Virtual Screening Workshop: Parte II

Isabella Guedes[1] e Alessandro S. Nascimento[2]
[1] *Laboratório Nacional de Computação Científica - LNCC/MCT. Av. Getúlio Vargas 333 - 1A11 - Quitandinha, Petrópolis, RJ, Brazil. 25651-075.*
[2] *Instituto de Física de São Carlos, Universidade de São Paulo. Av. Trabalhador SaoCarlense, 400. Parque Arnold Schmidit. São Carlos, SP, Brazil. 13566-590.* (Dated:)

Resumo

Nesta série de documentos, faremos um tutorial sobre a triagem virtual, passando pela preparação de um alvo, calibração do modelo e avaliação dos resultados. Na sequência, faremos a seleção de moléculas para a triagem, a triagem e a avaliação dos resultados. Estes documentos foram preparados como parte do material da IX EMMSB 2018.

1 Introdução

Nesta segunda parte, faremos a análise dos dados da calibração do nosso sistema. Adicionalmente, faremos a preparação e triagem da base de uma base de dados de repropósito na nossa campanha de VS.

2 Análise dos Resultados de Calibração

A forma mais precisa de se analisar um resultados é checar o **enriquecimento**. Aqui, estamos interessados em ver o quanto o nosso método de docking pontua de forma mais favorável os ligantes verdadeiros em relação aos *decoys*. A forma de fazer isto é através de uma curva ROC. O script python ROC.py deve ser capaz de fazer isto.

Primeiramente, vamos extrair os dados necessários do arquivo de saída do LiBELa: `$ more AMPC.log | grep ZINC > ampc.dat`. Com este comando, vamos extrair as informações do arquivo de log do LiBELa. No arquivo que geramos, as seguintes informações estão presente: número da molécula, nome, nome do resíduo, valor da função de superposição, energia eletrostática e de van der Waals, dessolvatação do receptor e do ligante (neste caso, zero porque não usamos o modelo de dessolvatação), energia total, número do confôrmero e indicadores de otimização.

A sintaxe para o uso é `$ ROC.py -a [número de ativos] -d [número de decoys] -i [arquivo de input] -o [arquivo de output]`. Neste caso, teremos `$ ROC.py -i ampc.dat -o roc_auc.dat -a 13 -d 390`. Nosso teste em pequena escala revela que temos um enriquecimento elevado, com logAUC ajustado de 24.4% e AUC de 68.6%, como mostra a Figura 1. Nosso modelo é robusto e está funcionando bem!

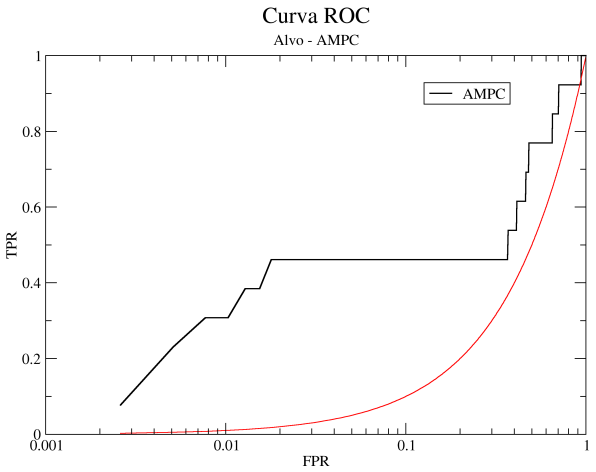

Figura 1: Curva ROC para o alvo AMPC.

Vamos avaliar outros dois modelos que foram gerados para a mesma estrutura cristalográfica (PBD 1L2S) e mesmo ligante de referência. Você consegue identificar o que está errado com estes modelos?

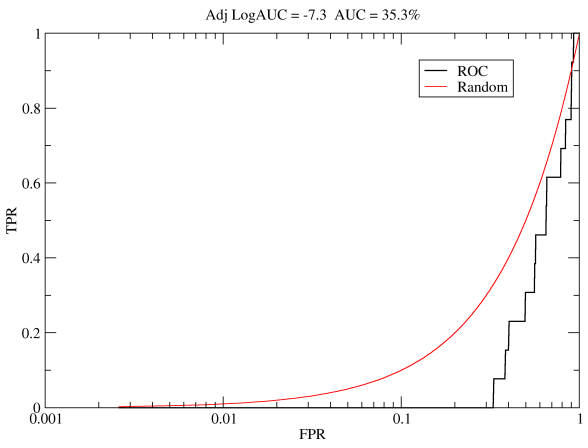

Figura 2: Curva ROC para o alvo AMPC, versão 0. A mesma estrutura cristalográfica foi empregada para a preparação do alvo.

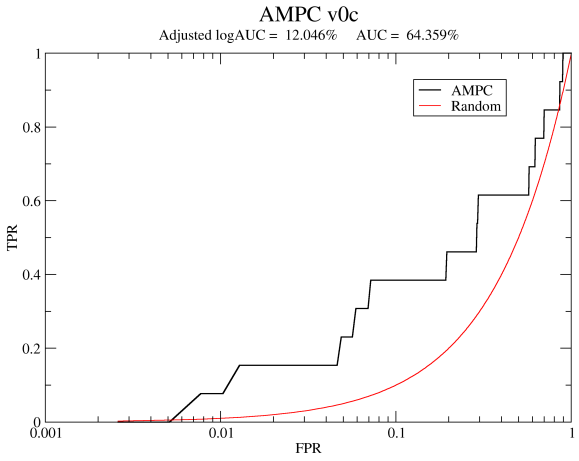

Figura 3: Curva ROC para o alvo AMPC, versão 0c. A mesma estrutura cristalográfica foi empregada para a preparação do alvo.

