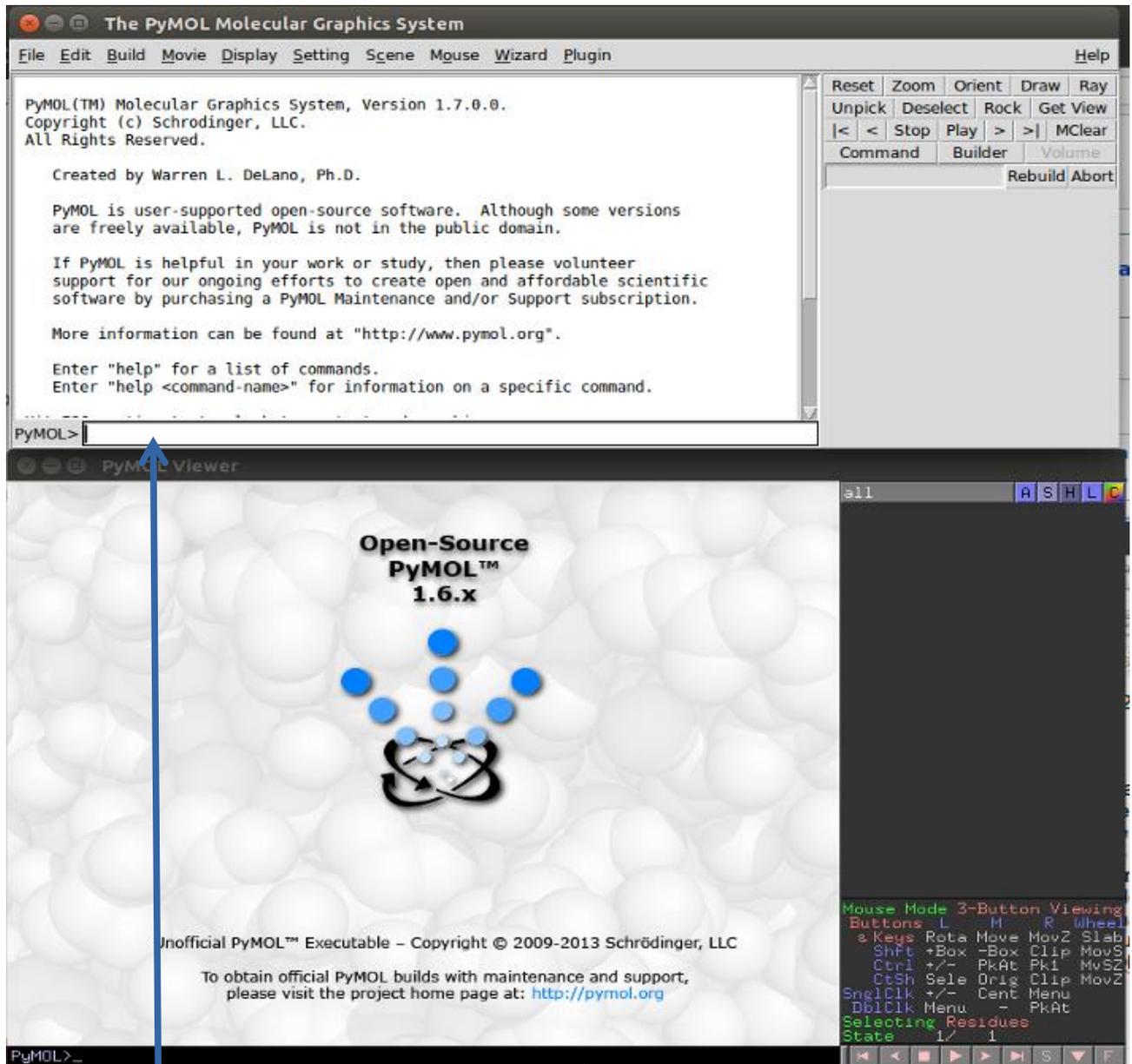


Tutorial – Visualização Gráfica com Pymol e Docking com Vina

Aula Prática – 22/10/2020

Interface do Pymol



Linha de comando

Menu de comandos:



A = Action, S = Show, H = Hide, L = Label, C = Color

Comandos básicos:

Para arrastar – clique com o botão do meio do mouse e arraste.

