

A influência dos parâmetros no *docking* com o QuickVina 2

Caio Cedrola Rocha

GET Engenharia Computacional, Universidade Federal de Juiz de Fora

12 de maio de 2020

1 Introdução

Um dos principais desafios na descoberta de medicamentos é a identificação de fármacos que poderiam combater certo patógeno pesquisado. Nesse contexto, a atuação do composto farmacológico consiste em estabelecer uma ligação estável com a proteína, geralmente uma enzima, do vírus ou de outro patógeno, ligação essa a qual modulará a atividade do agente patogênico.

A atracagem molecular, também conhecida como *docking* molecular, é um método computacional largamente utilizado na detecção de potenciais compostos com propriedades farmacológicas. A metodologia do *docking* consiste em, dados uma proteína alvo e um ligante, prever a afinidade entre esses reagentes, assim como as melhores orientações espaciais desse ligante no sítio de ligação da molécula alvo, referido como sítio ativo. Como forma geral, na maioria das técnicas de atracagem, esse sítio ativo é conhecido, então melhor é a eficiência do experimento de *docking*[1]. Aliada ao *docking*, que trabalha com pares proteína-ligante, emprega-se a triagem virtual, ou *virtual screening*, a qual consiste no uso de abordagens de atracagem molecular em grandes bancos de dados de compostos bioquímicos, com o objetivo de identificar esses pares para posterior desenvolvimento de medicamentos[2].

Dentro da conjuntura do *docking*, surgem as funções de pontuação, que são algoritmos de atracagem molecular apoiados sobre bases principalmente teóricas ou empíricas, que visam quantificar a afinidade de ligação entre dois compostos fornecidos como entrada. Durante o desenvolvimento deste relatório, serão detalhadas as funções de pontuação majoritariamente empíricas. Os algoritmos de pontuação intentam encontrar e ranquear as melhores poses relativas às possíveis conformações espaciais do ligante dentro do sítio ativo (previamente determinado) da molécula. Para tal, essas funções exploram a disposição do composto no espaço, por meio da rotação de ligações interatômicas, e realizam variados cálculos com base nos recursos lidos da molécula e nos parâmetros fornecidos[1].

2 Experimento

O estudo apresentado a seguir refere-se ao algoritmo de pontuação do *QuickVina 2*[3], baseado no *AutoDock Vina*[4], versão 1.1.2, o qual utiliza um algoritmo genético de otimização global por pesquisa local iterativa. Os algoritmos genéticos são amplamente empregados no *docking* molecular, e se baseiam nas teorias de evolução e de seleção natural, de forma que promovem mutações na estrutura do ligante e otimizações locais de energia a cada passo da iteração, até que a energia da molécula converja para o mínimo global[5].

Como um dos fatores mais importantes em campanhas de atracagem molecular tem-se a qualidade da estrutura cristalográfica, ou seja, a estrutura tridimensional do cristal de entrada. Portanto, como *dataset* para o experimento, utilizou-se o banco de dados *PDBbind*[6], o qual contém compostos curados do PDB (Protein Data Bank), principal *database* de complexos bioquímicos. O conjunto de dados utilizado foi o *core dataset* versão 2013 extraído do *PDBbind*, contendo 195 complexos proteína-ligante.

O experimento consistiu em realizar *redockings* para cada um desses complexos, utilizando diferentes parâmetros. Dessa maneira, os objetivos finais foram avaliar a capacidade da metodologia usada em reproduzir as estruturas dos cristais dos ligantes e em classificar as poses dos *dockings*, de forma que aquelas mais próximas da conformação cristalina seriam as melhor qualificadas. Todos os arquivos utilizados, assim como o [notebook](#)[7] aqui analisado, podem ser encontrados em um [repositório](#)[8] cujo endereço consta na seção de referências ao final deste relatório.

3 Protocolo

Os arquivos estruturais dos receptores foram preparados com a função *prepare_receptor4.py* do *mglttools*[9], com fusão de cargas, hidrogênios não polares, pares iônicos e resíduos de água removidos. Em seguida, os arquivos estruturais dos ligantes também foram pré-processados com o *mglttools*, utilizando a função *prepare_ligand4.py*, com fusão de pares iônicos.

