MINISTÉRIO DA SAÚDE
FUNDAÇÃO OSWALDO CRUZ
INSTITUTO OSWALDO CRUZ

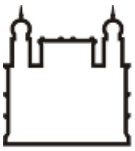
Doutorado em Programa de Pós-Graduação em Biologia Computacional e Sistemas

MELHORAMENTO DE *DOCKING-BASED VIRTUAL SCREENING* USANDO ABORDAGEM DE *DEEP LEARNING*.

JANAINA CRUZ PEREIRA

Rio de Janeiro

2017



Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ
Pós-Graduação em Biologia Computacional e Sistemas

JANAINA CRUZ PEREIRA

Melhoramento de *docking-based virtual screening* usando abordagem de *deep learning*.

Tese apresentada ao Instituto Oswaldo Cruz como parte dos requisitos para obtenção do título de Doutor em Biologia Computacional de Sistemas.

Orientador: Dr. Ernesto Raúl Caffarena

Rio de Janeiro

2017

Pereira, Janaina Cruz .

Melhoramento de docking-based virtual screening usando abordagem de deep learning. / Janaina Cruz Pereira. - Rio de janeiro, 2017.
168 f.; il.

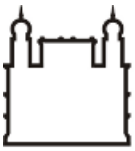
Tese (Doutorado) – Instituto Oswaldo Cruz, Pós-Graduação em Biologia Computacional e Sistemas, 2017.

Orientador: Ernesto Raúl Caffarena.

Bibliografia: f. 143-159

1. Aprendizado de máquina. 2. Planejamento de fármacos. 3. Redes neurais. 4. Doença de Chagas. 5. Docking molecular. I. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da Biblioteca de Manguinhos/ICICT com os dados fornecidos pelo(a) autor(a).



Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ
Pós-Graduação em Biologia Computacional e Sistemas

AUTOR: JANAINA CRUZ PEREIRA

MELHORAMENTO DE *DOCKING-BASED VIRTUAL SCREENING* USANDO ABORDAGEM DE *DEEP LEARNING*.

ORIENTADOR: Prof. Dr. ERNESTO RAÚL CAFFARENA

Aprovada em:

EXAMINADORES:

Prof. Dr. Leonardo Soares Bastos - **Presidente** (PROCC-Fiocruz)
Prof. Dr. Flávio Codeço Coelho (FGV)
Prof. Dr. Laurent Emmanuel Dardenne (LNCC)
Prof. Dr. Camila Silva de Magalhães - Suplente (UFRJ)
Prof. Dr. Fabrício Alves Barbosa da Silva - Suplente (IOC-Fiocruz)

Rio de Janeiro, 10 de abril de 2017

Anexar a cópia da Ata que será entregue pela SEAC já assinada.

Aos meus amores:
meus Pais, meu Esposo e meu Pluto.

AGRADECIMENTOS

Ao professor Dr. Ernesto R. Caffarena, por toda a orientação, incentivo, dedicação ao longo deste projeto e, acima de tudo, por ter acreditado que eu seria capaz de realizar o trabalho.

Ao meu esposo, Cícero, pelo amor, paciência, incentivo, debates, dicas e principalmente por acreditar no meu sonho de ser uma cientista e me ajudar a construí-lo.

Aos meus pais, Dalva e Roberto, que mesmo sem entender direito o que eu estava fazendo, se sentiam realizados com a minha felicidade. Em especial à minha mãe, por todos os dias ligar e perguntar se eu já havia terminado a tese.

Ao meu irmão, Betinho, por cuidar dos meus pais e dessa forma me deixar tranquila para dar continuidade aos meus estudos.

Ao Dr. Cícero Nogueira dos Santos pela ajuda constante durante o desenvolvimento desse trabalho e à IBM por possibilitar a parceria.

À Dra. Rafaela S. Ferreira pela supervisão durante o trabalho com a Cruzaína.

Ao trio parada dura: Gisele Rocha, Priscila Moura e Vanessa Silva. Às três pelo apoio psicológico. À Priscila por me colocar “pra cima” quando eu mais precisava e por, sempre que precisei, ceder sua casa para minha estadia, onde eu me sinto parte da família. À Gisele por todas as dicas e principalmente por conseguir os artigos que eu precisava. À Vanessa por todas as dicas de programas, tutoriais e consultoria no trabalho sobre a Cruzaína. Vocês moram no meu coração meninas.