Para alterar o zoom – clique com o botão direito do mouse e arraste para cima (diminuir zoom) ou para baixo (aumentar zoom).

Para girar – clique com o botão esquerdo do mouse e arraste.

Para “cortar” a exibição de parte da imagem (em níveis de profundidade) – role o botão do meio do mouse.

Como executar comandos no Pymol?

Em geral, os comandos podem ser executados a partir do painel de comandos, utilizando os menus descritos acima, ou a partir da linha de comando. Para utilizar a linha de comando, digite o comando e **dê enter**. Neste tutorial utilizaremos tanto a linha de comando quanto o painel de comandos.

Como carregar uma nova molécula no Pymol

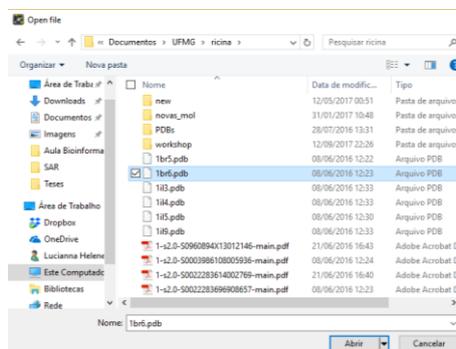
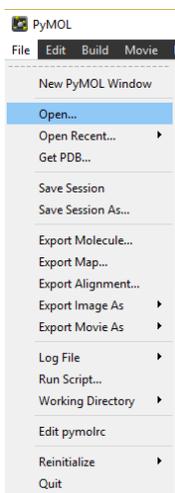
A) Comando fetch <PDB> . Exemplo: fetch 1br6.

Explicação: Faz o download de uma estrutura diretamente do site do PDB.

B) Comando load <PDB> . Exemplo: load 1br6.pdb.

Explicação: Carrega uma estrutura de um arquivo local.

C) File > Open.



Explicação: Abre uma estrutura PDB navegando as pastas no computador.

Parte 1 – Explorando opções de representação e preparando uma figura no pymol

- 1) Abra a estrutura PDB 1L2S, um complexo da enzima β -lactamase com um inibidor descoberto através de triagem virtual de bases de dados:

```
PyMOL> fetch 1L2S
```

- 2) Inicialmente estarão representados dois monômeros, conforme observado na unidade assimétrica. Trabalharemos apenas com a cadeia A. Primeiramente, vamos visualizar as duas cadeias antes de deletar a cadeia B.

Vamos representar a proteína como cartoon:

```
PyMOL> hide lines  
PyMOL> show cartoon
```

Para distinguir melhor as duas cadeias, altere a cor da cadeia B para ciano:

```
PyMOL> color cyan, chain B
```

Para remover a cadeia B:

```
PyMOL> select chain B  
PyMOL> remove sele
```

Pronto! Agora temos apenas a cadeia A.

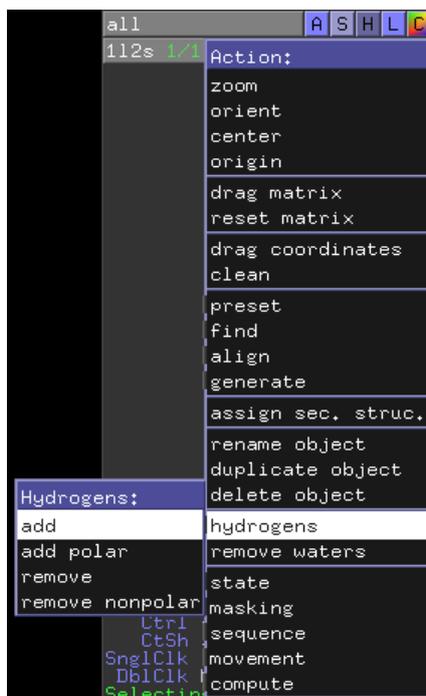
- 3) Vamos explorar diferentes representações da estrutura secundária, alterando a forma de representação de hélices e folhas-beta. A cada comando abaixo, observe as modificações:

```
PyMOL> set cartoon_flat_sheets, 0  
PyMOL> set cartoon_smooth_loops, 0  
PyMOL> set cartoon_fancy_helices, 1  
PyMOL> set cartoon_discrete_colors, 1  
PyMOL> set cartoon_highlight_color, 1
```

