

Estudos de *Docking* Proteína-Ligante

CAMILA MAGALHÃES – camilasm@ufrj.br

DIOGO MARINHO – diogoma@lncc.br

ISABELLA ALVIM – isabella@lncc.br

LAURENT DARDENNE – dardenne@lncc.br

Informações:

Login: vemmsb

Senha: emmsb

Diretório: Desktop/TurmaX/

***Substituir X pelo número da turma**

Comandos Básicos do Linux:

cd diretório/

Entra na pasta diretório/

ls diretório/

Lista as subpastas e arquivos existentes em diretório/

cd ..

Retorna para a pasta anterior

mkdir diretório/

Cria a pasta diretório/

rm arquivo.txt

Remove arquivo.txt (.txt ou outro formato de arquivo)

rmdir diretório/

Remove a pasta diretório/

cp diretório1/arquivo.txt diretório2/

Copia arquivo.txt no diretório1/ para diretório2/

*** Neste curso os arquivos já estão salvos pastas mencionadas no Tutorial.**

Dia 1 – Preparação dos Arquivos da Proteína e do Ligante

1. Download dos arquivos da proteína e do ligante

*** O arquivo já foi baixado (1HSG) no diretório 1HSG/, não sendo necessário realizar as seguintes etapas em aula.**

- 1.1 Buscar o arquivo com as coordenadas tridimensionais (.pdb) do complexo no *Protein DataBank*: www.pdb.org. No campo de busca em branco, digitar o código do complexo 1HSG e clicar em **Search**;

- 1.2 As informações sobre o complexo serão carregadas na página. Analisar a presença de ligantes e cofatores, número de cadeias e outras informações relevantes (organismo de origem, resolução do cristal...);
- 1.3 Fazer o download do arquivo do complexo (**Download Files → PDB File**) para sua pasta pessoal;
- 1.4 Clique no código do ligante **MK1** para carregar a página contendo informações específicas sobre este inibidor. É possível fazer o download da estrutura 3D do ligante em sua conformação ideal, ou seja, conformação mais estável do ligante no vácuo (**Download Files → Structure Data File**). Essa conformação é calculada pelos programas Corina e OpenEye Omega. O *docking* com essa estrutura será feito no terceiro dia, não sendo necessário fazer o download em aula;
- 1.5 Analisar a estrutura tridimensional do complexo no visualizador Pymol:

pymol 1HSG.pdb &

Inicialmente o complexo é carregado na visualização de linhas, e os átomos de oxigênio das moléculas de água são representados por **x**. Visualize as estruturas secundárias da proteína (**S → Cartoon**). No Pymol, é possível visualizar rapidamente as interações existentes entre ligantes e proteína. Carregue essa visualização e analise algumas interações (**A → Preset → Ligand Sites → Cartoon**). Com isso, apenas os resíduos da proteína que interagem com o ligante são representados como linhas. Todos os outros só são representados por *cartoon*. Automaticamente são calculadas as ligações de hidrogênio formadas entre a proteína, o ligante e moléculas de água no sítio ativo. Após analisar as interações, feche o programa;

- 1.6 Analisar o arquivo texto (.pdb) do complexo num editor de texto. Digite no terminal:

kate 1HSG.pdb &

Observe os comentários e as informações existentes no arquivo. As linhas do arquivo que iniciam com ATOM se referem a cada átomo da proteína. As linhas que iniciam com HETATM se referem aos átomos de ligantes (cofatores, inibidores, íons) e a moléculas de água (sempre nomeadas como HOH). A seguir a descrição de cada campo do arquivo PDB seguido por uma parte da 1HSG:

COLUMNS	DATA TYPE	CONTENTS
1 - 6	Record name	"ATOM "
7 - 11	Integer	Atom serial number.
13 - 16	Atom	Atom name.
17	Character	Alternate location indicator.
18 - 20	Residue name	Residue name.
22	Character	Chain identifier.
23 - 26	Integer	Residue sequence number.
27	AChar	Code for insertion of residues.
31 - 38	Real(8.3)	Orthogonal coordinates for X in Angstroms.
39 - 46	Real(8.3)	Orthogonal coordinates for Y in Angstroms.
47 - 54	Real(8.3)	Orthogonal coordinates for Z in Angstroms.
55 - 60	Real(6.2)	Occupancy.
61 - 66	Real(6.2)	Temperature factor (Default = 0.0).
73 - 76	LString(4)	Segment identifier, left-justified.
77 - 78	LString(2)	Element symbol, right-justified.
79 - 80	LString(2)	Charge on the atom.