À equipe do Grupo de Biofísica Computacional e Modelagem Molecular por toda ajuda durante o meu processo de doutorado, desde do simples ato de ligar o meu computador, quando precisei, até as divertidas conversas durante o almoço. Saibam que o mundo é movido por pequenos atos.

À Lucianna Santos por toda a consultoria com relação à Cruzaína/Dock6 e pela paciência em ler e responder e-mails enormes, mas muito divertidos.

Ao Rafael Ferreira por gerenciar os servidores e realizar a logística relacionada às minhas máquinas. Sem o seu trabalho seria impossível realizar o meu.

Ao Dr. Leonardo Soares Bastos por revisar a minha tese e prontamente aceitar o convite para ser o presidente da banca.

Aos demais membros e suplentes da banca, Dr. Laurent Emmanuel Dardenne, Dr. Flávio Codeço Coelho, Dra. Camila Silva de Magalhães e Dr. Fabrício Alves Barbosa da Silva pelas contribuições e por prontamente aceitarem o convite para in-

tegrarem o corpo da banca.

To my dear friends Makiko, Rieko, Arlene, Jutta, Noeli, Maria, Renata, Haruka, Inés, Kalsoom, Ania, Michiko, Minako, Jessie, Yenne, Débora, Alejandra, Hellen, Maria Luzia, Satoko e Luciana for all the support during this time.

À todos que corretamente pagaram seus impostos, o que tornou possível à Fundação Oswaldo Cruz conceder auxílio financeiro a esse projeto.

À Secretaria do Programa de Pós-Graduação em Biologia Computacional e Sistemas, em especial a Rose Pani, por auxiliar nos processos burocráticos inerentes ao doutorado.

À Fundação Oswaldo Cruz, através do Programa de Pós-Graduação em Biologia Computacional e Sistemas (corpo docente e discente), que proporcionou a minha formação.

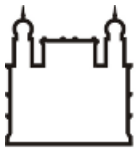
À todos aqueles que direta ou indiretamente contribuíram para a realização do trabalho.

Gratidão!

Sei de vossa cordialidade e conto com a vossa boa vontade
para com as incertezas e falhas do neófito.
Carlos Chagas Filho

Eu não tenho nenhuma escolha,
mas o dever de ser a primeira.
Mary Jackson
(Hidden Figures - Margot Lee Shetterly)

Seja a heroína da sua vida,
não a vítima.
Nora Ephron



Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

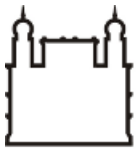
MELHORAMENTO DE *DOCKING-BASED VIRTUAL SCREENING* USANDO ABORDAGEM DE *DEEP LEARNING*.

