

X: 28.035454 Y:119.400581 Z:9.578382

- Grid size : double of the ligand size. (~ 30 Å).

Setting the config file for vina...

```
receptor = path_to_receptor.pdbqt
ligand = path_to_ligand.pdbqt
out= path_to_out_file.pdbqt
log= path_to_log_file.txt
center_x= X
center_y = Y
center_z= Z
```

```
num_models = Num_models
exhaustiveness= 8
```

```
size_x = 30
size_y = 30
size_z = 30
```

```
>vina --help_advanced
```

```
fran@davide-desktop:~/Documents/master_valencia$ ./vina --help_advanced
Input:
  --receptor arg      rigid part of the receptor (PDBQT)
  --flex arg          flexible side chains, if any (PDBQT)
  --ligand arg         ligand (PDBQT)

Search space (required):
  --center_x arg      X coordinate of the center
  --center_y arg      Y coordinate of the center
  --center_z arg      Z coordinate of the center
  --size_x arg         size in the X dimension (Angstroms)
  --size_y arg         size in the Y dimension (Angstroms)
  --size_z arg         size in the Z dimension (Angstroms)

Output (optional):
  --out arg            output models (PDBQT), the default is chosen based on
                        the ligand file name
  --log arg             optionally, write log file

Advanced options (see the manual):
  --score_only          score only - search space
                        can be omitted
  --local_only          do local search only
  --randomize_only      randomize input, attempting
                        to avoid clashes
  --weight_gauss1 arg (=0.035579)  gauss_1 weight
  --weight_gauss2 arg (=0.005156)  gauss_2 weight
  --weight_repulsion arg (=0.8402450000000000002)  repulsion weight
  --weight_hydrophobic arg (=0.03506900000000000003)  hydrophobic weight
  --weight_hydrogen arg (=0.5874390000000000004)  Hydrogen bond weight
  --weight_rot arg (=0.0584599999999999998)  N_rot weight

Misc (optional):
  --cpu arg             the number of CPUs to use (the default is to try to
                        detect the number of CPUs or, failing that, use 1)
  --seed arg            explicit random seed
  --exhaustiveness arg (=8)  exhaustiveness of the global search (roughly
                        proportional to time): 1+
  --num_modes arg (=9)    maximum number of binding modes to generate
  --energy_range arg (=3)  maximum energy difference between the best binding
                        mode and the worst one displayed (kcal/mol)

Configuration file (optional):
  --config arg          the above options can be put here

Information (optional):
  --help                display usage summary
  --help_advanced        display usage summary with advanced options
  --version              display program version

  --version              display program version
  --help                 display usage summary
  --help_advanced         display usage summary with advanced options
  --version                display program version
```

Run Vina!!

```
> vina --config=/path_to_config_file/config.txt
```

```
fran@davide-desktop:~/Documents/master_valencia$ ./vina --config=config 2.txt
```

```
#####  
# If you used AutoDock Vina in your work, please cite: #  
# O. Trott, A. J. Olson, #  
# AutoDock Vina: improving the speed and accuracy of docking #  
# with a new scoring function, efficient optimization and #  
# multithreading, Journal of Computational Chemistry 31 (2010) #  
# 455-461 #  
# DOI 10.1002/jcc.21334 #  
# Please see http://vina.scripps.edu for more information. #  
#####
```

WARNING: The search space volume > 27000 Angstrom^3 (See FAQ)

Detected 4 CPUs

Reading input ... done.

```
Setting up the scoring function ... done.
```

```
Analyzing the binding site ... done.
```

Using random seed: -1305734982

Performing search ...

0% 10 20 30 40 50 60 70 80 90 100%

© 2006 Pearson Education, Inc. All rights reserved.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Analyze the output...