Quanto à definição de eixos rotáveis no ligante, todas as ligações exceto aquelas que continham aminas ou amidas foram especificadas como rotáveis. Além disso, o espaço de busca conformacional (caixa de ligação) foi calculado por um *script* caseiro, que determinou a caixa como sendo, a partir do centro geométrico do ligante, igual às dimensões máximas da molécula (decompostas em eixos x, y, z) mais um espaço extra de 2 Å em cada direção. É importante ressaltar que, apesar de não serem as dimensões mínimas recomendadas pelos desenvolvedores do *AutoDock Vina*, decidiu-se assim visando reduzir o custo computacional do experimento. Porém, como será detalhado na seção de conclusão, essa decisão provou-se imprudente, pois custou uma considerável perda de acurácia nos resultados.

Por fim, o principal parâmetro opcional do *AutoDock Vina* é a exaustividade. Resumidamente, a atracagem molecular engloba um número de repetições de cálculos, cada qual promove um pequeno desequilíbrio do mínimo local em que a busca conformacional se encontra, fato que a converge para o mínimo global. Sendo assim, a exaustividade controla o número de repetições dessa busca. Por consequência, o acréscimo da exaustividade aumenta linearmente o tempo despendido na função de busca e diminui a probabilidade de não encontrar o mínimo global[10]. Como protocolo, os experimentos foram reproduzidos em triplicata com exaustividades 8 e 32.

	Complex	Initial Pose	Pose	Program	Exhaustiveness	Run	Energy	RMSD
0	10gs	crystal	1	qvina	8	1	-7.900	2.502
1	10gs	crystal	2	qvina	8	1	-7.000	1.646
2	10gs	crystal	3	qvina	8	1	-6.900	4.382
3	10gs	crystal	4	qvina	8	1	-6.900	4.231
4	10gs	crystal	5	qvina	8	1	-6.800	3.990

Tabela 1: Dados e resultados para cada protocolo utilizado

Como métrica de avaliação da qualidade das poses de saída de cada experimento de *docking*, definiu-se que um resultado é considerado bem representado se o seu Desvio Médio da Raiz Quadrada em relação à pose nativa é menor ou igual a 2 Å[11]. Esse método, conhecido como RSMD, calcula o desvio médio de cada átomo da molécula de saída para com o ligante original, seguindo a seguinte fórmula:

$$RMSD'_{ab} = \sqrt{\frac{1}{N} \sum_i \min_j r_{ab}^2} \quad [4]$$

Logo, tendo como referência a conformação de entrada, a qual é conhecida experimentalmente, em laboratório, de ser bioativa, quanto mais próximos os resultados do *docking* dessa pose, melhores são esses resultados, pois mostra-se que o algoritmo de *docking* consegue reproduzir melhor a realidade.

Em vista disso, o RMSD foi calculado em relação as coordenadas do cristal usando a função *intra_rms* com o programa PyMOL[12] para os cálculos. Um resultado é considerado bem representado se o seu RMSD em relação à pose nativa é menor ou igual a 2 Å[11].

4 Resultados e Discussão

Após realizados os experimentos, obtiveram-se as seguintes conclusões:

4.1 Qualidade das predições

Considerando uma pose por cada complexo, a porcentagem de compostos obtidos com RMSD abaixo do limite predefinido de 2 Å mostrou-se satisfatória, com 51.78% enquadrados dentro desse limiar. Além disso, as poses foram classificadas em boas, médias e ruins ($RMSD \leq 1.5$, $1.5 < RMSD \leq 2.5$ e $RMSD > 2.5$, respectivamente), das quais 34.9% foram boas, 33.7% foram médias e 31.4% foram ruins, portanto uma proporção equilibrada.

