

# Virtual Screening Baseado no Ligante com o LiBELa

Alessandro S. Nascimento<sup>1</sup>

<sup>2</sup> Instituto de Física de São Carlos, Universidade de São Paulo. Av. Trabalhador SaoCarlense, 400. Parque Arnold Schimidit. São Carlos, SP, Brazil. 13566-590.

## Resumo

Neste documento, faremos um tutorial sobre a triagem virtual, baseada na busca por ligantes similares no espaço 3D. Na sequência, faremos a seleção de moléculas para a triagem, a triagem e a avaliação dos resultados.

## 1 Introdução

Neste tutorial, faremos a busca de moléculas candidatas a ligantes da enzima *RmlA* usando o substrato desta enzima, a piridoxina, como molécula de busca. Usaremos o programa LiBELa [1] para a busca e analisaremos os resultados usando o programa UCSF Chimera [2].

## 2 Triagem Virtual Baseada no Ligante

Sabemos que aproximadamente de 45% dos fármacos no mercado são endereçados a receptores de membranas [3]. Neste contexto, o desafio em se obter moléculas candidatas é aumentado em função da presença minoritária desta classe de proteínas no PDB. Apesar dos esforços recentes que renderam o Prêmio Nobel de Química em 2012 para Lefkowitz e Kobilka (veja [4], para detalhes), a cristalização de proteínas de membrana ainda é um desafio em biologia estrutural. Em muitos casos, portanto, teremos um projeto, um alvo definido e nenhuma estrutura para usar na triagem virtual. O que pode ser feito em situações como esta?

Uma estratégia que pode ser empregada nestas situações é o emprego da triagem baseada na estrutura de um ligante conhecidamente bioativo. A premissa aqui é que *ligantes similares podem se ligar de modo similar a um mesmo receptor biológico*. Todo o trabalho passa a ser então: (i) como definir similaridade entre ligantes; (ii) como usar esta similaridade em triagem virtual.

Há várias ferramentas capazes de realizar este tipo de comparação (e.g., ROCS, ElectroShape, etc). Uma ferramenta computacional que também permite fazer este tipo de busca é o programa LiBELa, desenvolvido no nosso grupo de pesquisa. Neste contexto, a triagem baseada na estrutura do ligante ativo oferece algumas vantagens:

1. Costuma ser (muito) mais rápida que a triagem baseada no receptor;

2. Dados retrospectivos de enriquecimento costumam revelar capacidade de enriquecimento similar ou maior que aquele observado para os métodos que se baseiam na estrutura do receptor.
3. Dispensa a estrutura do receptor e sua preparação;

Nas seção seguinte, faremos a triagem da base de dados do *ChEMBL Drug Store* usando a estratégia baseada somente no ligante de referência. Vamos ver o que podemos obter como resultado desta análise.

## 3 Preparando o Ambiente

### 3.1 Instalando o LiBELa (Opcional)

O programa LiBELa para Windows está disponível na plataforma GitHub. As versões para Linux e MacOS estarão disponíveis em breve também. Para baixar e instalar, siga as instruções abaixo.

- Faça o download do [arquivo de instalação no GitHub](#), de acordo com o seu sistema operacional.
- Siga as instruções do instalador e escolha um diretório para a instalação. Eu recomendo algo do tipo C:/LiBELa.

Para as etapas deste tutorial, vamos assumir que o programa esteja instalado e rodando em um ambiente Linux. Efetivamente, os comando que usarei aqui são totalmente funcionais no nosso *cluster* local.

### 3.2 Preparação do Conjunto de Moléculas

Para iniciar, vamos buscar as moléculas que vamos usar na nossa campanha de busca, ou triagem virtual. Para esta finalidade, usaremos a base de dados ZINC [5]. O ZINC cataloga milhões de moléculas que podem ser adquiridas comercialmente a as disponibiliza em diversos formatos de arquivo que

podem ser utilizados para o docking ou outros tipos de estudos computacionais.

No caso do nosso *cluster*, a base de dados *ChEMBL Drug Store* já está salva em `/share/data3/asn/ZINC/ChEMBL-DrugStore` e o arquivo `multimol.dat`, que se encontra no mesmo diretório, traz a lista de todos os compostos disponíveis nesta base. O conjunto de arquivo que temos atualmente soma 13.215 compostos. Há outras bases de moléculas no *cluster* além da ChEMBL, como *CheBI*, *ChEMBL21*, *Drug Bank*, *HMDBDrug*, dentre outras. Você pode querer testar outras bases, posteriormente.