- 4) Exiba a estrutura da proteína na forma de bastões e as águas como esferas, e esconda a representação em cartoon:

```
PyMOL> show sticks  
PyMOL> show nb_spheres  
PyMOL> hide cartoon
```

- 5) Adicione hidrogênios (Estruturas PDB originadas de cristalografia normalmente não possuem hidrogênios).



6) Crie um objeto contendo apenas o inibidor e um contendo o inibidor e as moléculas de água ou resíduos de proteína a no máximo 6 Å de distância do composto.

- para extrair um objeto nomeado **ligante**, contendo a molécula nomeada **STC** no arquivo PDB:

PyMOL> **extract ligante, resn STC**

- a partir do objeto anteriormente criado (**ligante**), para criar um objeto nomeado **sitio_ligante**, que contenha o objeto anterior e todos os átomos a até 6 Å de distância dos átomos contidos neste objeto:

PyMOL> **create sitio_ligante, ligante around 6**

7) Represente a superfície da proteína, a partir do objeto 1L2S.

PyMOL> **show surface, 1L2S**

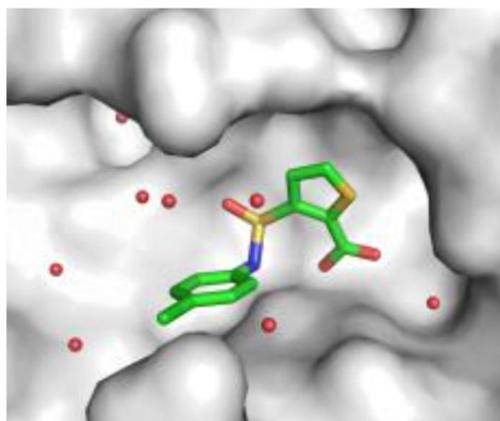
PyMOL> **color gray, 1L2S**

8) Mude o fundo para branco e amplie na região do ligante.

PyMOL> **bg_color white**

PyMOL> **zoom sitio_ligante**

Você deverá obter uma imagem semelhante à figura abaixo (se necessário, utilize o mouse para dar zoom e alterar a orientação até obter uma imagem semelhante à mostrada abaixo):



9) Para melhorar a resolução da imagem, defina o tamanho da figura no momento de executar o comando ray:

PyMOL> **ray 640, 480** (define o tamanho da imagem, nas duas dimensões)

10) Salve a imagem com uma resolução específica:

PyMOL> **png 1L2S.png, dpi=300**

11) **Salve a sessão do Pymol** (menu File > Save as session) como **1L2S_parte1.pse**

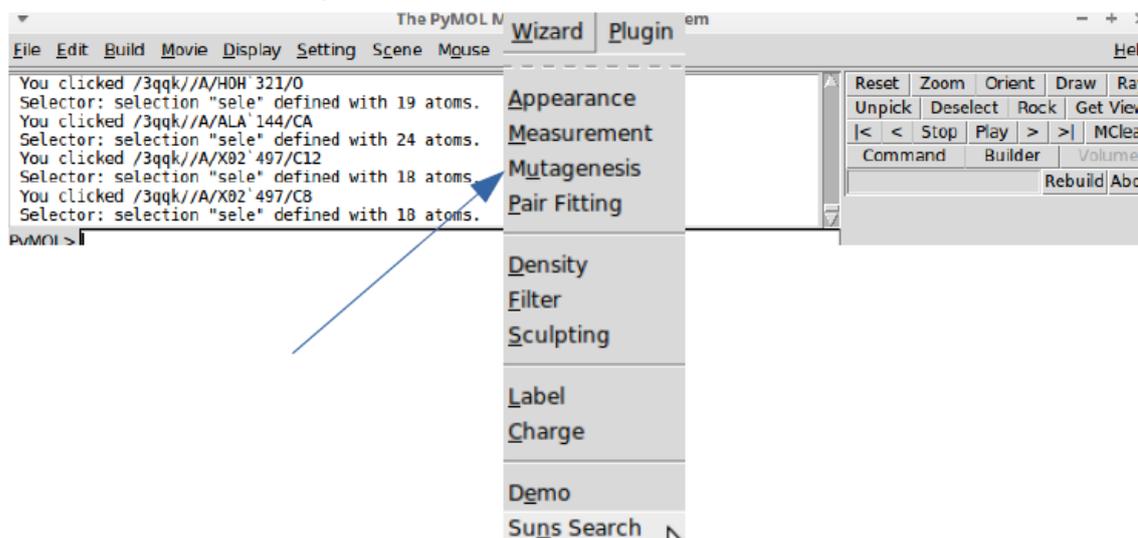
12) Vamos analisar as interações intermoleculares entre inibidor e enzima.