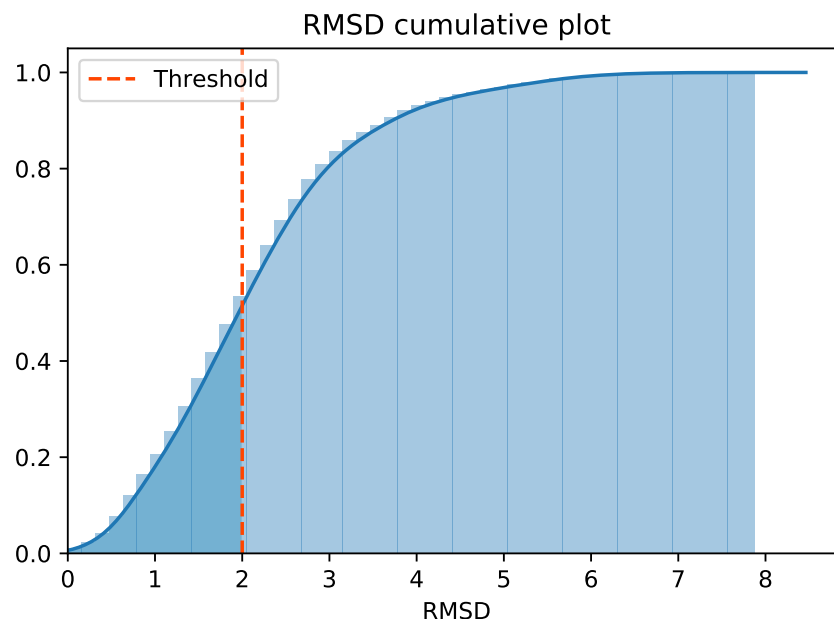


Figura 1: Gráfico cumulativo do RMSD. O gráfico mostra a porcentagem de poses com RMSD menor ou igual ao valor correspondente de RMSD no eixo x. A área azul escura indica aqueles compostos que ficaram incluídos dentro do limite de $RMSD < 2$.

4.1.1 Influência da exaustividade

Ao aumentar-se a exaustividade, esperava-se que haveria uma maior qualidade das poses, pois diminuem-se as chances de a função de pontuação produzir falsos positivos[10].

Porém, como mostra a tabela 2, não foi o que se constatou, pois a proporção da qualidade das poses é equilibrada independentemente da exaustividade. Uma possível explicação é que, como os compostos fornecidos de entrada já são os de menor energia, a busca pelas conformações foi influenciada pelo ligante nativo, e a exaustividade não possuiu um grande impacto no resultado final, o qual já era bem próximo ao ideal.

Exhaustiveness	Quality	
	Quality	
8	bad	1149
	good	1257
	medium	1282
32	bad	1177
	good	1238
	medium	1299

Tabela 2: Qualidade das predições ordenadas por exaustividade.

4.1.2 Influência das replicatas

A repetição do experimento em triplicata produziu resultados levemente diferentes a cada vez, como mostra a tabela 3, como o esperado, porque, a função de busca explora um espaço amostral diferente a cada execução, devido à sua característica estocástica de utilizar uma semente randômica para o início da busca. Dessa maneira, obtiveram-se mais conformações classificadas como adequadas pelo programa.

Exhaustiveness	Run	Quality	
		Quality	
8	1	bad	390
		good	404
		medium	451
	2	bad	371
		good	434
		medium	411
	3	bad	388
		good	419
		medium	420
32	1	bad	380
		good	415
		medium	447
	2	bad	409
		good	402
		medium	436
	3	bad	388
		good	421
		medium	416

Tabela 3: Qualidade das predições agrupadas pela exaustividade e pelo número de execução.

4.2 Taxas de sucesso

O experimento englobou 195 complexos, 1 variável (exaustividade) com dois valores 8 e 32, e 3 repetições, o que resultou em um total de 1770 atracagens moleculares realizadas, cada qual produzindo até 10 poses como saída. Como nem toda execução produz o número máximo de poses especificado, obtiveram-se 7402 resultados, o que representa uma média de, aproximadamente, 6 conformeros por *docking*. Dessa forma, foram calculadas taxas de sucesso em diferentes cenários, explicitados a seguir.

4.2.1 Taxas de sucesso por execução

Tendo em vista o sucesso do programa em cada repetição do experimento, definiu-se como sucesso toda simulação que produziu pelo menos uma pose com RSMD menor que 2 Å, dentre as 10 poses consideradas. Logo, a taxa de sucesso calculada é a razão entre o número computado de *dockings* bem sucedidos e o número total de *dockings*.