Exemplo:

	1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890123456789012345678901234567890								
ATOM	63	N	ARG A	8	26.992	23.877	6.169	1.00 27.34 N
ATOM	64	CA	ARG A	8	25.757	24.566	6.593	1.00 29.87 C
ATOM	65	C	ARG A	8	26.029	26.025	6.996	1.00 31.94 C
ATOM	66	O	ARG A	8	26.947	26.291	7.775	1.00 33.83 O
ATOM	67	CB	ARG A	8	25.087	23.849	7.776	1.00 28.16 C
ATOM	68	CG	ARG A	8	24.646	22.409	7.505	1.00 27.47 C
ATOM	69	CD	ARG A	8	23.728	21.896	8.637	1.00 25.45 C
ATOM	70	NE	ARG A	8	22.952	20.730	8.230	1.00 23.06 N
ATOM	71	CZ	ARG A	8	22.367	19.871	9.064	1.00 28.80 C
ATOM	72	NH1	ARG A	8	22.376	20.074	10.370	1.00 30.95 N
ATOM	73	NH2	ARG A	8	21.776	18.789	8.589	1.00 26.02 N

1.7 Para calibrar os parâmetros para o *docking* devemos primeiro fazer um *redocking* com a estrutura do ligante na conformação que ele se encontra complexado com a proteína. Por isso, precisamos retirar o ligante diretamente do arquivo do complexo no formato PDB.

1.7.1 Copie o arquivo 1HSG.pdb para outro arquivo ind.pdb:

```
cat 1HSG.pdb > ind.pdb
```

1.7.2 Abra o arquivo ind.pdb num editor de texto:

```
kate ind.pdb &
```

1.7.3 Remova todas as linhas, exceto as que iniciam com HETATM e que sejam referentes ao ligante MK1 (apague as que se referem às moléculas de água);

1.7.4 Salve o arquivo do ligante;

1.7.5 Copie o arquivo 1HSG.pdb para 1hsg_p.pdb:

```
cat 1HSG.pdb > 1hsg_p.pdb
```

- 1.7.6 Abra o arquivo 1hsg_p.pdb num editor de texto:

```
kate 1hsg_p.pdb &
```

- 1.7.7 Remova todas as linhas que iniciam com HETATM que sejam referentes ao ligante MK1 (mantenha as que se referem às moléculas de água);

- 1.7.8 Salve o arquivo da proteína;

- 1.8 Para adicionar hidrogênios podemos usar a ferramenta *Babel*. Escolha como *input* o formato .pdb e como *output* o formato .pdb.

```
babel -ipdb ind.pdb -opdb ind.pdb -h
```

- 1.9 Abra o arquivo do ligante preparado pelo Babel e observe os hidrogênios adicionados.

```
pymol ind.pdb &
```

2. Preparação da proteína

- 2.1 Os arquivos provenientes do PDB não são perfeitos. Além de não possuírem hidrogênios, podem ter átomos faltando. Nesta etapa, vamos preparar os arquivos da proteína e do ligante usando o AutoDock4. Abra o ADT – *AutoDock Tools* e escolha a versão 4.0;

- 2.1.1 Abrir o arquivo da proteína (clique com botão direito do *mouse* em **PMV Molecules**) no formato .pdb;

- 2.1.2 Visualise a proteína com cores por átomo (clique no losango embaixo de **Atom**, na linha referente à proteína). Experimente outras visualizações;