Calcule as interações entre átomos polares do inibidor e a enzima ou moléculas de água.

- Selecione o ligante no menu de seleção
- No menu **Actions** da seleção, clique em **Find – polar contacts – to any atoms**

13) Meça a distância entre os átomos envolvidos em cada uma das interações intermoleculares.

- No Menu **Wizard**, clique em **Measurement**.



Para medir as distâncias entre dois átomos, clique sucessivamente em cada um.

- 14) Vamos alterar as cores das linhas para visualizarmos melhor. Para isso, digite na linha de comando
PYMOL> **set dash_color, black**
- 15) Após medir todas as distâncias, clique em **Done**, no Menu **Measurement**, e salve como 1L2S_dist.pse.

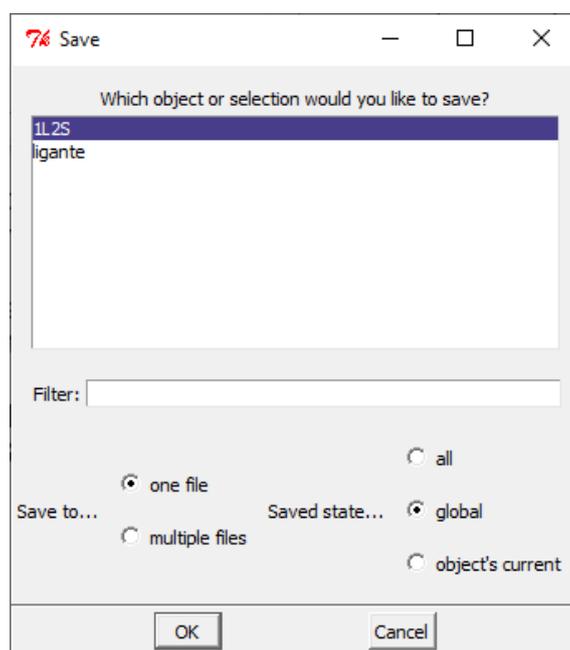
Parte 2 – Atracamento molecular utilizando o Autodock Vina

Para o tutorial de atracamento molecular vamos utilizar a metodologia de reprodução de pose, chamado também de **re-docking**. A metodologia consiste em tentar reproduzir a posição adquirida pelo ligante no cristal utilizando o programa de atracamento. Portanto, obtemos uma validação do protocolo e um teste do algoritmo de atracamento para o caso da proteína estudada. Ao final, podemos comparar interações e o RMSD entre pose e posição cristalográfica do ligante.

16) Para o atracamento vamos primeiro remover as moléculas de água:

PYMOL> **remove resn hoh**

17) Ainda com a estrutura 1L2S. Vamos salvar arquivos separados para o **receptor** e para o **ligante**. Em **File > Save Molecule**



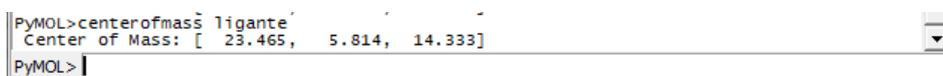
Dar o nome de **receptor** para a proteína.