Nesse contexto, a exaustividade não influenciou a qualidade dos resultados, assim como também não interferiu na taxa de sucesso. Esse fato é um tanto inesperado, já que tal parâmetro é o principal componente de controle do número de otimizações locais, dada a natureza iterativa do algoritmo de busca conformacional. Hipoteticamente, esse resultado deriva do baixo grau de liberdade dos ligantes para girarem dentro do sítio de ligação, ou seja, o bolsão de ligação definido (*binding box*) não foi grande suficientemente para permitir que o composto gire livremente dentro desse sítio. Alternativamente, também é possível que a função de busca foi parcial em relação às poses-solução do *docking*, pois a orientação espacial dada como entrada foi, já de início, a conformação do cristal, que é aquela que possui bioatividade comprovada em laboratório. Desse modo, o algoritmo de busca sempre convergia para a solução correta afinal, independentemente do valor de exaustividade especificado.

Já a realização do experimento em triplicata produziu leves oscilações na taxa de sucesso para cada execução, o que é justificado pela natureza estocástica do algoritmo. Logo, comprova-se a importância da repetição dos experimentos para a obtenção de resultados mais apurados.

Exhaustiveness	Run	Success Rate
8	1	0.841
	2	0.821
	3	0.841
	Overall	0.834
32	1	0.831
	2	0.810
	3	0.862
	Overall	0.834

Tabela 4: Taxas de sucesso para cada execução, ordenado por exaustividade

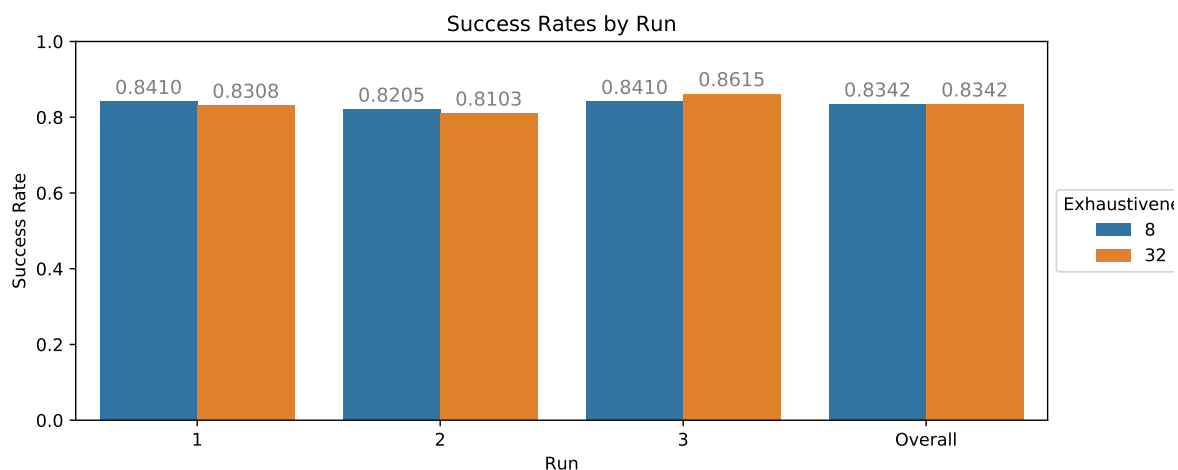


Figura 2: Gráfico de barras da taxa de sucesso por execução.

4.2.2 Taxa de sucesso cumulativa

O objetivo primário de uma simulação de *docking* molecular é que a função de pontuação consiga reproduzir a estrutura de menor energia do ligante. Logo, frequentemente, deseja-se que a primeira conformação de saída do programa se assemelhe ao resultado esperado. Porém, quando é permitida uma maior flexibilidade na consideração dos resultados, pode-se utilizar a taxa de sucesso cumulativa, que considera, a cada passo, a replicata atual e as anteriores. Essa taxa cresce à medida que mais poses são consideradas, como se verifica na tabela 5.

Exhaustiveness	Success Rate	
	Number of Poses	
8	1	0.699
	2	0.766
	3	0.786
32	1	0.708
	2	0.783
	3	0.797

Tabela 5: Média das taxas de sucesso entre as três execuções considerando-se até a terceira pose