3 Preparação da Base de Dados para o *Screening*

Neste experimento de triagem, vamos usar a estratégia de repropósito [1, 2]. A estratégia de repropósito baseia-se em empregar fármacos já aprovados e em uso para novas aplicações, i.e., buscando interações em novos alvos. Os fármacos em repropósito podem servir como *chemical probes* ou como novos fármacos e têm como vantagem a segurança já estabelecida. As referências [1, 2] mostram exemplos recentes de repropósito de fármacos para o tratamento da Malária.

- Primeiramente, crie uma pasta chamada VS dentro do diretório EMMSB_VS. É nela que vamos trabalhar. Dentro desta pasta crie outras duas pastas: *mol2* e *docking*.
- Vá ao site do ZINC, clique em *Catalogs* e, depois, em *Subsets*. Clique em *FDA* e, depois, em *DrugBank FDA only*.
- Clique em *Catalog items*. Na janela que se abre, clique no botão com uma seta apontada para baixo (*Download all as)* e selecione TXT.
- Vamos extrair os códigos ZINC destes compostos: `$ more dbfda-items.txt | awk '{print $2}'`. Nesta lista, temos 1657 compostos.
- Com os Ids dos compostos, faça a busca no ZINC, pedindo o *Output Format* como MOL2. Salve este arquivo na pasta VS como FDA.mol2. Dados os diferentes estados de protonação, temos um pouco mais que 2.000 compostos neste arquivo. Vamos gerar os múltiplos MOL2 na pasta mol2: `cd mol2` e, em seguida, `$ babel -imol2 ../FDA.mol2 -omol2 fda..mol2 -m .` Para economizar espaço em disco, `$ gzip *.mol2`.

4 Preparação Para a Docagem

Vamos deixar tudo pronto para a triagem de compostos. Temos algo em torno de 2150 compostos. Mas vamos docar um conjunto menor, talvez 300 compostos. Podemos usar o script em python *picker.py* para gerar uma lista aleatória de 300 moléculas para usarmos no docking. Na pasta *docking* execute `$ picker.py`. Ao final ele deve gerar uma lista de 301 compostos para o docking no arquivo *multimol.dat*. Novamente, vamos precisar de um arquivo de input. Um modelo está mostrado no apêndice deste texto.

Na pasta *VS/docking*, certifique-se de que esteja presente o arquivo de input (e.g., libela.inp) e o arquivo multimol.dat, que contém a lista de moléculas que devem ser docadas. A execução do LiBELA pode explorar o número de processadores/*threads* disponível no computador. Verifique quantos threads podem ser executados paralelamente e adicione este número na keyword `parallel_jobs`. No exemplo abaixo, temos `parallel_jobs 4`, para 4 threads disparados em paralelo.

Finalmente, vamos disparar o job com o comando `$ time McLiBELa.openMP libela.inp`. Esta execução deve levar algo em torno de 1 hora em 2 *threads*.

Referências

[1] Yun Chen, Claribel Murillo-Solano, Melanie G Kirkpatrick, Tetyana Antoshchenko, Hee-Won Park, and Juan C Pizarro. Repurposing drugs to target the malaria parasite unfolding protein response. *Scientific Reports*, 8(1):10333, 2018.

[2] Nila Madassary Pazhayam, Jyoti Chhibber-Goel, and Amit Sharma. New leads for drug repurposing against malaria. *Drug Discovery Today*, 2018.

5 Apêndice: Arquivo de Input para o *Screening*

```
# mode
mode                    dock
dock_parallel           yes
parallel_jobs           4

# input files

rec_mol2                ../../ampc_rec.mol2.gz
lig_mol2                ../../ampc_xtallig.mol2.gz
reflig_mol2             ../../ampc_xtallig.mol2.gz
mol2_aa                 no
multifile               ./multimol.dat

# force field parameters

scoring_function        3
dielectric_model        r
diel                    1.0
deltaij                 0.0
deltaij_es              0.0
use_grids               yes
use_delphi              no
grid_spacing            0.3
grid_box                30.0    30.0    30.0
write_grids             McGrid
solvation_alpha         0.25
solvation_beta          -0.005

# Optimization

search_box              12.0    12.0    12.0
minimization_tolerance  1.0e-6
minimization_delta      1.0e-6
dock_min_tol            1.0e-6
minimization_timeout    30
overlay_optimizer       ln_auglag
energy_optimizer        direct
ignore_h                no
deal                    no
elec_scale              1.0
vdw_scale               1.0
sort_by_energy          no

# output

output_prefix           AMPC_FDA
write_mol2              yes

# flexible ligands

generate_conformers     yes
number_of_conformers    10
conformers_to_rank      1
conformer_generator     GA
conformer_min_steps     1000
```