Faça o mesmo para o **ligante**, salve como **ligante**.

18) O Vina não faz um pré-cálculo de Grid. Porém, uma demarcação da área de docking (sítio ativo ou até mesmo a proteína toda) precisa ser estabelecida. Como estamos usando uma estrutura que já possui o ligante podemos calcular o centro de massa do ligante para ser o centro da nossa Grid.

PYMOL> **centerofmass ligante**



Anote esses números. Vamos utilizá-los logo adiante.

19) Podemos fechar o Pymol não vamos precisar dele para os próximos passos. A preparação para o docking será feita por linha de comando usando o servidor. Já que o programa de docking que vamos utilizar, Vina, foi projetado para não ter uma interface e com isso ser um programa mais leve.

20) Para preparar os arquivos temo que ACESSAR O SERVIDOR no terminal de vocês:

```
$ ssh bioufmg@bioinfo.icb.ufmg.br
```

DIGITE A SENHA.

21) Encontre a sua pasta (com seu nome):

```
lucianna@DESKTOP-JQF7JLS:~$ ssh bioufmg@bioinfo.icb.ufmg.br
bioufmg@bioinfo.icb.ufmg.br's password:
Last login: Wed Oct 14 17:09:15 2020 from 201.17.140.105
[bioufmg@bioinfo ~]$ ls
INTRO          elisaamancio   larama         milenap
ace222         eusoujacu     larissa_cler   monic_lops
amandacpa     flavia         laryhenriques moysesmn
anna_menezes  flaviam       letbarbosa    nathalia_alv
arthurtrcf    franciscoacarmo letchris       pedrobala
barbaramas    fredericogabriel lucasduque     rayssa_soares
beatrizapgaua gabijager     luizacaixeta  renan
caizin        gabriel_ferrari macl3y        tarefa1
camilabd      gabrielrod    madumascarenhas tarefa2
camilacllc    gapmo         mari_1305     teste
ceciliahorta jessica_alves marinak        thiagoac
clecio_alonso jorge         markko
danikayali    karenkanda    mbarcelos
dg_sousa      key           michelle_157
[bioufmg@bioinfo ~]$ _
```

*se não possuir pasta. Crie uma com o comando `mkdir <seunome>`

22) Acesse sua pasta com

```
$ cd minhapasta
```

23) Confira se você está realmente na pasta com o seu nome:

```
$ pwd
```

24) Dentro da sua pasta, crie uma nova pasta chamada `docking_pratica`:

```
$ mkdir docking_pratica
```

25) Acesse a pasta

```
$ cd docking_pratica
```

26) Copie o conteúdo da pasta `/home/treinamento/docking_pratica` para cá:

```
$ cp /home/treinamento/docking_pratica/* .
```

Pronto. Temos um diretório para rodar preparar os arquivos de entrada e rodar o Vina.

```
config.txt ligante.pdb receptor.pdb
```

- 27) Para o docking com o Vina precisamos dos arquivos de entrada no formato PDBQT. Esse formato é uma adaptação do formato PDB. Nele além das coordenadas atômicas temos informação de carga e descrição dos átomos, específico para os programas Vina e Autodock4.

Para criar o PDBQT do receptor basta:

```
$ pythonsh $mglscripsts/prepare_receptor4.py -r receptor.pdb
```

***pythonsh é uma versão (antiga) compilada do python para o pacote MGLTools. O caminho dele é:**

```
/opt/treinamento/mgltools_x86_64Linux2_1.5.7rc1/bin/python
```

**** \$mglscripsts é um PATH criado para não precisar digitar todo o caminho até o script. O caminho é:**

```
/opt/treinamento/mgltools_x86_64Linux2_1.5.7rc1/MGLToolsPkgs/AutoDockTools/Utilities24/prepare_receptor4.py
```

- 28) Para criar o PDBQT do ligante basta:

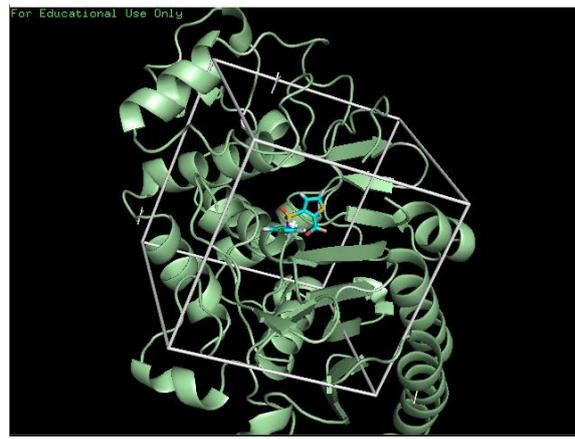
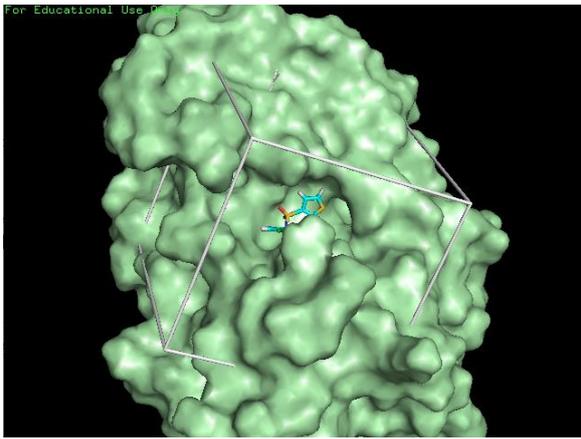
```
$ pythonsh $mglscripsts/prepare_ligand4.py -l ligante.pdb -U "
```

*** -U " pede para que o script mantenha todos os hidrogênios do ligante. Tirando essa "flag", apenas os hidrogênios polares permaneceriam.**

- 29) O último arquivo de entrada necessário é o arquivo com os parâmetros do Vina. É um simples arquivo de texto (vamos chamar de **config.txt**) contendo os seguintes parâmetros:

```
receptor = receptor.pdbqt      #Nome do arquivo PDBQT do receptor
ligand = ligante.pdbqt        #Nome do arquivo PDBQT do ligante
center_x = 23.58              #Coordenada X do centro da caixa (aqui o centro do ligante)
center_y = 5.62               #Coordenada Y do centro da caixa (aqui o centro do ligante)
center_z = 14.4               #Coordenada Z do centro da caixa (aqui o centro do ligante)
size_x = 25.00                #Tamanho da caixa para o eixo X
size_y = 25.00                #Tamanho da caixa para o eixo Y
size_z = 25.00                #Tamanho da caixa para o eixo Z
out = docked_ligante.pdbqt    #Nome do arquivo de saída com as poses do docking
log = docked.log              #Arquivo de log com informações da saída do docking
num_modes = 10                #Quantidade de poses
```

Center_x, center_y e center_z são as coordenadas do centro de massa do ligante calculados anteriormente no pymol. Isso quer dizer que o programa terá sua "região de busca" centrada nessas coordenadas. Para delimitar uma região que compreenda parte da proteína ou um sítio de interesse temos os parâmetros size_x, size_y e size_z, que no nosso caso, estende 25 Å para cada eixo desse ponto formando uma caixa cúbica nessa região da proteína. A região para docking será então:



30) Agora que temos todos os arquivos necessário, podemos rodar o Vina. Utilizamos o comando:

```
$ vina --config config.txt
```

