

Tutorial – Visualização Gráfica com Pymol e Docking com Vina

Aula Prática – 13/06/2019

Sobre o Pymol

PyMOL é um software de visualização molecular criado por Warren Lyford DeLano. É um programa open source, lançado sob a licença do Python. Ele foi comercializado inicialmente por DeLano científico LLC, que foi uma empresa privada de software dedicada à criação de ferramentas úteis que se tornam universalmente acessíveis às comunidades científicas e educacionais. Atualmente é comercializado pela Schrödinger, Inc. No PyMOL pode-se produzir imagens 3D de alta qualidade de pequenas moléculas e macromoléculas biológicas, tais como proteínas. De acordo com o autor original, até 2009, quase um quarto de todas as imagens publicadas de estruturas 3D de proteínas na literatura científica foram feitas utilizando PyMOL.

Wiki do Pymol

https://pymolwiki.org/index.php/Main_Page

Download Pymol

<https://pymol.org/>

Instalação em Linux (Ubuntu)

```
sudo apt-get install pymol
```

Sobre o Vina

AutoDock Vina é um programa de código aberto para fazer docking molecular. Ele foi projetado e implementado pelo Dr. Oleg Trott no Molecular Graphics Lab no The Scripps Research Institute. A filosofia de design do Vina é não exigir que o usuário entenda seus detalhes de implementação, ajuste parâmetros obscuros de busca, resultados de cluster ou saiba álgebra avançada. Tudo o que é necessário são as estruturas das moléculas a se encaixar e a especificação do espaço de pesquisa, incluindo o local de ligação.

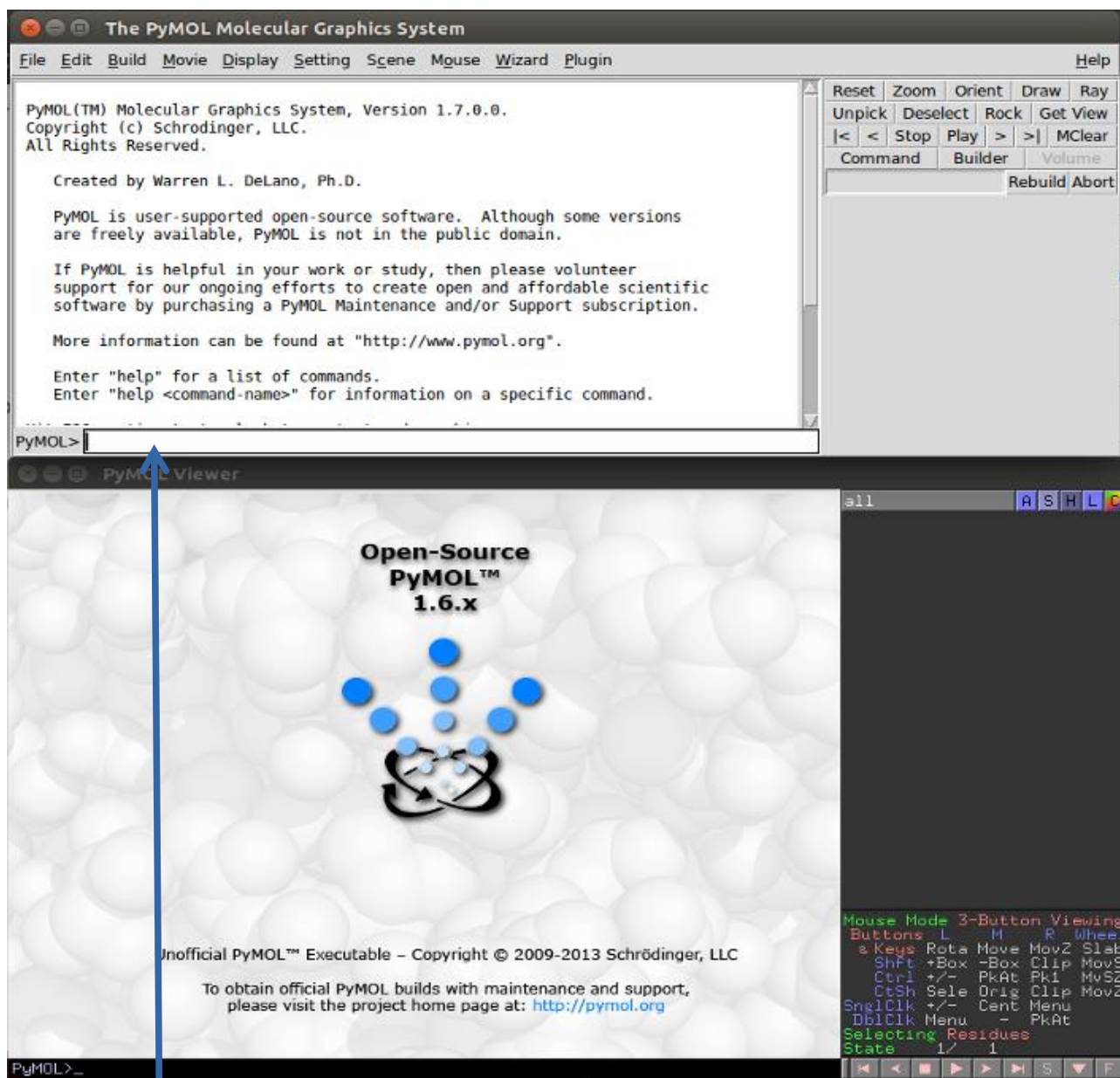
Wiki do Vina

<http://vina.scripps.edu/>

Download do Vina

<http://vina.scripps.edu/>

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

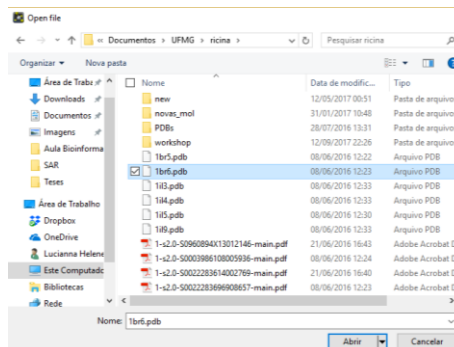
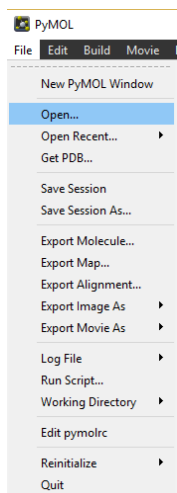
A) Comando fetch <PDB> . Exemplo: fetch 1br6.

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.