>more log_file.txt

```
#####
# If you used AutoDock Vina in your work, please cite: #
# lev/sda3 4396 3726 636 86% /home #
# O. Trott, A. J. Olson, 76 4016 4706 47% /media/c15113b0-9369 #
# AutoDock Vina: improving the speed and accuracy of docking #
# with a new scoring function, efficient optimization and #
# multithreading, Journal of Computational Chemistry 31 (2010) #
# 455-461 #
# #
# DOI 10.1002/jcc.21334 #
# #
# Please see http://vina.scripps.edu for more information. #
#####

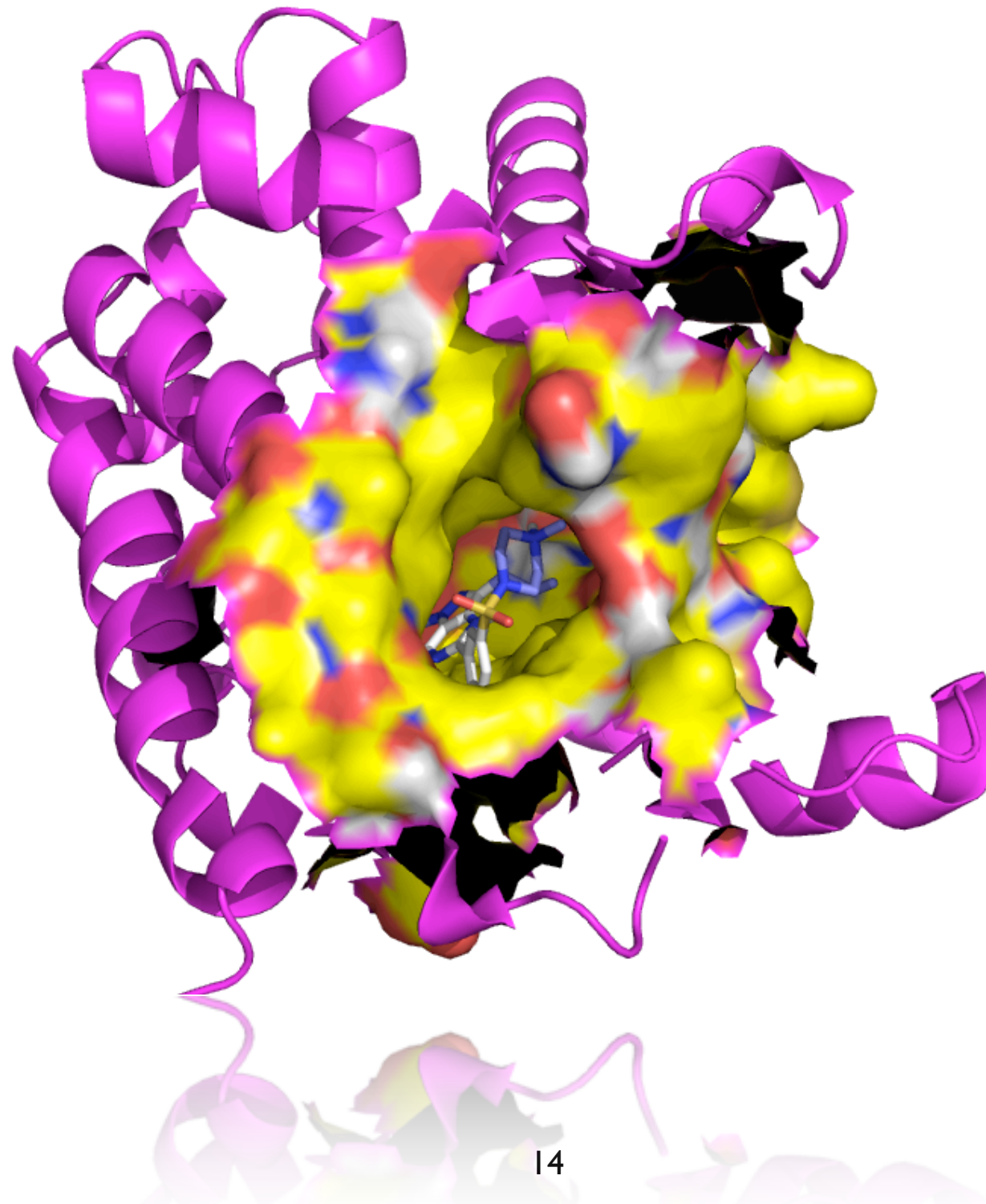
WARNING: The search space volume > 27000 Angstrom^3 (See FAQ)
Detected 4 CPUs
Reading input ... done.
Setting up the scoring function ... done.
Analyzing the binding site ... done.
Using random seed: -1305734982
Performing search ... done.
Refining results ... done.

mode |  affinity | dist from best mode
      | (kcal/mol) | rmsd l.b. | rmsd u.b.
-----+-----+-----+-----
  1   |    -9.4   |    0.000   |    0.000
  2   |    -9.3   |    1.716   |    2.266
  3   |    -9.1   |    4.545   |    8.194
  4   |    -9.1   |    6.205   |    8.979
  5   |    -9.1   |    3.806   |    8.467
  6   |    -9.0   |    3.756   |    7.807
  7   |    -8.9   |    2.479   |    4.540
  8   |    -8.9   |    2.492   |    3.125
  9   |    -8.8   |    2.300   |    3.561
 10   |    -8.7   |    8.711   |   11.420
 11   |    -8.7   |    1.843   |    2.343
 12   |    -8.6   |    4.321   |    8.198
 13   |    -8.6   |    2.421   |    3.471
 14   |    -8.6   |    2.246   |    3.591
 15   |    -8.4   |    5.066   |    8.604
 16   |    -8.4   |    2.579   |    3.888
 17   |    -8.0   |    2.929   |    4.943
 18   |    -7.9   |    2.722   |    4.951

Writing output ... done
 18   |    -7.9   |    2.722   |    4.951
 11   |    -8.7   |    1.843   |    2.343
 10   |    -8.7   |    8.711   |   11.420
 12   |    -8.6   |    4.321   |    8.198
 13   |    -8.6   |    2.421   |    3.471
 14   |    -8.6   |    2.246   |    3.591
 15   |    -8.4   |    5.066   |    8.604
 16   |    -8.4   |    2.579   |    3.888
 17   |    -8.0   |    2.929   |    4.943
 18   |    -7.9   |    2.722   |    4.951
```