A seguinte tela vai aparecer:

```
#####
# 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. #
#####

Detected 4 CPUs
Reading input ... done.
Setting up the scoring function ... done.
Analyzing the binding site ... done.
Using random seed: 1197525432
Performing search ...
0% 10 20 30 40 50 60 70 80 90 100%
|----|----|----|----|----|----|----|----|----|----|
|*****|
done.
Refining results ... done.

mode | affinity | dist from best mode
      | (kcal/mol) | rmsd l.b. | rmsd u.b.
-----|-----|-----|-----
1 | -7.1 | 0.000 | 0.000
2 | -7.0 | 1.863 | 2.220
3 | -6.7 | 1.960 | 2.204
4 | -6.5 | 3.607 | 6.281
5 | -6.4 | 3.239 | 5.258
6 | -6.4 | 1.283 | 2.747
7 | -6.3 | 1.616 | 1.915
8 | -6.2 | 12.030 | 14.052
9 | -6.2 | 2.125 | 2.942
10 | -6.1 | 1.994 | 3.017

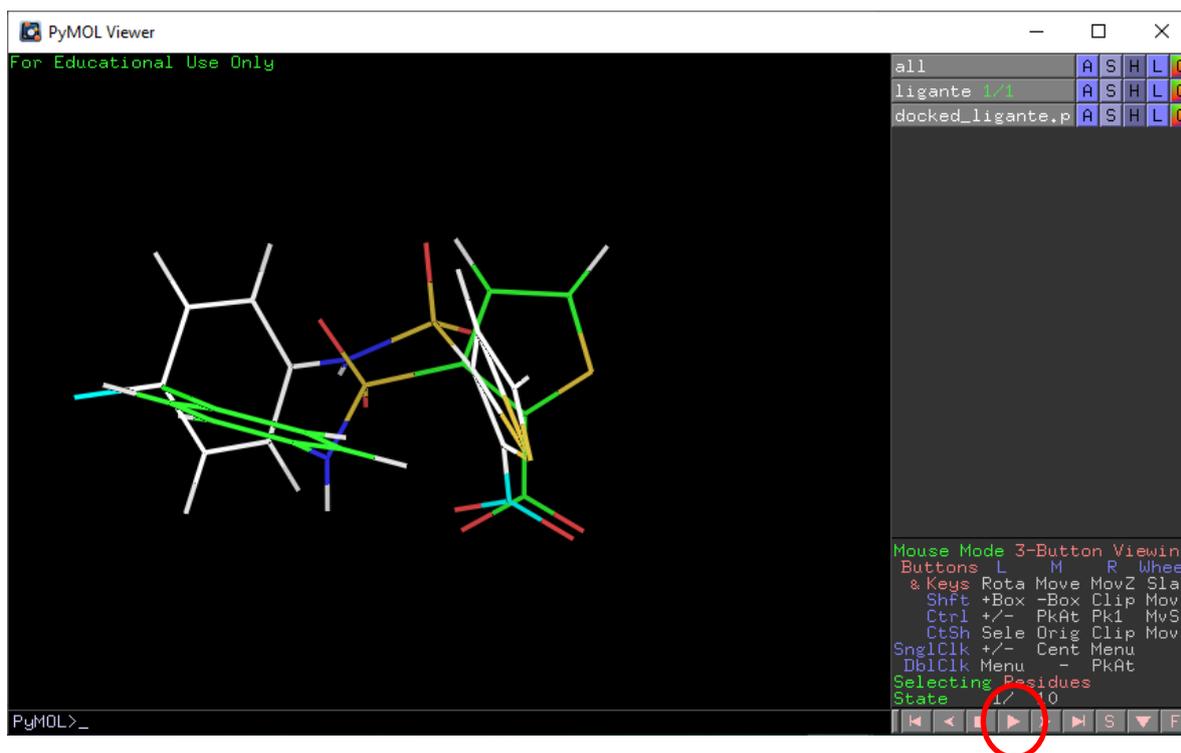
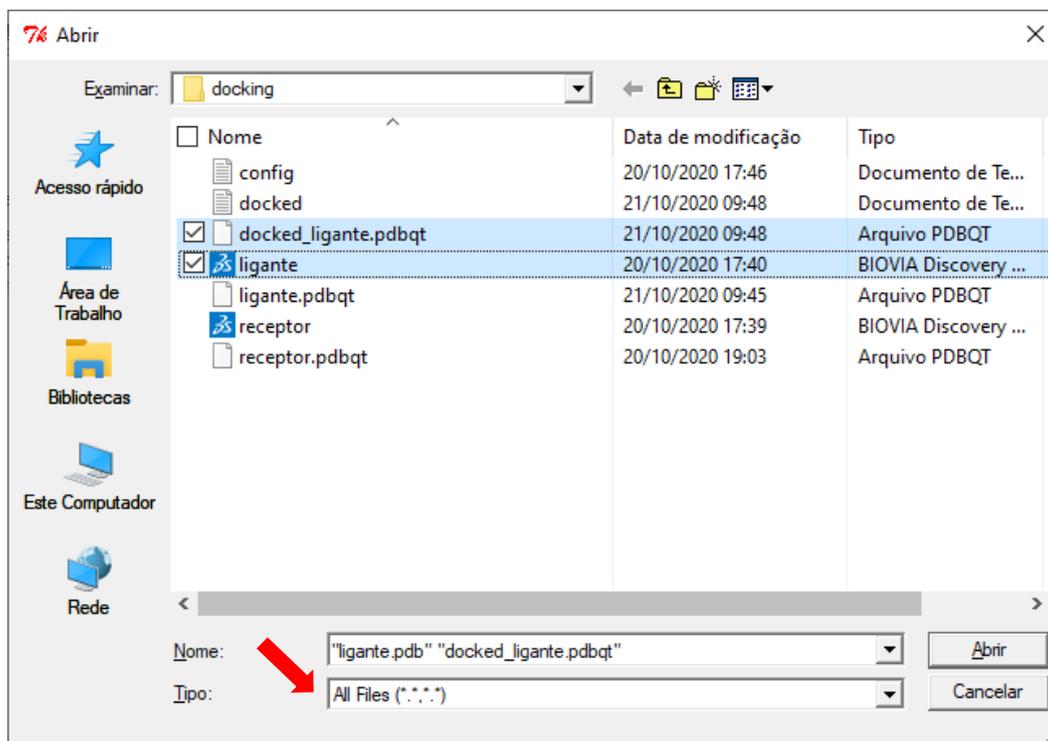
Writing output ... done.
```