Criando arquivos log

Quando a linha de comando é utilizada, é possível criar um arquivo log, no formato .pml, conterá todos os comandos utilizados. Esse arquivo pode ser posteriormente editado, e pode ser empregado como um script, para repetir a mesma série de comandos.

- Para criar um arquivo log (*my_script.pml*), que conterá todos os comandos digitados durante uma sessão:

```
PyMOL> log_open my_script.pml
```

- Para fechar o arquivo log:
PyMOL> log_close

- Para carregar um script:
PyMOL> @my_script.pml

Instalando Plugin Autodock/Vina

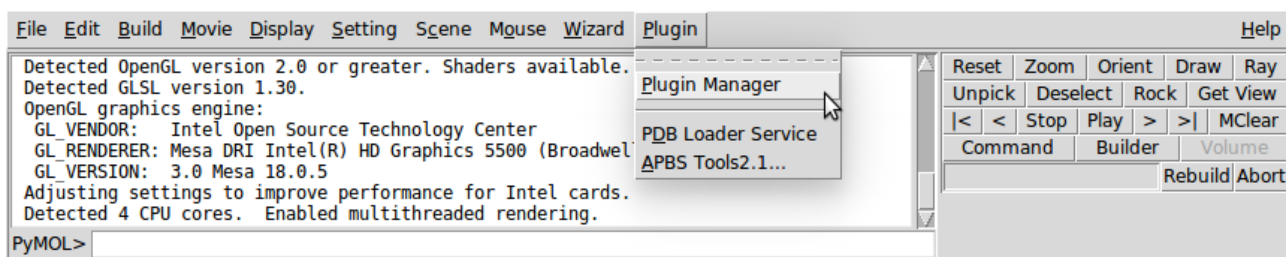
1) Baixar o arquivo **autodock.py**

- No site do desenvolvedor <https://github.com/ADplugin/ADplugin>

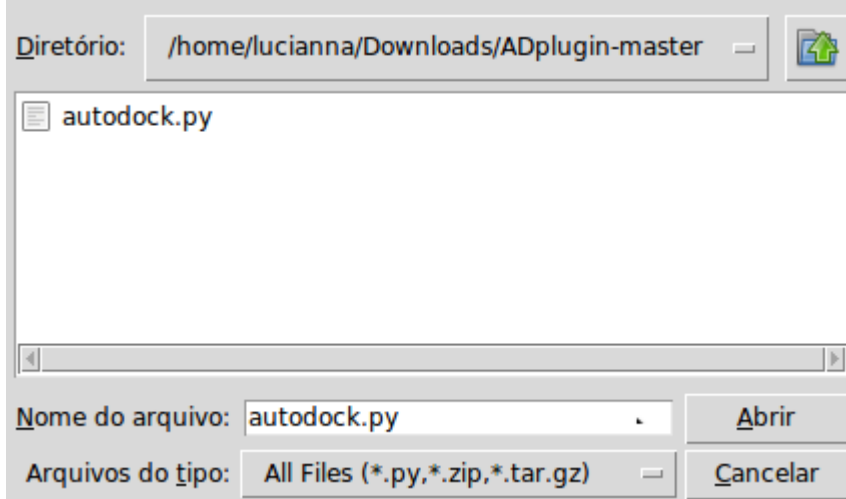
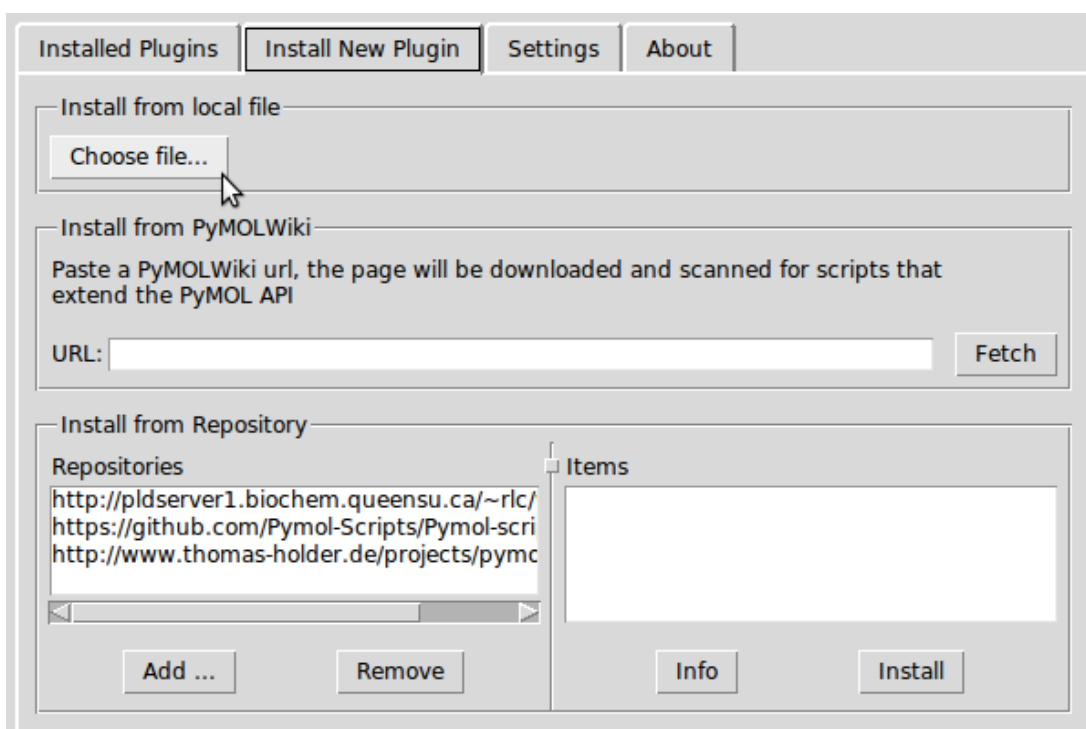
2) Abrir o pymol.

Pressione as teclas **Ctrl + Alt + T** para abrir o Terminal e digite **pymol**.

3) Na aba **Plugin > Plugin Manager**



4) Escolher a aba **Install New Plugin**. Vamos instalar com o arquivo autodock.py, utilizando o botão **Choose file** em **Install from local file**.



Clique em Abrir. Escolha um dos diretórios e clique Sim.

5) Abrir o plugin **Autodock/Vina**. Configurar os caminhos dos programas na aba **Configuration**:

AutoDockTools: /usr/local/MGLTools-1.5.6/MGLToolsPckgs/AutoDockTools/Utilities24/
 autogrid4 executable: /usr/local/bin/autogrid4

autogrid4 executable: /usr/local/bin/autodock4
vina executable: /usr/local/bin/vina
Working Directory: /home/aluno

Clicar em **Save Plugin Configuration File**.