- 2.1.3 Como o *docking* será feito sem as moléculas de água, precisamos retirá-las do arquivo. Isso pode ser feito manualmente (no arquivo texto do PDB, como feito na etapa anterior) ou automaticamente pelo AutoDock4 (**Select** → **Select From String**). Em **Residue** coloque **HOH***, depois clique em **Add** e depois em **Dismiss** para fechar a janela de seleção. Com isso, 127 átomos são selecionados. Apague esta seleção (**Edit** → **Delete** → **Delete AtomSet**) e clique em **CONTINUE** para confirmar;

- 2.1.4 Adicionar hidrogênios na proteína (**Edit** → **Hydrogens** → **Add**). Escolha adicionar **All Atoms** com o método **noBondOrder**, escolha **yes** para renumerar e clique em **OK**;

- 2.1.5 Adicionar cargas na proteína (**Edit → Charges → Compute Gasteiger**). A carga total tem valor próximo a 6.0;
- 2.1.6 Como o AutoDock4 usa uma aproximação de átomo unido em seu campo de forças, é preciso fazer um *merge* dos hidrogênios não polares (**Edit → Hydrogens → Merge Non-Polar**);
- 2.1.7 Assinalar os tipos de átomos do ADT 4.0 (**Edit → Atoms → Assign AD4 type**);
- 2.1.8 Salvar o arquivo da proteína no formato.pdbqt (**File → Save → Write PDBQT**). Remova as opções HELIX, SHEET e HETATM, mantendo apenas ATOM.
- 2.1.9 Apague a estrutura da proteína do AutoDock4, uma vez que está no formato .pdb e vamos usar o .pdbqt (clique com o botão **direito** no nome da proteína e escolha **Delete**);
- 2.1.10 Abra o arquivo da proteína no formato .pdbqt no terminal para visualizar a diferença do formato:

```
kate 1hsg_prep.pdbqt &
```

3. Preparação do ligante

- 3.1 Abrir o arquivo do ligante no formato .pdb no AutoDock4 (**Ligand → Input → Open**). Para visualizar todos os arquivos, clique em **PDBQT Files: (*.pdbqt)** e escolha **PDB Files: (*.pdb)**. Selecione o arquivo **ind.pdb**. Uma janela com uma mensagem irá abrir indicando o resultado da preparação automática do ligante feita pelo ADT. Clique em **OK**;
- 3.2 Selecionar um átomo como *root* (**Ligand → Torsion Tree → Detect Root**);
- 3.3 Calcular o número de torções existentes no ligante (**Ligand → Torsion Tree → Choose Torsion**);
- 3.4 Selecionar o número de torções para o cálculo de *docking* (**Ligand → Torsion Tree → Set Number of Torsions**). Nas aulas serão testados quatro protocolos diferentes. Salve um arquivo do ligante no formato .pqbt para cada configuração de torção:
 - 3.4.1 Ligante rígido (coloque **0** em **active torsions**);
 - 3.4.2 Torções mais internas (coloque **6** em **active torsions** e escolha **fewest**);

- 3.4.3 Torções mais periféricas (coloque **6** em **active torsions** e escolha **most**);
- 3.4.4 Todas as torções (coloque **14** em **active torsions**);
- 3.5 Feche a janela (clique em **Dismiss**);
- 3.6 Salvar o arquivo do ligante no formato.pdbqt (**Ligand** → **Output** → **Save as PDBQT**).
Para cada protocolo salve com os nomes:
 - 3.6.1 ind_rig.pdbqt;
 - 3.6.2 ind_few6.pdbqt;
 - 3.6.3 ind_most6.pdbqt;
 - 3.6.4 ind_all.pdbqt;
- 3.7 Abra o arquivo ind_all.pdbqt no editor de texto e observe as informações incluídas, como o número de torções ativas;
- 3.8 Apague a estrutura do ligante, pois a que está carregada ainda está no formato .pdb (clique com o botão **direito** no nome do ligante e escolha **Delete**);
- 3.9 Faça o download e prepare os arquivos dos complexos 2MCP e 4DFR que serão usados no último dia de aula. Salve os arquivos nos respectivos diretórios de cada complexo.