## 4 Realizando a Triagem

Nesta etapa vamos fazer a etapa de triagem propriamente dita. Como vamos rodar remotamente no *cluster*, vamos usar a versão do programa em terminal.

- Crie uma pasta onde os arquivos gerados pela triagem serão salvos. Eu criei uma chamada LBVS.
- Nossa *molécula de referência* para a busca de compostos similares será o dTTP. Vamos usar um arquivo com as coordenadas atômicas e cargas para o dTTP. Para isto, vá até o ZINC e faça o download do [arquivo em formato mol2](#). Em seguida, salve o arquivo no diretório local com nome `dTTP.mol2.gz`. O arquivo está compactado, mas pode ser visualizado em programas como o Chimera ou PyMol.
- Vamos gerar um arquivo com as instruções que o programar vai utilizar na busca virtual. Para isto, eu preparei um modelo de arquivo de *input* (`iMcLiBELa.inp`). O modelo encontra-se no apêndice (seção 6). Verifique os endereços dos arquivos de *input* e corrija, onde necessário.
- À medida em que cada molécula vai sendo otimizada, o programa imprime algumas informações na área de Log. A informação que nos interessa neste momento é o último número, que é o índice de similaridade (SI) de Hodgkin. Este número deve ser algo entre -1.0 e 1.0, sendo -1.0 uma perfeita antissimilaridade e 1.0 uma perfeita similaridade. Estamos interessados nas moléculas com SI mais próximos de 1.
- No modelo de *input*, eu impus um critério de 0.75 (*threshold*). Isto quer dizer que o programa vai escrever em um arquivo apenas as moléculas que tiverem um  $SI \geq 0.75$ .

- Finalmente, estamos quase prontos para a submissão do *job* no *cluster*. Para isto, vamos criar um arquivo que instrui o sistema de submissão de *jobs* sobre como rodar nossa busca. Para isto, eu gerei um arquivo de submissão (`iMcLiBELa.sub`) e que está na seção 7. Neste arquivo, requisitamos 48 processadores do *cluster* para a submissão do *job*.
- O *job* pode ser submetido com o comando `qsub iMcLiBELa.sub`. O comando `qstat` permite monitorar o *job* e ver seu *status*: `qw = queue waiting` e `r = running`. No teste que fiz, o programa levou 97 minutos para triar 13.215 moléculas, ou seja, triou cerca de 136 moléculas por minuto.

## 5 Análise do *Screening* com o UCSF Chimera

O LiBELA vai gerar  $N$  arquivos, com  $N$  igual ao número de *jobs* paralelos. Ele faz isto para evitar que *jobs* diferentes escrevam no mesmo arquivo e causem problemas. A primeira etapa da análise dos resultados deve ser, portanto, concatenar os resultados em um único arquivo:

- No terminal, use o comando: 

```
zcat dTTP_LBDD-ChemBL*_dock.mol2.gz > dTTP_LBDD-ChemBL_dock.mol2.
```

 Em seguida, utilize o comando `gzip dTTP_LBDD-ChemBL_dock.mol2` para compactar o arquivo de saída. Este arquivo final (`dTTP_LBDD-ChemBL_dock.mol2.gz`) deve ser copiado para análise dos resultados no programa Chimera.
- Abra o programa Chimera. Em seguida, clique "*File*--> "*Open*" e selecione o arquivo em formato MOL2 `dTTP.mol2.gz`.
- Agora vamos analisar efetivamente os resultados da triagem. O Chimera tem uma ferramenta muito útil para esta finalidade: Vá em *Tools* -> *Surface/Binding Analysis* -> *ViewDock*. Na janela que se abre, selecione o arquivo `dTTP_LBDD-ChemBL_dock.mol2.gz`. Quando perguntado sobre o tipo de arquivo, selecione o *Dock 4, 5 or 6* e clique em OK.
- O ViewDock abrirá uma janela para a visualização dos resultados e carregará a primeira molécula no visualizador. Na janela do ViewDock, clique em *Column* -> *Show* -> *RMSD*. Os índices de similaridades da triagem serão mostrados para cada molécula que foi triada. Clicando sobre o *RMSD*, no cabeçalho da tabela, as moléculas serão ordenadas por seus SI e podemos inspecionar uma a uma, bem

como analisar a similaridade estrutural dom a molécula de referência dTTP.