6) **IMPORTANTE**: Após configurar os caminhos **FECHAR** o pymol e abrir novamente.

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

IMPORTANTE: No VirtualBox, pode ocorrer “segmentation fault” no Pymol. Então precisamos digitar o seguinte comando:

```
set use_shaders, 0
```

1) Crie um arquivo log:

```
PyMOL> log_open parte1.pml
```

2) 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
```

3) 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.

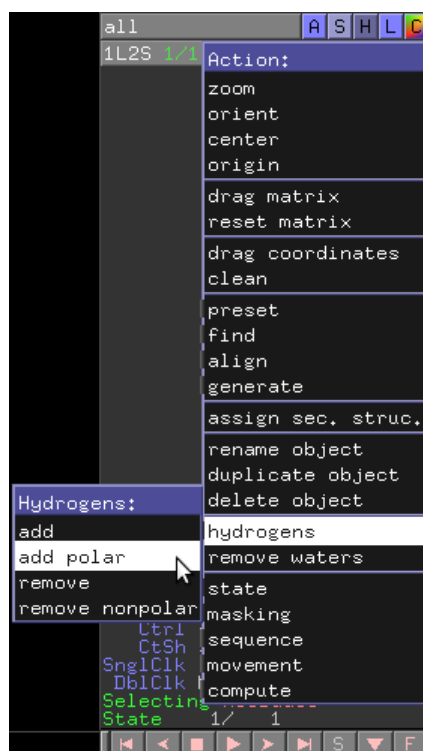
- 4) Vamos explorar diferentes representações da estrutura secundária, alterando a forma de representação de hélices e folhas-beta. A cada comando abaixo, observe as modificações:

```
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
```

- 5) 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
```

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



- 7) 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**

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

PyMOL> **show surface, 1L2S**

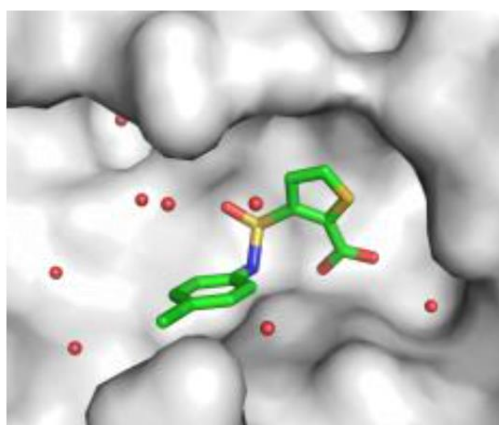
PyMOL> **color gray, 1L2S**

9) 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):



10) 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)

11) Feche o arquivo log:

PyMOL> **log_close**

12) Salve a imagem com uma resolução específica:

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

13) Salve a sessão do Pymol (menu File, save as session) como 1L2S_parte1.pse

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

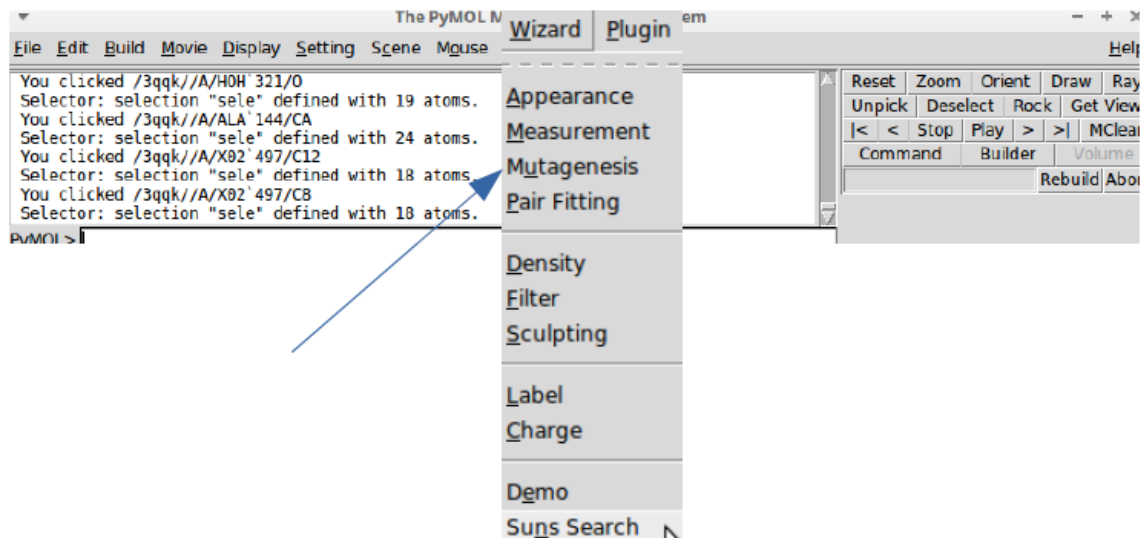
Calcule as interações entre átomos polares do inibidor e a enzima ou moléculas de água.

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

15) 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.

16) Vamos alterar as cores das linhas para visualizarmos melhor. Para isso, digite na linha de comando

```
PYMOL> set dash_color, black
```

17) Após medir todas as distâncias, clique em **Done**, no Menu **Measurement**, e salve como 1L2S_dist.pse.

Exercício:

- 1) Abra a estrutura 1KE4.
- 2) Elimine a cadeia B.
- 3) Mostre a proteína em **cartoon** e cor laranja (Orange). Utilize o comando **ray** e crie uma figura em alta resolução com o comando **png**.
- 4) Com a proteína em **cartoon**, mude a coloração para destacar a estrutura secundária. Utilize o comando **ray** e crie uma figura em alta resolução com o comando **png**.

Dica:



- 5) Esconda a representação **cartoon** e mostre a proteína na representação **ribbon**. Utilize o comando **ray** e crie uma figura em alta resolução com o comando **png**.
- 6) Esconda a representação **ribbon** e mostre a proteína na representação **sticks** colorindo por elemento. Utilize o comando **ray** e crie uma figura em alta resolução com o comando **png**.