Dia 2 – Docking de Ligante Bastante Flexível – 1HSG

4. Arquivo de Grid

- 4.1 Abrir o arquivo previamente preparado da proteína (**Grid** → **Macromolecule** → **Open**) no formato **.pdbqt**. O AutoDock4 verifica a proteína, assim como as cargas parciais, a presença de hidrogênios não-polares e tipos de átomos. Clique em **Yes** para manter as cargas previamente calculadas;
- 4.2 Abrir o arquivo do ligante para o cálculo da Grid (**Grid** → **Set Map Types** → **Open Ligand**. Se o ligante não estiver carregado no programa, ou **Chose Ligand** caso contrário);
- 4.3 Criar a caixa da Grid para o ligante (**Grid** → **Grid Box**) usando as seguintes configurações: *número de pontos em cada dimensão* **60, 60 e 60** (clique com o botão direito em cima do valor para abrir uma caixa de edição e edite esses valores), e *centro*

da *grid* (**Center** → **Center on Ligand**). Feche esta janela, salvando as configurações (**File** → **Close saving current**);

4.4 Salvar o arquivo de configuração da Grid (**Grid** → **Output** → **Save GPF**) como **grid.gpf**;

4.5 Abra o arquivo de parâmetros da *grid* para visualizar o texto (**Grid** → **Edit GPF**);

4.6 Executar AutoGrid4 (**Run** → **Run AutoGrid** → **Launch**);

5. Arquivo de Parâmetros para o *Docking*

5.1 Selecionar o arquivo dos resíduos fixos no *docking*. Neste tutorial, será utilizado o próprio arquivo da proteína (**Docking** → **Macromolecule** → **Set Rigid Filename**) no formato .pdbqt;

5.2 Escolher o ligante para o *docking* (**Docking** → **Ligand** → **Choose**). Irá abrir um painel com várias informações sobre o ligante, como nome, tipos de átomos, centro, número de torções ativas e graus de liberdade. Neste tutorial, vamos manter os valores padrão. Observe as informações que foram alteradas na etapa de preparação do ligante, como número de torções e os graus de liberdade. Clique em **Close**;

5.3 Definir os parâmetros do Algoritmo Genético (**Docking** → **Search Parameters** → **Genetic Algorithm**). Escolha a opção **short** no número máximo de avaliações, ou seja, 250.000. Clique em **Accept** para continuar;

5.4 Analisar o arquivo de parâmetros do *docking* (**Docking** → **Docking Parameters**). Neste tutorial, serão mantidos os valores padrão. Clique em **Close**;

5.5 Salvar o arquivo de saída com os parâmetros do *docking* (**Docking** → **Output** → **Lamarckian GA**). Salve como **docking_ind.dpf**;

5.6 Executar AutoDock4 (**Run** → **Run AutoDock** → **Launch**);

5.7 Abrir o arquivo de log gerado **docking_ind.dlg** e observar a duração do cálculo;

5.8 Abra o arquivo de resultados do *docking* (**Analyse** → **Dockings** → **Open**) e selecione **docking.dlg**;

5.9 Escolha a proteína para analisar os resultados (**Analyse** → **Macromolecule** → **Choose**). Caso a proteína não esteja carregada no ADT, clique em **Open**;

5.10 Carregue as conformações encontradas, ordenadas por energia (**Analyse** → **Conformations** → **Play, ranked by energy**);

5.10.1 Analise as diferentes conformações (1), abra a caixa de opções e experimente algumas opções (2):



- 5.11 Observe os agrupamentos formados (**Analyse → Clusterings → Show**);
- 5.12 Experimente os outros protocolos de *docking* para o ligante. Os arquivos do ligante, e recarregue o próximo;
- 5.13 Não será necessário criar a *grid* uma vez que o arquivo já está pronto para este ligante;
- 5.14 Repita o passo 5.2 para alterar o ligante para ser calculado;
- 5.15 Salve o arquivo de parâmetros do *docking* como feito na etapa 5.5;
- 5.16 Repita as etapas para executar o AutoDock4 e para analisar os resultados.