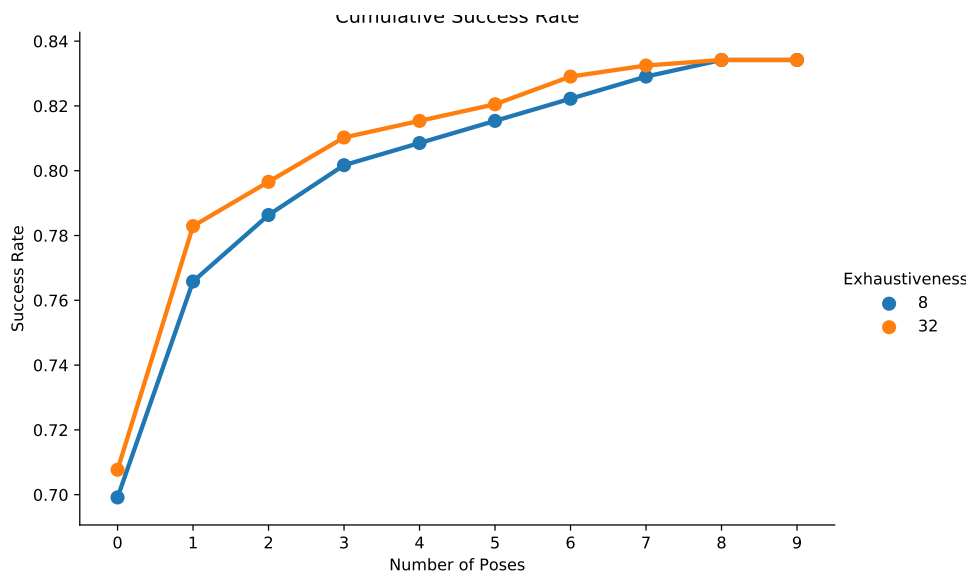


Figura 3: Gráfico da taxa de sucesso cumulativa. Observa-se o aumento da taxa de sucesso proporcionalmente ao número de poses consideradas.

4.3 Curva ROC e AUC

No contexto do aprendizado de máquina, um recurso útil para a avaliação do desempenho de algoritmos é a curva ROC (Receiver Operating Characteristic, ou Característica de Operação do Receptor, em tradução livre do inglês). Essa curva é uma função de probabilidade que tem como eixo X a taxa de falsos positivos e como eixo Y a taxa de verdadeiros positivos. Como nota de esclarecimento, verdadeiros positivos são elementos de saída de uma função de *machine learning* que foram classificados corretamente pelo algoritmo como positivos. Analogamente, falsos positivos foram categorizados erroneamente como positivos, verdadeiros negativos foram identificados corretamente como negativos, e falsos negativos foram classificados erroneamente como negativos. Em adição à curva ROC, recorre-se à área sob a reta da curva ROC, denominada AUC (Area Under Curve), a qual é uma métrica utilizada para mensurar a capacidade do algoritmo em distinguir classes de um modelo, resumindo-a em um número. Uma área sob a curva acima de 0.5 indica que a função de aprendizado de máquina consegue discriminar positivos de negativos, assim como

uma área abaixo de 0.5 mostra que essa função realizou uma associação inversa à esperada entre as variáveis e uma área com valor próximo ou igual a 0.5 aponta que o algoritmo não estabeleceu nenhuma relação entre essas variáveis.

Logo, para este experimento, fez-se uso da curva ROC e da AUC para avaliar o *docking power*, ou potencial de ranqueamento, da função de pontuação estudada. Desse modo, considerou-se como positivas as três melhores poses de saída de cada *docking*, seguindo o RMSD como segunda métrica de agrupamento, conforme detalhado na tabela a seguir.

Classification	TP	FP	TN	FN
Pose	[1, 2, 3]	[1, 2, 3]	> 3	> 3
RMSD	< 2.0	>= 2.0	< 2.0	>= 2.0

Tabela 6: Parâmetros de classificação dos compostos. TP, FP, TN, FN representam, respectivamente, verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo.

As curvas ROC e as áreas sob a curva obtidas respectivamente através das funções *roc_curve* e *auc*, ambas da biblioteca *scikit-learn* em Python, encontram-se apresentadas no gráfico e na tabela abaixo.