Dica:

```
all A S H L C
```

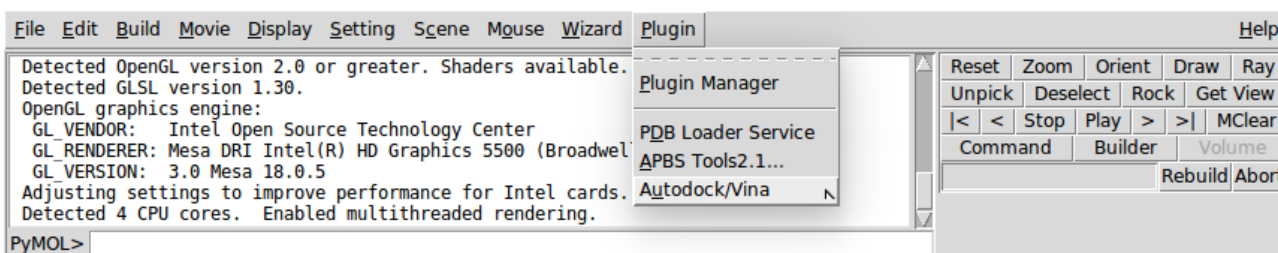
Atoms	Color:
HNOS...	by element
CHNOS...	by chain
CHNOS...	by ss
CHNOS...	by rep
CHNOS...	spectrum
CHNOS...	auto
CHNOS...	reds
CHNOS...	greens
CHNOS...	blues
set 2	yellow
set 3	magentas
set 4	cyans
set 5	oranges
set 6/H	tints
	grays

- 7) Esconda a representação **sticks** e mostre a proteína na representação **surface**. Utilize uma cor de sua escolha. Utilize o comando **ray** e crie uma figura em alta resolução com o comando **png**.
- 8) Crie e salve todas as figuras de alta resolução em um documento (Word, libre office, etc).

Parte 2 – Atracamento molecular utilizando o plugin 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.

- 1) Para o atracamento vamos primeiro remover as moléculas de água:
PYMOL> **remove resn hoh**
- 2) Ainda com a estrutura 1L2S. Vamos abrir o plugin do Autodock/Vina no pymol em **Plugin > Autodock/Vina**.



- 3) Na aba Grid, vamos estabelecer o valor da caixa e seu centro no ligante. Troque os valores de **Grid Definition-Parameters** para:

Spacing: **1.0**

X-points: **25**

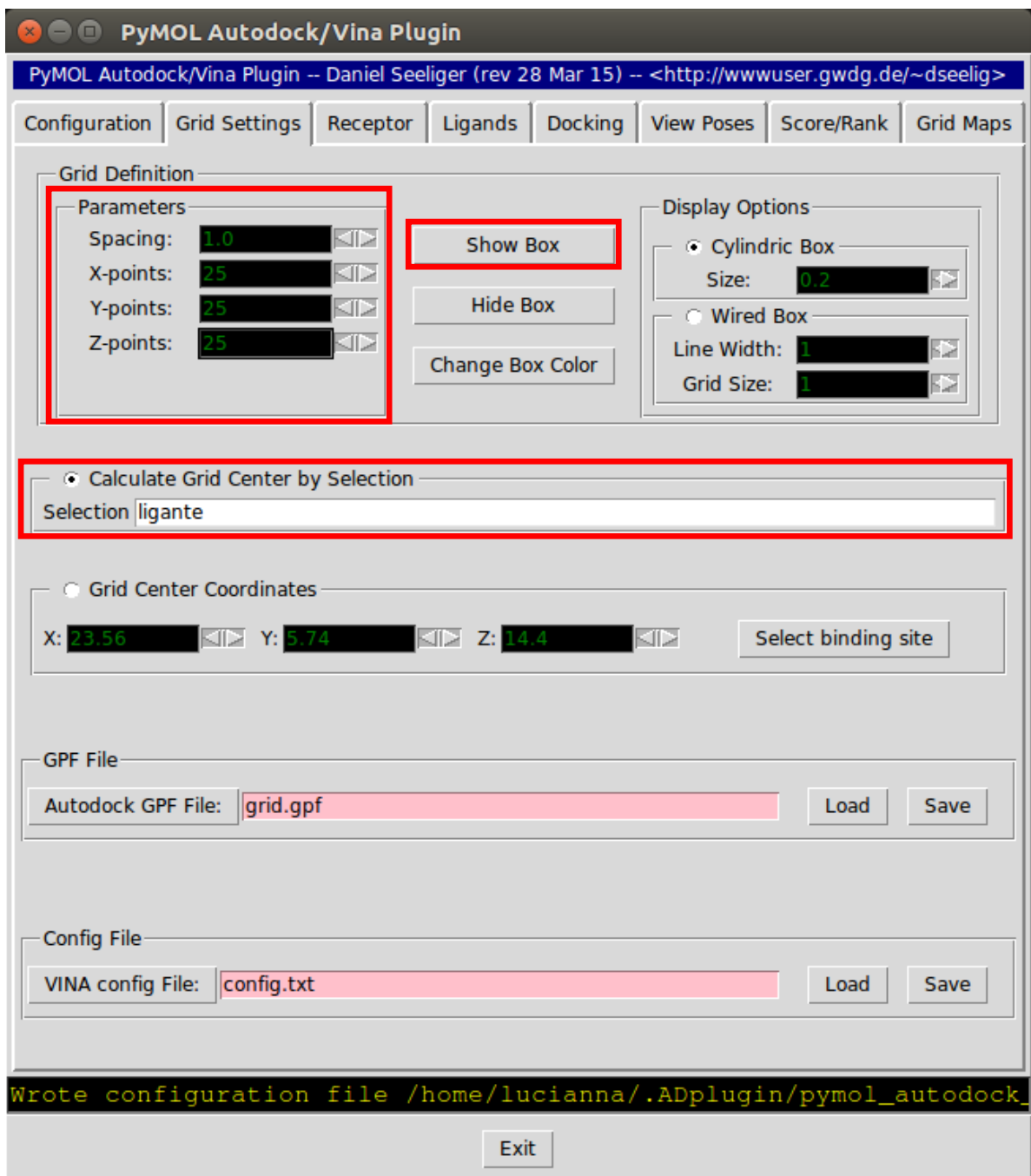
Y-points: **25**

Z-points: **25**

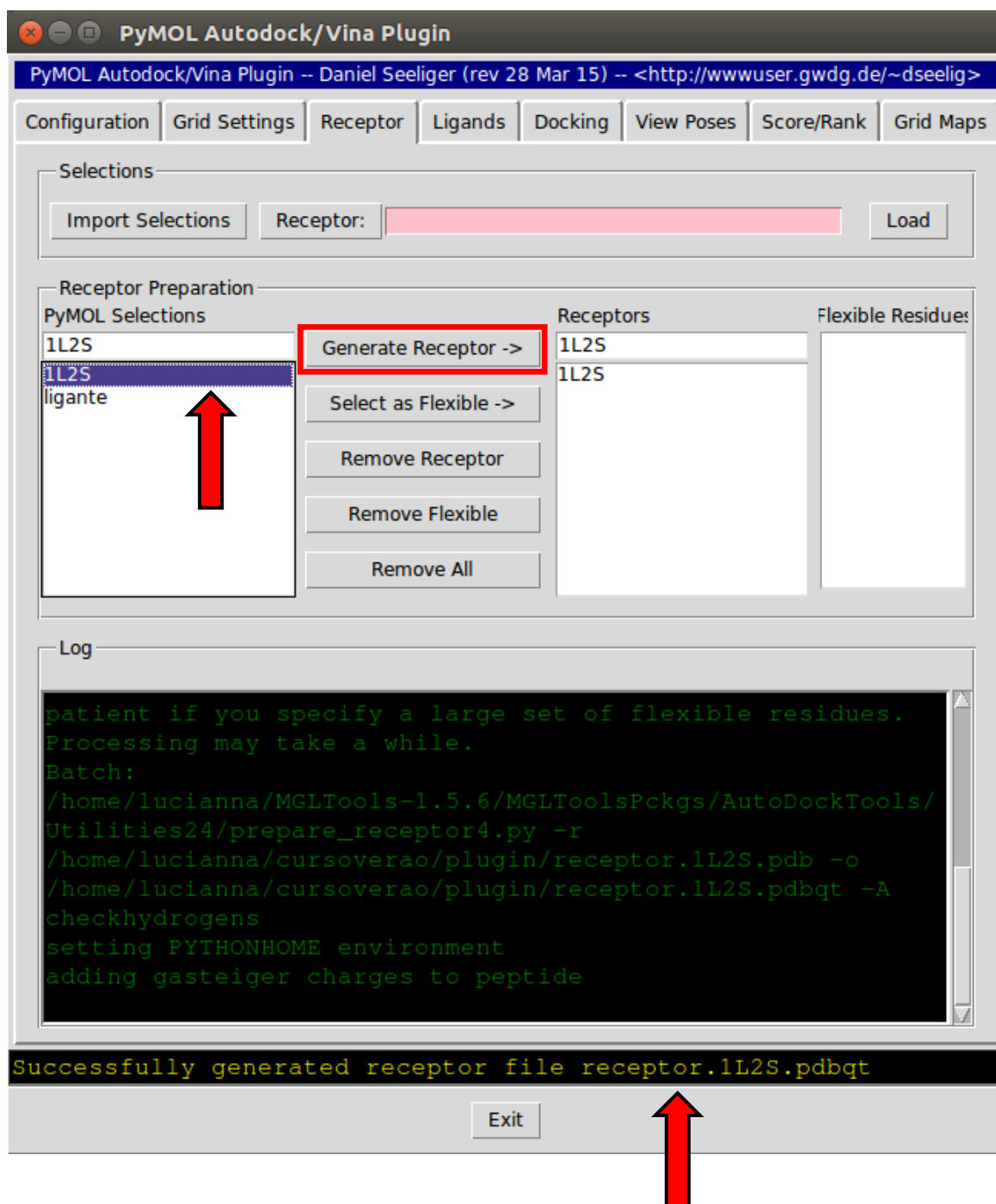
Clique no botão **Show Box**.

Selecione a opção **Calculate Grid Center by Selection** e digite **ligante** na caixa de diálogo. Dê enter.

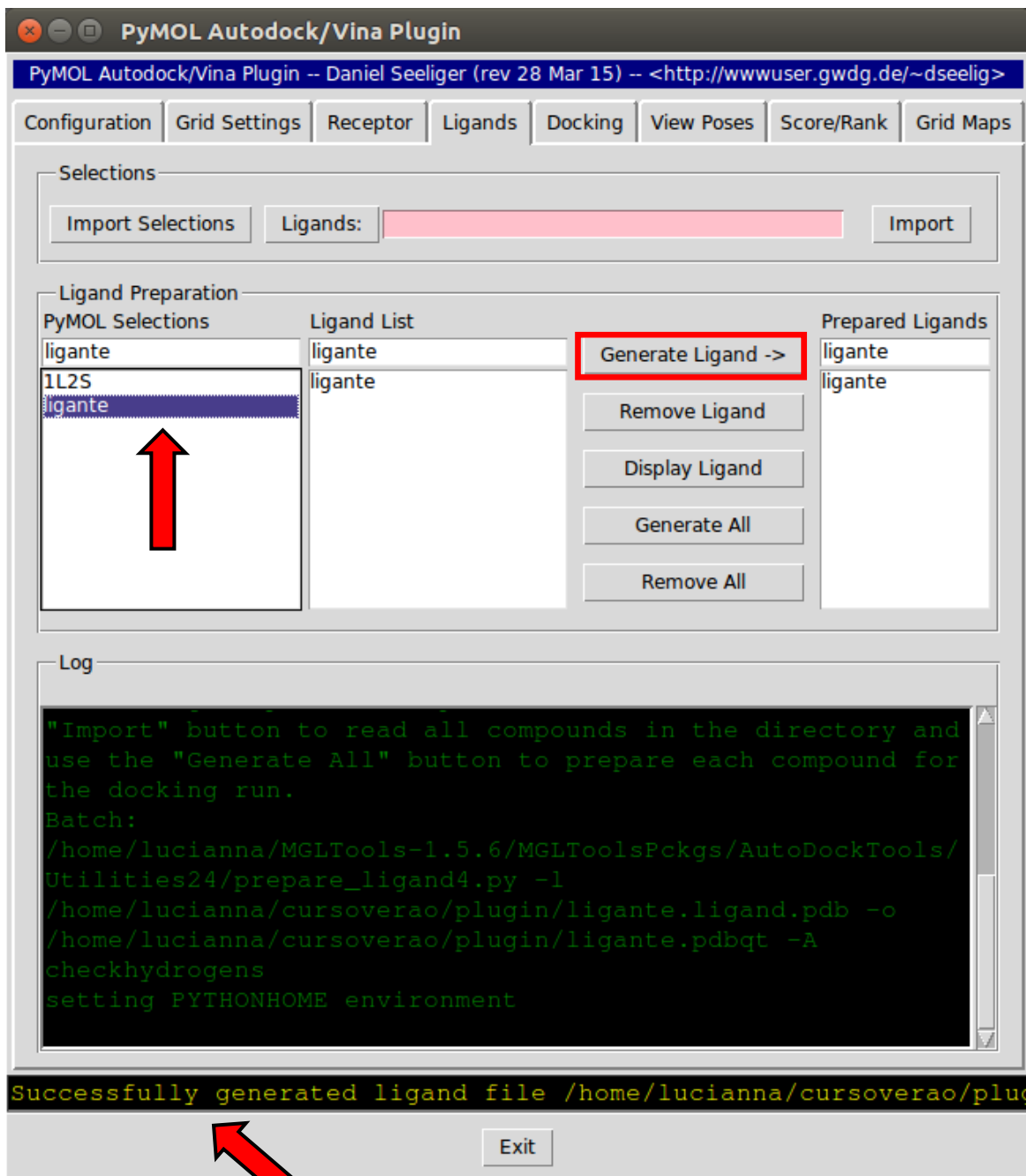
OBS: O Vina não faz um pré-cálculo de Grid. Porém, uma demarcação da área de docking (sítio ativo ou até mesmo a proteína toda) precisa ser estabelecida.