Dia 3 – Virtual Screening ou Docking com Diferentes Protocolos

Este protocolo tem por objetivo demonstrar o protocolo de *Virtual Screening* usando comandos *shell* e scripts em Python usando o pacote de programas do AutoDock4. Todos os passos serão feitos a partir de scripts prontos, inclusive a preparação dos arquivos. Todos os arquivos deste tutorial foram obtidos no site do AutoDock (<http://autodock.scripps.edu/>), e posteriormente alterados. Iremos manter o foco na HIV-1 protease, no entanto, para o *Virtual Screening* iremos usar a proteína complexada com outro inibidor, L-735.524 (PDB: 1HPV), uma vez que não iremos fazer *redocking*. Para validar o protocolo usado, vamos fazer *crossdocking*, ou seja, fazer o *docking* de um inibidor na estrutura proteína complexada outro inibidor, usando parâmetros do algoritmo genético menos robustos que o utilizado para o *Virtual Screening*.

6. Populando o diretório de ligantes obtendo os arquivos .mol2

6.1 Os ligantes que serão utilizados no *Virtual Screening* foram baixados do banco de ligantes ZINC (<http://zinc.docking.org/>). Quando efetuamos uma busca no ZINC e baixamos o arquivo no formato mol2, na verdade estamos baixando um multimol2. Este é um arquivo único que possui todos os ligantes no formato .mol2 resultantes da busca. No entanto, para carregar esses ligantes no AutoDock4 precisamos separá-lo em arquivos .mol2 diferentes. O script **ex01.csh** irá fazer isso:

6.1.1 Entre na pasta VirtualScreening

```
cd VirtualScreening/
```

6.1.2 Abra o arquivo **ex01.csh** num editor de texto

```
kate ex01.csh &
```

Observe o passo-a-passo do que o script vai fazer (todas as linhas que começarem com # são apenas comentários).

6.1.3 Execute o script **ex01.csh**

```
./ex01.csh
```

Olhe o resultado da execução. As pastas foram criadas, os arquivos .mol2 para cada ligante foram criados e o arquivo ind.pdb foi copiado para a pasta **Ligands/**.

7. Preparando os ligantes: .mol2 para .pdbqt

7.1 No tutorial do dia 2, preparamos o arquivo do ligante através da interface gráfica do ADT. No entanto, para a preparação de dezenas ou centenas de ligantes para o *Virtual Screening* é necessário utilizarmos uma ferramenta automática para isso. É o que o script em Python **prepare_ligand4.py** e o **ex02.csh** fazem.

7.1.1 Abra o arquivo **ex02.csh** num editor de texto

```
kate ex02.csh &
```

Observe o passo a passo do que o script vai fazer (todas as linhas que começarem com # são apenas comentários).

7.1.2 Execute o script **ex02.csh**

```
./ex02.csh
```

Olhe o resultado da execução. Este script converte e prepara todos os arquivos presentes na pasta **Ligands/**, tanto o ind.pdb quanto os ligantes no formato .mol2. Com isso, são criados arquivos no formato .pdbqt para cada ligante. Todas as etapas que realizamos pela interface gráfica do ADT são também executadas aqui, como adição de cargas Gasteiger, assinalar tipos de átomos, adição de átomos de hidrogênio e definir o átomo *root*.

8. Personalizando a biblioteca: determinando os tipos de átomos

8.1 Nesta etapa os scripts em Python **examine_ligand_dict.py** e **ex03.csh** vão criar um arquivo que contém apenas um mapa de *grid* para cada tipo de átomo de todo o conjunto de ligantes. Além disso, irá excluir os ligantes que tenham muitos átomos, tipos de átomos, ligações rotacionáveis, etc.

8.1.1 Abra o arquivo **ex03.csh** num editor de texto

```
kate ex03.csh &
```

Observe o passo-a-passo do que o script vai fazer (todas as linhas que começarem com # são apenas comentários).

8.1.2 Execute o script **ex03.csh**

```
./ex03.csh
```

Observe os tipos de átomos encontrados, bem como informações com relação às características dos ligantes no arquivo **etc/summary.txt**.