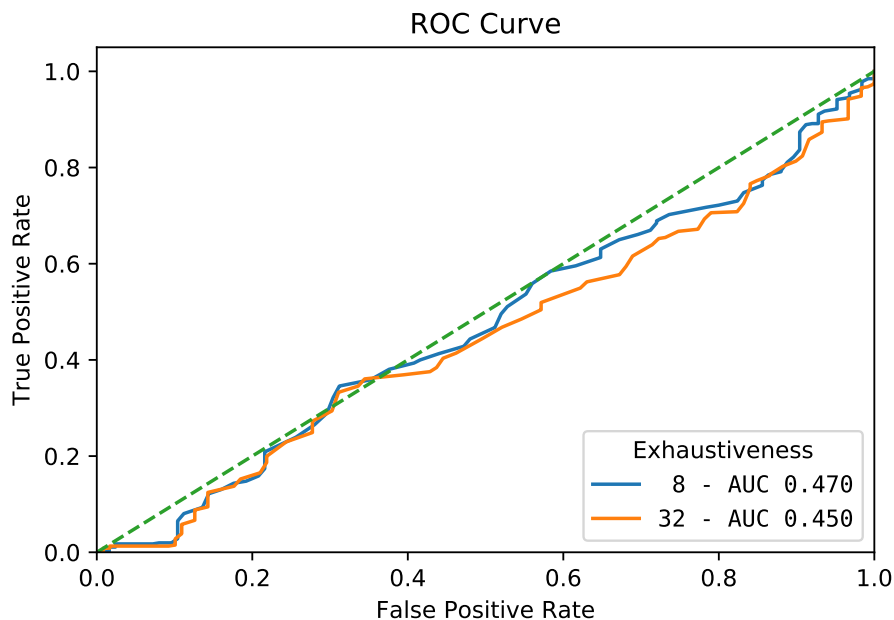


Figura 4: Gráfico das curvas ROC para cada valor de exaustividade.

Exhaustiveness	AUC
8	0.470
32	0.450

Tabela 7: Resultados da Área Sob a Curva separados pela exaustividade.

Como observa-se, as baixas razões de AUC mostram que, apesar das altas taxas de sucesso demonstradas previamente neste relatório, a função de pontuação não conseguiu um resultado satisfatório em relação à classificação das conformações de saída em positivas e negativas, ou seja, as energias calculadas pelo programa não refletem os valores de RMSD entre essas poses e o ligante original, portanto não refletem a qualidade esperada. Esse resultado trouxe dúvidas em relação à validade do protocolo utilizado para a preparação dos arquivos estruturais das moléculas. Após uma revisão dessa etapa, descobriu-se que o raio máximo permitido para o giro do ligante no sítio ativo não possibilitava que a função de pontuação explorasse todas as conformações plausíveis da molécula, o que resultou em resultados incompletos. É importante ressaltar que esse cálculo das dimensões de ligação foi feito por um algoritmo caseiro que não tem alguma relação com o programa de *docking* avaliado. Após tal constatação, o algoritmo de cálculo das dimensões da atracagem foi corrigido.

5 Conclusão

Tendo em vista o objetivo de analisar a influência dos parâmetros da atracagem molecular com o *software QuickVina 2*, estabeleceram-se como métricas de avaliação a qualidade das poses dos ligantes, as taxas de sucesso e a curva ROC/AUC. Inicialmente, qualificou-se a estrutura de saída das simulações, que deveriam ser, para cada ligante, próximos de no mínimo 2 Å da pose nativa, ou original. Logo, os dados obtidos já não refletiram os resultados esperados, pois a exaustividade não alterou a proporção de poses consideradas boas, médias ou ruins. Em seguida, determinou-se taxas de sucesso para cada situação testada, de acordo com o RMSD, o qual deveria valer menos de 2 Å em relação ao ligante original para uma pose ser considerada como bem-sucedida. Assim, as taxas de sucesso, apesar de mais altas do que o provável, seguiram as tendências esperadas, ou seja, aumentaram de acordo com o aumento da exaustividade e variaram minimamente para diferentes execuções. Como última métrica de ponderação, desenhou-se a curva ROC e calculou-se a AUC, as quais indicaram uma clara inconsistência da distribuição dos dados com as taxas de sucesso obtidas. Esse fator despertou ceticismo quanto ao protocolo de preparação das simulações, precisamente na definição das dimensões da caixa de ligação. Após revisão de um [artigo\[13\]](#) publicado a respeito do cálculo do tamanho ótimo do sítio de ligação para o docking com o *AutoDock Vina*, concluiu-se que essa etapa foi, provavelmente, o fator que impactou negativamente os resultados, pois as dimensões mínimas recomendadas não foram respeitadas.