- 4) A preparação do receptor é feita na aba **Receptor**. Em **PyMOL Selections** escolha **1L2S**. Clique em **Generate Receptor**.

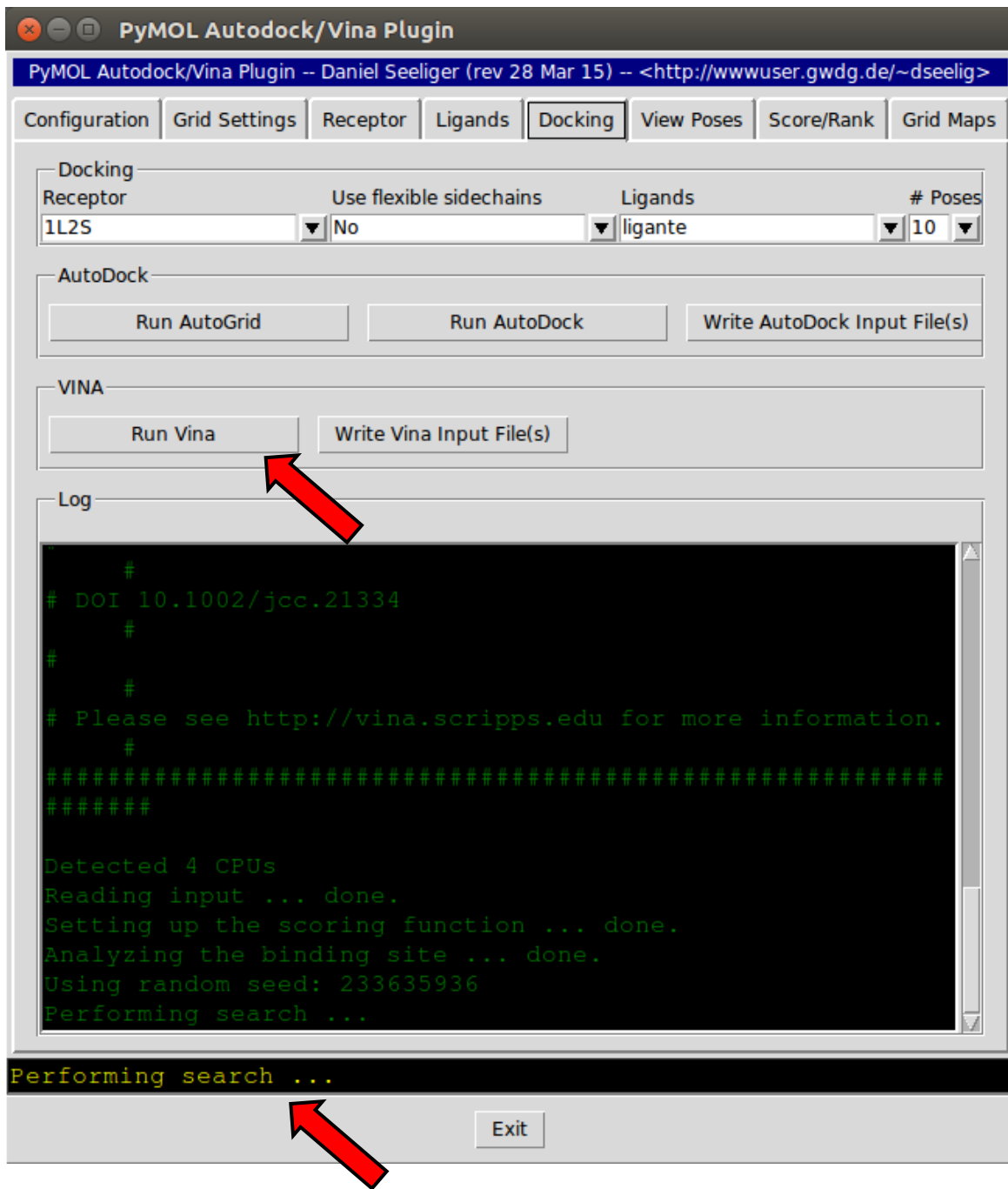


- 5) preparação do ligante é feita na aba **Ligands**. Em **PyMOL Selections** escolha **ligante**. Clique em **Generate Ligand**.

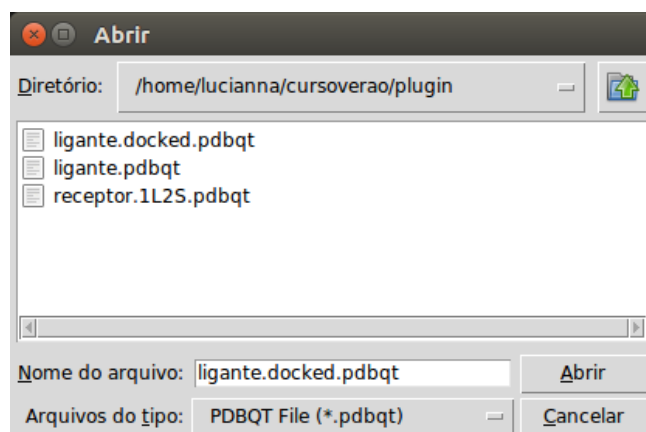


OBS: Vários ligantes podem ser preparados ao mesmo tempo.