9. Preparando o receptor: .pdb para .pdbqt

9.1 Da mesma forma que foi feito com os ligantes, vamos preparar o receptor usando scripts. O formato gerado será o .pdbqt, que difere do .pdb pelo “q” de carga e “t” de tipos de átomos do AutoDock4.

9.1.1 Abra o arquivo **ex04.csh** num editor de texto

```
kate ex04.csh &
```

Observe o passo a passo do que o script vai fazer (todas as linhas que começarem com # são apenas comentários).

9.1.2 Execute o script **ex04.csh**

```
./ex04.csh
```

O arquivo da proteína no format .pdbqt foi criado adicionando os hidrogênios.

10. Preparando o arquivo de parâmetros da grid para da biblioteca de ligantes

10.1 Essa etapa será executada no ADT, uma vez que novos tipos de átomos devem ser inseridos.

10.1.1 Abra o ADT;

10.1.2 Abra o arquivo **x1hvp.gpf** (**Grid → Macromolecule → Open**);

10.1.3 Carregue o arquivo já fornecido com configurações prévias da *grid* **x1hvp.gpf** (**Grid → Open GPF**);

10.1.4 Adicione os outros tipos de átomos que faltam (**Grid → Set Map Types → Directly**), e inclua **CL F S BR** caso não apareçam na caixa de mensagem. Clique em **Accept**;

10.1.5 Salve o arquivo de configuração da *grid* na pasta **Receptor/** (**Grid → Output → Save GPF**) como **x1hvp.gpf** na pasta **Receptor/**;

10.1.6 Feche o ADT;

10.1.7 Abra o arquivo de configurações da *grid* e observe as diferenças:

```
kate x1hvp.gpf &
```

11. Executar o AutoGrid4 para a biblioteca de ligantes

11.1 Nesta etapa o AutoGrid4 calcula os mapas para os diferentes tipos de átomos presentes no arquivo **x1hvp.gpf**.

11.1.1 Abra o arquivo **ex06.csh** num editor de texto

```
kate ex06.csh &
```

Observe o comando usado para chamar o AutoGrid4, o arquivo de *input* e o arquivo de *output*.

11.1.2 Execute o script **ex06.csh**

```
./ex06.csh
```

Entre na pasta **Receptor/** e observe os arquivos gerados pelo AutoGrid4.

12. Validando o Protocolo com um Controle Positivo

- 12.1 Para testar os arquivos de *input* do *docking*, como os arquivos de *grid*, do ligante e da proteína, vamos fazer o *docking* do ligante **ind.pdbqt** para servir como um controle. Primeiro vamos criar um arquivo de parâmetros para o *docking* usando um protocolo bem rápido, apenas para testar os arquivos de entrada. Vamos usar os scripts **prepare_dpf4.py** e **ex07.csh** para executar essa tarefa apenas para o ligante controle.

- 12.1.1 Abra o arquivo **ex07.csh** num editor de texto

```
kate ex07.csh &
```

Observe as etapas para organização dos arquivos em pastas, e o comando usado para chamar o AutoDock4, o arquivo de *input* e o arquivo de *output*. Observe ainda os parâmetros do algoritmo genético, como o número de avaliações e o número de execuções.

- 12.1.2 Execute o script **ex07.csh**

```
./ex07.csh
```

- 12.1.3 Entre na pasta **etc/ind_x1hvp/** e observe os arquivos de *docking* criados, bem como os links para os mapas de tipos de átomos da *grid*.

13. Preparando os diretórios do *docking* e os arquivos de parâmetros de cada ligante da biblioteca

- 13.1 Nesta etapa os mesmo passos executados para o controle **ind.pdbqt** serão usados para cada ligante do ZINC. Com isso, cada ligante terá sua própria pasta, incluindo seus respectivos arquivos nos formatos .dpf e .dlg.

- 13.1.1 Abra o arquivo **ex08.csh** num editor de texto

```
kate ex08.csh &
```

Observe as etapas para organização dos arquivos em pastas, e o comando usado para chamar o AutoDock4, o arquivo de *input* e o arquivo de *output*.