RESUMO

TESE DE DOUTORADO EM BIOLOGIA COMPUTACIONAL E SISTEMAS

JANAINA CRUZ PEREIRA

Funções de pontuação são um dos grandes problemas na metodologia de *Docking-Based Virtual Screening* - DBVS, pois elas não são capazes de classificar de forma confiável ligantes docados. Nesse trabalho propomos um novo método baseado em *Deep Learning* para melhoramento de DBVS. Nossa abordagem usa a saída do *docking* para aprender como extrair *features* relevantes a partir de informações básicas como tipos de átomo e tipos de resíduos provenientes do complexo proteína-composto. Nossa abordagem introduz o conceito de *embeddings* para átomos e aminoácidos e implementa uma forma efetiva de criar representações de vetores distribuídos para complexos proteína-composto. Uma da maiores vantagens da abordagem proposta em detrimento aos métodos encontrados na literatura é a capacidade de aprender *features* com pouca ou nenhuma intervenção humana. Para verificarmos a eficácia da DeepVS, executamos experimentos de *docking* com o programa Autodock-vina1.2 e Dock 6.6 utilizando o banco de dados DUD com cargas corrigidas. Adicionalmente, reportamos resultados usando um subconjunto do banco de dados DUD-E. O desempenho da DeepVS é avaliado com o uso da abordagem de validação cruzada (*leave-one-out*) e empregando-se métricas bem estabelecidas como fator de enriquecimento e AUC. Usando a saída do programa Autodockvina1.2, a DeepVS registra uma AUC ROC de 0,81, que até onde sabemos é o melhor resultado de AUC já reportado para DBVS usando os 40 receptores do DUD. Para o subconjunto de 44 proteínas do DUD-E, os experimentos de validação cruzada resultaram em uma AUC média de 0,93, valor que é superior ao reportado por trabalhos recentes. Adicionalmente, aplicamos a DeepVS (treinada com o DUD) em um estudo de caso envolvendo a enzima Cruzaína. O cisteíno-protease Cruzaína é considerado como o principal cisteíno do protozoário *Trypanosoma cruzi* agente etiológico da doença de Chagas. Nesse estudo de caso abordamos todas as principais etapas de *virtual screening* baseado em estrutura envolvendo escolha da estrutura cristalográfica, estudo do sítio de ligação, estudos com controles positivos para verificação do método a ser aplicado, seleção de um conjunto de compostos a serem ranqueados em um banco de dados de ligantes, *virtual screening*, seleção de compostos potencialmente ativos e inspeção visual dos compostos selecionados. As estratégias utilizadas no estudo de caso tornaram possível a identificação de sete compostos candidatos a fármacos em um conjunto de dados de 90.769 compostos comercialmente adquiríveis.



Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

IMPROVING DOCKING-BASED VIRTUAL SCREENING WITH DEDEEP LEARNING

ABSTRACT

PHD THESIS IN COMPUTATIONAL BIOLOGY AND SYSTEMS

JANAINA CRUZ PEREIRA

Scoring functions are one of the biggest problems in Docking-Based Virtual Screening - DBVS approach because these functions are not able to reliably classify docked ligands. In this work, we propose a new Deep Learning based approach for improving DBVS. The proposed deep neural network, DeepVS, uses the output of a docking program and learns how to extract relevant features from basic data such as atom and residues types obtained from protein-ligand complexes. Our approach introduces the use of atom and amino acid embeddings and implements an effective way of creating distributed vector representations of protein-ligand complexes by modeling the compound as a set of atom contexts that is further processed by a convolutional layer. One of the main advantages of the proposed method is that it does not require feature engineering. We evaluate DeepVS on the Directory of Useful Decoys (DUD), using the output of two docking programs: Autodock Vina1.2 and Dock 6.6. In addition, we also report results using a subset of the DUD-E database. DeepVS performance is evaluated with the leave-one-out cross-validation approach and using the well-established metrics enrichment factor and AUC. Using the output of the Autodockvina1.2 program, DeepVS achieves an AUC ROC of 0.81, which to the best of our knowledge is the best AUC result reported so far to DBVS using the 40 receptors in DUD. For the subset of 44 DUD-E receptors used in this work, the cross-validation experiments resulted in an AUC of 0.93, which is also better than the AUC reported on recently published works. Finally, we applied DeepVS for a case study involving the enzyme Cruzain. The cysteine protease Cruzain is considered as the main cysteine protease of the protozoan parasite Trypanosoma cruzi, etiologic agent of Chagas disease. In this case study we perform all the main steps of structured based virtual screening involving, choice of crystallographic structure, study of the binding site, studies with positive controls to verify the method to be applied, selection of a set of compounds from a docking database, virtual screening, selection of potentially active compounds and visual inspection of selected compounds. The strategies used in the case study made it possible to identify seven drug candidate compounds in a dataset of 90,769 commercially available compounds.

PREFÁCIO

O trabalho apresentado nessa tese é fruto de dois projetos desenvolvidos pela autora ao longo do seu doutorado. Uma extensa investigação de estratégias para *virtual screening* baseado em estrutura levou à combinação de ambos os projetos em duas partes separadas na tese.

O primeiro projeto, trata-se de um novo método baseado em *deep learning* para melhoramento de *virtual screening* baseado em estrutura e foi desenvolvido em parceria com o pesquisador Dr. Cícero Nogueira dos Santos, da IBM Research. Nessa parceria, Dr. Cícero contribuiu com discussões sobre a concepção e com a implementação da rede neural apresentada nesse trabalho. Todo o projeto foi supervisionado integralmente pelo o Dr. Ernesto R. Caffarena.