Dessa maneira, enquanto não é possível tirar alguma conclusão sobre a influência dos parâmetros na simulação, em virtude da parcialidade do experimento conduzido, é possível explicar o baixo desempenho da função de pontuação. Logo, como não se permitiu ao programa de *docking* explorar todas as poses razoáveis do ligante no sítio de ligação, as conformações de saída repetidamente careciam da pose esperada. Em consequência, apesar das altas taxas de sucesso, as quais indicavam que havia um número razoável de compostos com o RMSD abaixo de 2.0 Å, a curva ROC apontou a inconsistência dos dados, pois exibiu uma distribuição quase aleatória dos resultados.

Além deste experimento, foram conduzidos outros dois ensaios relacionados. O primeiro foi a otimização local dos ligantes em seus respectivos receptores, utilizando as funções de pontuação do *QuickVina 2*, do *AutoDock Vina*, do *Smina* e do *Smina vinardo*[14], todos na versão 1.1.2. Logo, os dados obtidos foram comparados com os valores experimentais de $\log K_i/K_d$. Ademais, o segundo ensaio foi o *rescoring* dos resultados do primeiro experimento de *docking*, em sua primeira execução com exaustividade 8. Dessa forma, após completados os *rescores*, os valores obtidos também foram comparados com os dados experimentais de $\log K_i/K_d$. Tais ensaios não serão discutidos aqui, porém é possível encontrá-los no repositório do projeto[15][16].

Por fim, decidiu-se repetir um experimento similar, com o objetivo de avaliar diferentes funções de pontuação, utilizando o mesmo *dataset* do *PDBbind*. Cabe dizer que a função de cálculo das dimensões da caixa de ligação foi corrigida de acordo com o artigo[13] previamente citado, para ser empregada nesse novo ensaio, o qual será detalhado em um relatório separado, dedicado a essa outra avaliação.

Referências

- [1] Xuan-Yu Meng, Ming-Xing Song, Mihaly Mezei, and Meng Cui. Molecular docking: A powerful approach for structure-based drug discovery. *Current computer-aided drug design*, 7:146–57, 06 2011.

- [2] Douglas Kitchen, Hélène Decornez, John Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: Methods and applications. *Nature reviews. Drug discovery*, 3:935–49, 12 2004.
- [3] Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 02 2015.
- [4] Oleg Trott and Arthur Olson. Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31:455–61, 11 2009.
- [5] Leonardo Ferreira, Ricardo Santos, Glaucius Oliva, and Adriano Andricopulo. Molecular docking and structure-based drug design strategies. *Molecules*, 20:13384–13421, 07 2015.
- [6] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The pdbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, 47:2977–80, 07 2004.
- [7] Caio Rocha. <https://github.com/caiocrocha/pdbbind-coreset-2013/blob/master/RESULTS.ipynb>, 2020. Accessed: 2020-10-09.
- [8] Caio Rocha. <https://github.com/caiocrocha/pdbbind-coreset-2013>, 2020. Accessed: 2020-10-09.
- [9] Garrett Morris and R. Huey. Autodock4 and autodocktools4: automated docking with selective receptor flexibility. *J. Comput. Chem.*, 48:443–453, 01 2009.
- [10] Oleg Trott. AutoDock Vina manual. <http://vina.scripps.edu/manual.html#exhaustiveness>, 2008. Accessed: 2020-07-09.
- [11] Zhe Wang, Huiyong Sun, Xiaojun Yao, Dan Li, Lei Xu, Youyong Li, Sheng Tian, and Tingjun Hou. Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power. *Phys. Chem. Chem. Phys.*, 18:12964–12975, 2016.
- [12] Schrödinger, LLC. The PyMOL molecular graphics system, version 2.2.0. PyMOL The PyMOL Molecular Graphics System, Version 2.2.0, Schrödinger, LLC., November 2015.
- [13] Wei Feinstein and Michal Brylinski. Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. *Journal of Cheminformatics*, 7, 12 2015.
- [14] David Koes, Matthew Baumgartner, and Carlos Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53, 02 2013.
- [15] Caio Rocha. https://github.com/caiocrocha/pdbbind-coreset-2013/blob/master/local_opt/Local%20optimization%20scores.ipynb, 2020. Accessed: 2020-16-09.
- [16] Caio Rocha. <https://github.com/caiocrocha/pdbbind-coreset-2013/blob/master/rescore/RESCORE.ipynb>, 2020. Accessed: 2020-16-09.