Observe que os parâmetros do *docking* para o *Virtual Screening* são diferentes dos parâmetros usados no controle, são mais robustos uma vez que irá tratar ligantes bastante diferentes do encontrado no cristal.

13.1.2 Execute o script **ex08.csh**

```
./ex08.csh
```

13.1.3 Entre na pasta **Dockings/**, e observe a existência de uma pasta para cada ligante. Entre em qualquer uma e olhe a presença dos arquivos de configuração do *docking*, bem como os links para os mapas de tipos de átomos da *grid*.

14. Executar uma lista de *dockings*

14.1 Através de scripts é possível executar vários cálculos de *docking* inseridos numa fila.

14.1.1 Abra o arquivo **ex09.csh** num editor de texto

```
kate ex09.csh &
```

Observe que o primeiro comando cria um arquivo de fila contendo todos os ligantes no arquivo **docking.list**. Depois ele executa o AutoDock4 para cada linha desse arquivo, ou seja, para cada ligante.

14.1.2 Execute o script **ex09.csh**

```
./ex08.csh
```

14.1.3 Entre na pasta **Dockings/** a existência de uma pasta para cada ligante. Entre em qualquer uma e olhe a presença dos arquivos de configuração do *docking*, bem como os links para os mapas de tipos de átomos da *grid*.

15. Identificando resultados interessantes para serem analisados

15.1 Nesta etapa criaremos uma lista ordenada por energia, iniciando da menor energia do *docking* para cada ligante.

15.1.1 Abra o arquivo **ex10.csh** num editor de texto

```
kate ex10.csh &
```

Este script cria um arquivo texto **summary.txt** com as energias de cada ligante. Depois, cria um novo arquivo **summary.sort** ordenado de forma crescente pelos valores de energia.

15.1.2 Execute o script **ex10.csh**

```
./ex10.csh
```

15.1.3 Abra o arquivo **summary.sort** num editor de texto e observe os ligantes que possuem os menores valores de energia no *docking*.

```
kate summary.sort &
```

Dia 4 – Docking com Outros Alvos

Os arquivos dos complexos 2MCP e 4DFR já foram preparados. Experimente gerar os arquivos necessários para fazer o *docking* e testar diferentes protocolos, alterando principalmente:

16. Graus de Liberdade do Ligante

17. Número de Avaliações

18. Número de Execuções

19. População Inicial

20. Algoritmo Genético não-Lamarckiano (sem Busca Local)

Referências

- ∞ AutoDock4 – G. M. Morris, R. S. Holliday, R. Huey, W. E. Hart, R. K. Belew, A. J. Olson. *Automated Docking Using a Lamarckian Genetic Algorithm and Empirical Binding Free Energy Function*. J. Computational Chemistry. **1998**, 19:1639-1662.
- ∞ PDB – H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. *The Protein Data Bank*. Nucleic Acids Research, **2000**, 28: 235-242.
- ∞ ZINC Database – Irwin and Shoichet. J. Chem. Inf. Model. **2005**, 45(1):177-82.
- ∞ Tutorial AutoDock – G. M. Morris, R. Huey. *Using AutoDock 4 with AutoDockTools: A Tutorial*. **2008**. (<http://autodock.scripps.edu/>).
- ∞ Tutorial AutoDock – C. Weber, R. Huey. *Using AutoDock 4 for Virtual Screening*. **2008**. (<http://autodock.scripps.edu/>).

Links Interessantes

- ∞ PDB2PQR – preparação de proteína como adição de hidrogênios, cálculo de cargas parciais (vários campos de forças) e cálculo de pKa dos resíduos.
Link: <http://pdb2pqr-1.wustl.edu/pdb2pqr/>
- ∞ Corina – calcula a estrutura 3D de pequenas moléculas para download, a partir do SMILE ou do desenho da estrutura em 2D.
Link: http://www.molecular-networks/online_demos/corina_demo
- ∞ Informações sobre o formato PDB:
Link: <http://www.wwpdb.org/documentation/format32/v3.2.html>