- Nosso objetivo agora é inspecionar cada uma das moléculas selecionadas e encontrar moléculas com **boa possibilidade de interação** com o nosso alvo. Elas devem preservar boa similaridade com a molécula de referência, com grupos polares em posições similares.

Analise as molécula que foram triadas e tente eleger 20-30 moléculas que melhor satisfaçam a este critério de escolha. Uma forma de selecionar as moléculas é clicar em *Purged* para as moléculas que você não deseja manter na lista final, de forma que a lista final tem as demais como *viáveis*. As moléculas marcadas como *purged* não apagadas. Só não são mostradas mais na lista inicial. Outra informação importante, o Chimera permite salvar a *sessão*, isto é, salvar o estado em que se encontra, com moléculas carregadas e análises realizadas, para continuar o trabalho depois: *File -> Save Session As*.

## Referências

- [1] Heloisa dos Santos Muniz and Alessandro S. Nascimento. Ligand- and receptor-based docking with LiBELa. *Journal of Computer-Aided Molecular Design*, 29(8):713–723, 2015.
- [2] E F Pettersen, T D Goddard, C C Huang, G S Couch, D M Greenblatt, E C Meng, and T E Ferrin. UCSF chimera - A visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25:1605–1612, 2004.
- [3] Robert Adams, Catherine L Worth, Stefan Guenther, Mathias Dunkel, Robert Lehmann, and Robert Preissner. Binding sites in membrane proteins – Diversity, druggability and prospects. *European Journal of Cell Biology*, 91(4):326–339, 2012.
- [4] Daniel M. Rosenbaum, Søren G. F. Rasmussen, and Brian K. Kobilka. The structure and function of G-protein-coupled receptors. *Nature*, 2009.
- [5] J J Irwin, T Sterling, M M Mysinger, E S Boltad, and R G Coleman. ZINC: A Free Tool to Discover Chemistry for Biology. *J Chem Inf Model*, 52(7):1757–1768, 2012.

## 6 Apêndice: Arquivo de *Input* para a Triagem LB

```
# mode

mode                dock
dock_parallel      yes
parallel_jobs      2

#
# input files
#

rec_mol2            dTTP.mol2.gz
lig_mol2            dTTP.mol2.gz
reflig_mol2        dTTP.mol2.gz
mol2_aa            no
multifile           /share/data3/asn/ZINC/ChEMBL-DrugStore/multimol.dat

#
# force field parameters
#

scoring_function    3
dielectric_model    r
diel                1.0
deltaij             0.0
deltaij_es          0.0
use_grids           no
grid_spacing        0.4
grid_box            30.0  30.0  30.0
write_grids         McGrid
delphi_grid         ../grids/phimap.phi
delphi_gsize        75
use_delphi          no
solvation_alpha     0.2
solvation_beta      -0.005
ligand_energy_model MMFF94
atomic_model_ff     GAFF
LJ_sigma 1.2

#
# Optimization
#

search_box          8.0  8.0  8.0
minimization_tolerance 1.0e-4
minimization_delta  1.0e-4
dock_min_tol        1.0e-4
minimization_timeout 45
overlay_optimizer    mma
energy_optimizer     none
ignore_h            no
deal                no
elec_scale           1.0
vdw_scale            1.0
sort_by_energy       no

#
```

```
# SA Options
#

sa_start_temp      100
temperature        0.001
sa_steps           1000
cushion            1.0
rotation_step      10.0
sa_mu_t            1.1

#
# output
#

output_prefix      dTTP_LBDD_ChEMBL
write_mol2         yes
use_writeMol2_score_cutoff yes
writeMol2_score_cutoff 0.75
use_writeMol2_energy_cutoff no
writeMol2_energy_cutoff 10.0

#
# flexible ligands
#

generate_conformers yes
number_of_conformers 10
conformers_to_rank 1
```

## 7 Apêndice: Arquivo de *Input* para a Submissão de *Job* Paralelo no *Cluster*

```
#!/bin/bash
#$ -S /bin/bash
#$ -N LBVS
#$ -pe orte 48
#$ -cwd
#$ -o libela.out
#$ -j y
#$ -V
ulimit
source /share/apps/iMcLiBELa/LiBELa.sh
libela=/share/apps/iMcLiBELa/bin/McLiBELa.mpi
time /share/apps/openmpi/bin/mpirun --mca btl tcp,self --prefix /share/apps/openmpi \
    -display-allocation -n $NSLOTS $libela -i iMcLiBELa.inp
```