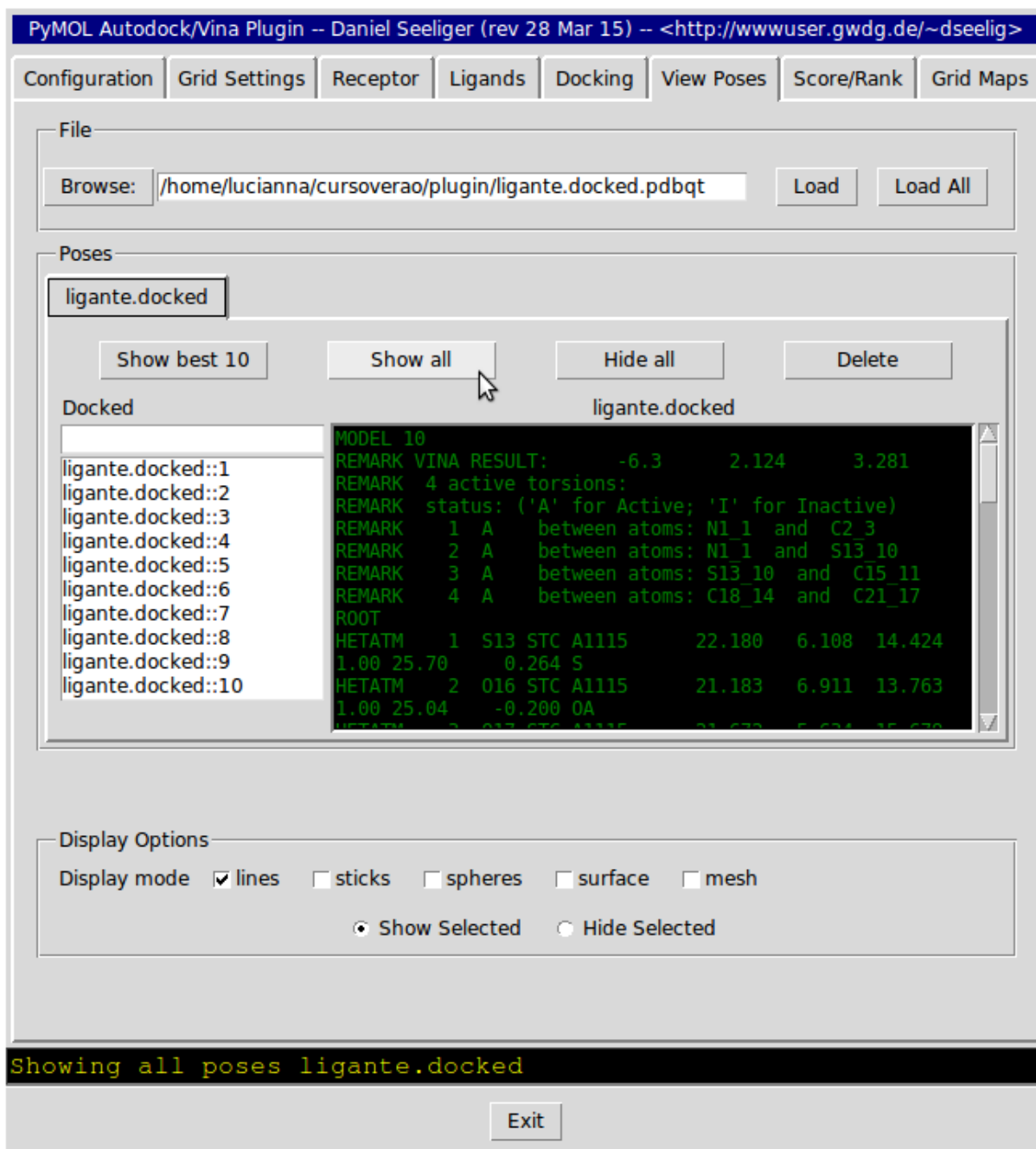
- 6) Com grid selecionado, receptor, e ligante preparados podemos realizar o atracamento. Na aba **Docking**, clique no botão **Vina**.



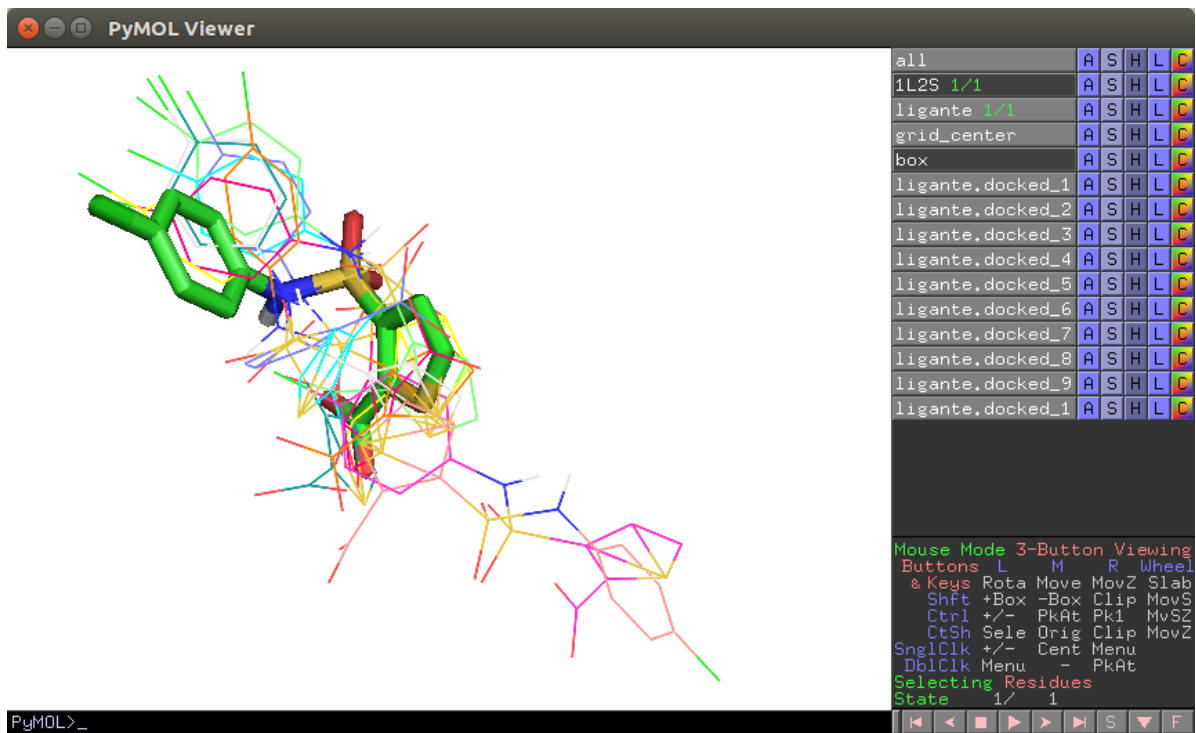
- 7) Assim que terminar podemos ver as poses resultantes na aba **View Poses**. Clique no botão **Browse** e selecione o arquivo **ligante.docked.pdbqt**.



- 8) Para mostrar todas as poses obtidas na janela de visualização do pymol, clique no botão **Show all**.



OBS: 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.