O segundo projeto, trata-se de um estudo de caso para a enzima Cruzaína, agente etiológico da doença de Chagas, no qual todas as etapas que compreendem *virtual screening* baseado em estrutura foram empregadas, desde da escolha da estrutura até a seleção de ligantes candidatos à fármacos. Essa parte do trabalho foi desenvolvida em parceria com a pesquisadora da Universidade Federal de Minas Gerais (UFMG) Dra. Rafaela S. Ferreira, apoiado pelo Projeto Casadinho. O Projeto Casadinho promove a interação entre a UFMG e o Instituto Oswaldo Cruz. O projeto foi inteiramente supervisionado pelo Dr. Ernesto R. Caffarena.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AD4	Autodock4.2
ADV	Autodockvina1.1.2
AI	*Artificial Intelligence* (Inteligência Artificial)
Anvisa	Agência Nacional de Vigilância Sanitária
AUC	Área sob a curva ROC
CNN	*Convolutional Neural Networks* (Redes Neurais Convolucionais)
DBVS	*Docking-based Virtual Screening* (*Virtual screening* baseado em estrutura)
DDFA	Docking Data Features Analysis
DL	*Deep Learning*(Aprendizado profundo)
DNN	*Deep Neural Network* (Redes neurais profundas)
DUD	*Directory of Useful Decoys*
DUD-E	*Directory of Useful Decoys - Enhanced*
1D	Unidimensional
2D	Bidimensional
3D	Tridimensional
DTN	Doenças Tropicais Negligenciadas
EF	*Enrichment Factor* (Fator de enriquecimento)
FP	*Molecular Fingerprints*
GPUs	*Graphics Processing Unit* (Unidade de processamento gráfica)
HTS	*High Throughput Screening* (Triagem de alto rendimento)
HTVS	*High Throughput Virtual Screening* (Triagem virtual de alto rendimento)
LBVS	*Ligand-Based Virtual Screening* (Triagem virtual baseado no ligante)
ML	*Machine Learning* (Aprendizado de máquina)
MLP	*Multilayer Perceptron*
NCI	*National Cancer Institute* (Instituto Nacional de Câncer)
NN	*Nerual Networks* (Redes neurais)
PDB	*Protein Data Bank*
PLN	Processamento de Linguagem Natural
QSAR	*Quantitative Structure-Activity Relationships*
RELU	*Rectified Linear Unit*

RF *Random Forest* (Floresta aleatória)

RL RosettaLigand3.4

ROC *Receiver Operating Characteristic*

SBVS *Structure-Based Virtual Screening* (Triagem virtual baseados em estrutura)

SGD *Stochastic Gradient Descent* (Gradiente descendente estocástico)

SP *Standard Precision* (Padrão de precisão)

SVM *Support Vector Machines* (Máquinas de vetores de suporte)

VS *Virtual Screening* (Triagem virtual de compostos)

XP *Extra Precision* (Precisão extra)

1 PARTE I

1.1 Introdução

1.1.1 Caracterização do problema

O processo de descoberta de um novo fármaco, ou até mesmo o seu reposicionamento, é uma tarefa de alto custo financeiro que demanda um longo período de tempo [1]. A descoberta de um novo fármaco envolve vários estágios. Dentre eles, no seu estágio inicial, a seleção de compostos potencialmente ativos para um determinado alvo farmacológico em bibliotecas com milhares ou até milhões de compostos químicos [2].

Métodos experimentais foram desenvolvidos para auxiliar na seleção de compostos ativos, porém esses métodos são caracterizados por possuírem estratégias altamente automatizadas, que estão associadas a um oneroso custo financeiro e uma alta demanda de tempo. A soma de todas essas características torna esses métodos inacessíveis à comunidade acadêmica [2, 3]. Além disso, altas taxas de falsos negativos (compostos ativos que não são identificados durante a triagem bioquímica) e falsos positivos (compostos inativos que são identificados como ativos) ainda são observadas em métodos experimentais tais como HTS (*High Throughput Screening*) [3, 4].