* O programa Vina é aleatório e a cada rodada ele escolhe um número aleatório, dado pelo parâmetro chamado seed. Portanto, os resultados podem ser diferentes em cada rodada.

31) Vamos baixar o arquivo **docked_ligante.pdbqt** para o nosso computador e abrir pelo PYMOL, junto com o ligante cristalográfico **ligante.pdb**.

No PYMOL vamos em **File > Open** navegar até a pasta onde estão os arquivos:

O PDBQT não vai aparecer. Então mude o **Tipo** para **All Files**. E selecione os arquivos.



As poses abrem todos juntas na visualização docked_ligante.pdbqt. Para ir passando por elas clique no botão > no canto direito inferior.

Podemos também separar cada estado usando o comando split_states
 PYMOL> **split_states docked_ligante.pdbqt**


```
docked_ligante_ligand_01.pdbqt  
docked_ligante_ligand_02.pdbqt  
docked_ligante_ligand_03.pdbqt  
docked_ligante_ligand_04.pdbqt  
docked_ligante_ligand_05.pdbqt  
docked_ligante_ligand_06.pdbqt  
docked_ligante_ligand_07.pdbqt  
docked_ligante_ligand_08.pdbqt  
docked_ligante_ligand_09.pdbqt  
docked_ligante_ligand_10.pdbqt
```

E usando um script vamos calcular o RMSD ente ligante e pose:

```
$ pythonsh $mglscripsts/compute_rms_between_conformations.py -f  
ligante.pdq -s docked_ligante_ligand_02.pdbqt
```

Para o re-docking ser considerado um sucesso, a pose de melhor score (a primeira colocada) tem que alcançar o RMSD abaixo de 2.0 Å. No caso ilustrado a pose de menor score não atingiu esse critério. Porém, a segunda pose ranqueada conseguiu satisfazer esse critério. Com isso, tem que se ter cuidado para olhar o grupo de poses ranqueadas com cuidado para não descartar verdadeiros positivos.

OBS: A verificação de RMSD só pode ser feita quando temos um ligante cristal para comparar. Em um experimento de docking com ligantes novos esse parâmetro **não pode ser calculado**. Porém, as interações formadas entre proteína e ligante podem (e devem) ser analisadas e exploradas.