9) Vamos calcular o RMSD entre os ligantes com o comando:

```
PYMOL> rms_cur ligante.docked_1, ligante
PYMOL> rms_cur ligante.docked_2, ligante
PYMOL> rms_cur ligante.docked_3, ligante
      :
PYMOL> rms_cur ligante.docked_10, ligante
```

Alguns valores ficaram abaixo de 2.0 Å? Ou seja, alguma pose ficou muito próxima da posição cristalográfica do ligante?

9.1) ALTERNATIVA!

Existe um script no AutodockTools que calcula o rmsd com melhor precisão que o pymol. Para executar precisamos:

- Separar as poses em arquivos únicos usando o vina_split


```
$ vina_split --input ligante.docked.pdbqt
```

- Como o script faz apenas uma comparação de vez podemos usar um script bash para iterar entre as poses. O script se encontra na página **Arquivos** da pasta do OneDrive:

```
$ bash compute_rmsd.sh
```

Obtemos o arquivo texto **summary_rms_results.txt**:

The screenshot shows a gedit window titled 'summary_rms_results.txt (~/.cursoverao/plugin/split) - gedit'. The window contains a table with three columns: 'reference filename', 'test filename', and 'rms'. The data is as follows:

reference filename	test filename	rms
../ligante.pdbqt,	ligante.docked_ligand_01.pdbqt,	2.3319
../ligante.pdbqt,	ligante.docked_ligand_02.pdbqt,	7.1446
../ligante.pdbqt,	ligante.docked_ligand_03.pdbqt,	0.7318
../ligante.pdbqt,	ligante.docked_ligand_04.pdbqt,	9.4748
../ligante.pdbqt,	ligante.docked_ligand_05.pdbqt,	2.5109
../ligante.pdbqt,	ligante.docked_ligand_06.pdbqt,	3.3603
../ligante.pdbqt,	ligante.docked_ligand_07.pdbqt,	2.5729
../ligante.pdbqt,	ligante.docked_ligand_08.pdbqt,	2.5026
../ligante.pdbqt,	ligante.docked_ligand_09.pdbqt,	3.4201

Notemos que existe uma pose com **RMSD < 2.0 Å** e está foi ranqueada na terceira posição entre as dez poses.

10) Vamos analisar as interações intermoleculares entre a pose de menor RMSD e a enzima.

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

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

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

Alguma interação foi igual a vista entre ligante do cristal e receptor?

12) Após medir as distâncias, clique em **Done**, no Menu **Measurement**, e salve como **1L2S_dock1.pse**. Feche o pymol.

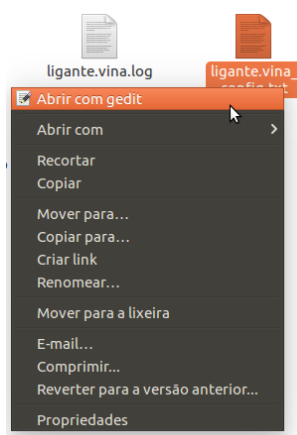
Exercício:

9) Escolha a pose do ligante que fica mais próxima do cristal e faça uma figura no pymol, com boa resolução, mostrando o ligante docado e o cristal.

Parte 3 – Modificando arquivo de configuração do Vina e rodando em linha de comando

O Vina na verdade foi projetado para não ter uma interface e com isso ser um programa mais leve. A preparação de receptor e ligante pode ser feita com outros programas como por exemplo o Openbabel, embora, obrigatoriamente tem que possuir o formato .pdbqt. A facilidade de usar o plugin do pymol fica por conta de visualização da caixa, na escolha do centro da caixa, e na criação do arquivo de configuração do Vina automaticamente. Porém, podemos modificar esse arquivo de configuração com um editor de texto e rodar o Vina novamente por linha de comando.

- 1) Abra o arquivo de configuração do vina criado na hora do docking com o editor de texto Gedit. Deve estar na pasta de trabalho com o nome **ligante.vina_config.txt**

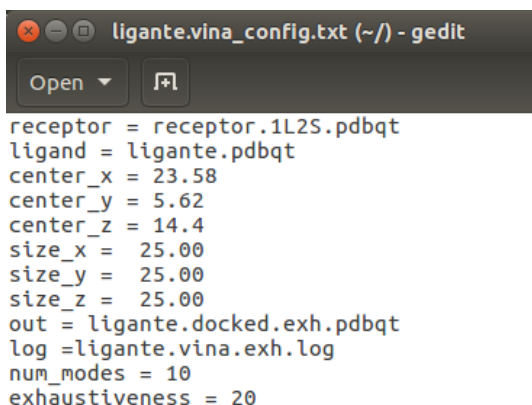


- 2) No final do arquivo adicione o parâmetro:
exhaustiveness = 20

```
*ligante.vina_config.txt (~/) - gedit
receptor = receptor.1L2S.pdbqt
ligand = ligante.pdbqt
center_x = 23.58
center_y = 5.62
center_z = 14.4
size_x = 25.00
size_y = 25.00
size_z = 25.00
out = ligante.docked.pdbqt
log = ligante.vina.log
num_modes = 10
exhaustiveness = 20
```

OBS: A explicação mais simples para o parâmetro **exhaustiveness** é o tempo gasto na busca de um mínimo local de ligação entre ligante e receptor. O valor padrão desse parâmetro é 9. E quanto maior for o número mais tempo computacional é gasto, ou seja, o tempo para realizar o atracamento aumenta.

- 3) Mude o nome dos arquivos de saída para não sobrescrever os anteriores.



```
receptor = receptor.1L2S.pdbqt
ligand = ligante.pdbqt
center_x = 23.58
center_y = 5.62
center_z = 14.4
size_x = 25.00
size_y = 25.00
size_z = 25.00
out = ligante.docked.exh.pdbqt
log = ligante.vina.exh.log
num_modes = 10
exhaustiveness = 20
```

- 4) Abra um terminal na pasta de trabalho. Com o direito do mouse, escolha **Abrir um terminal**. Ou utilize CTRL+ALT+T e redirecione para a pasta de trabalho. Por exemplo:

```
$ cd /home/aluno
```

- 5) Digite o comando:

```
$ vina --config ligante.vina_config.txt
```

- 6) Após o término da execução do programa. Abra o pymol de novo. E abra a sessão **1L2S_dist.pse**.
- 7) No plugin Autodock/Vina vá para a aba View Poses e abra o novo arquivo de resultado do atracamento **ligante.docked.exh.pdbqt**.
- 8) Faça as análises de RMSD e interações.

Alguma pose abaixo de 2.0 Å? Alguma interação se manteve?

- 9) Salve como **1L2S_dock2.pse**.