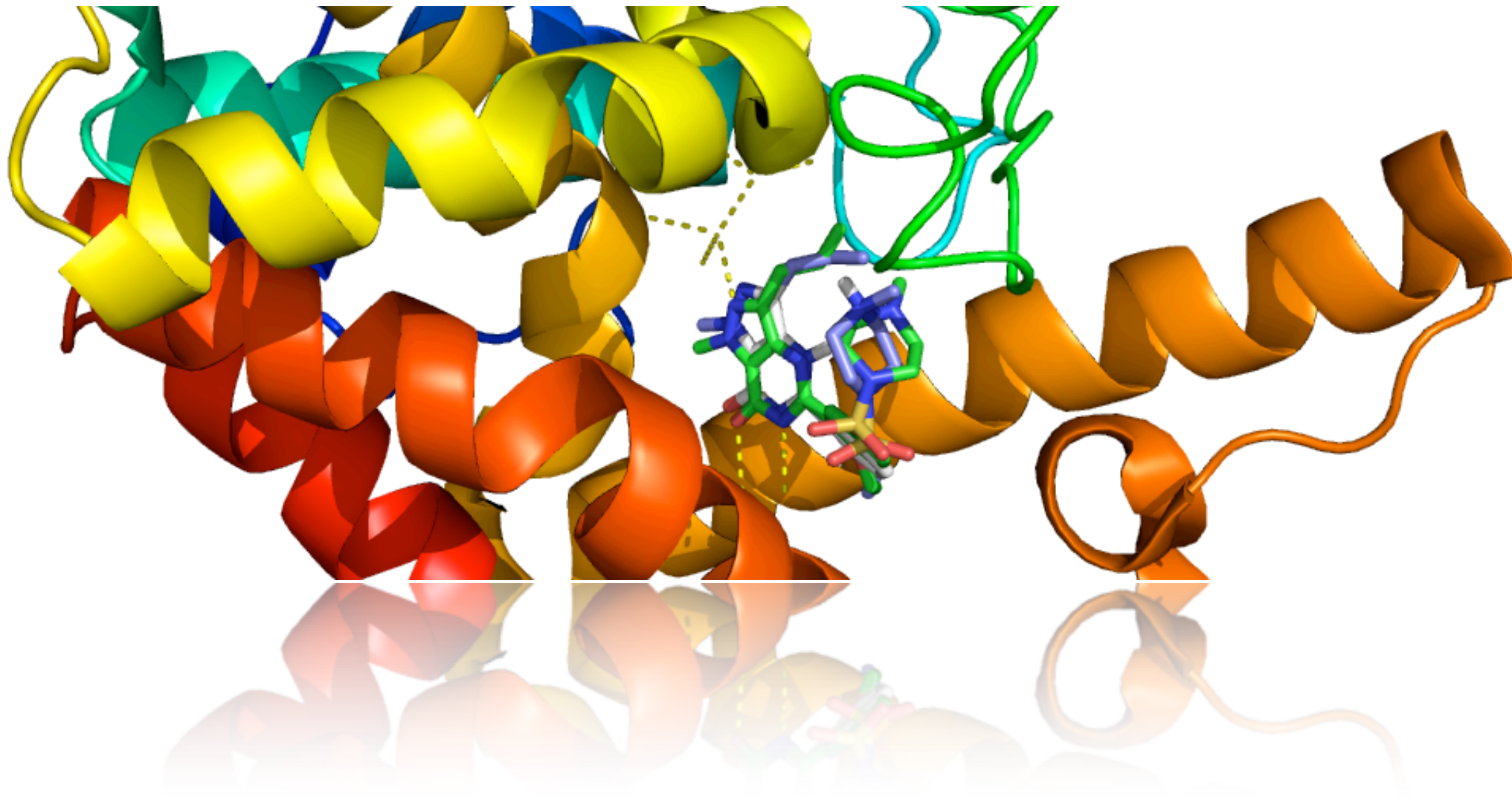

Visualize the output...

```
>pymol receptor.pdb out_docking.pdbqt
```



How good are our solutions?

pymol - Open -> Solutions/receptor_ligand.pdb



But... What is our complex?

Let's figure it out !

:)