Dessa forma, métodos computacionais são fortemente encorajados como uma alternativa de baixo custo para auxiliar no processo de desenvolvimento de novos fármacos [3,4]. Como por exemplos, podemos citar as estrategias de *Virtual Screening* (ou triagem virtual de compostos) que podem auxiliar na redução de custo, tempo e esforço humano no desenvolvimento de novos fármacos. Isso acontece por que, para cada composto selecionado são levados em consideração suas características estruturais e propriedades físicas, o que pode aumentar a probabilidade do composto selecionado ter algum tipo de atividade biológica para um alvo específico [5] (seção 1.1.2). A tabela 1.1 lista de forma resumida estratégias de sucesso utilizando *Virtual Screening* [6].

As metodologias de *virtual screening* são geralmente divididas em duas abordagens: (1) baseadas no ligante (seção 1.1.3), e (2) baseadas na estrutura (seção 1.1.4) [4, 12]. Para a identificação de novos compostos com potencial terapêutico, métodos baseados em estrutura geralmente têm um melhor desempenho quando com-

Tabela 1.1: Exemplos de trabalhos recentes que obtiveram sucesso utilizando estratégias de *Virtual Screening*. Fonte: (Villoutreix *et al.*, 2009).

Alvo e Mecanismo	Função da Doença	Estrutura	Tamanho inicial do Conjunto de Compostos	Ferramentas Computacionais	Referências Bibliográficas
Receptor GPCR/ Neuroquinina-1	media numerosos processos fisiológicos	modelo	827.000 moléculas	Selector, Unity e FlexX-Pharm	[7]
Kinase CK2 (sítio catalítico)	câncer	raio-X, 1JWH	2.000 compostos naturais	MOE-dock, Glide, Surflex e Gold	[8]
Acetil-CoA carboxilase (sítio catalítico)	tuberculose	raio-X, 2A7S	> 4 milhões de moléculas	DOCK, ICM e ChemDB algoritmo	[9]
Proteína G (proteína - proteína)	media numerosos processos fisiológicos	raio-X, 1XHM	1.990 moléculas	FlexX	[10]
Tirosina fosfatase (sítio catalítico)	câncer e diabetes	raio-X, 1DG9	875.866 moléculas	MPA, LINGOsim e Autodock	[11]

parados a métodos baseados no ligante [2,13]. Apesar disso, uma porcentagem baixa de compostos selecionados por *virtual screening* baseado em estrutura seguem até a etapa de ensaios clínicos (ou estágios finais) do desenvolvimento de novos fármacos [1].

Sistemas baseados em aprendizado de máquina (*Machine Learning* - ML) vêm sendo utilizados com sucesso para melhorar o resultado de *virtual screening* baseado em estrutura especialmente na subárea *virtual screening* baseado em *docking* ou *Docking-based Virtual Screening* - DBVS (seção 1.1.4.1), tanto no que diz respeito na melhora da performance das funções de pontuação quanto na construção de classificadores de afinidade de ligação [1,4] (seção 1.1.5). As principais estratégias usadas em *virtual screening* são redes neurais (*Neural Networks* - NN) [14], máquinas de vetores de suporte (*Support Vector Machines - SVM*) [15] e floresta aleatória (Random Forest - RF) [16]. Um dos grandes diferenciais das estratégias que empregam aprendizado de máquina é a capacidade de lidar com a dependência não linear entre diversas interações envolvidas entre o ligante e o receptor [4].

Estratégias tradicionais de aprendizado de máquina dependem diretamente a partir da forma de como o dado é apresentado. Por exemplo, em abordagens de *virtual screening*, cientistas normalmente analisam a saída do *docking* e identificam manualmente *features* que poderiam ser utilizadas para seleção de uma pose ("conformação de baixa energia") ou distinguir entre ligantes ativos e *decoys*. Um dos grandes problemas associados a esse processo é que ele pode ser efetivo até um certo ponto. O processo de identificação manual de *features* é trabalhoso, complexo e não pode ser aplicado em larga escala, resultando em uma perda de informação relevante e consequentemente conduzindo a um conjunto de *features* incapazes de explicar a atual complexidade do problema. [1,4,17,18].

Por um outro lado, trabalhos recentes em *Deep Learning* (DL) (seção 1.1.9), uma família de abordagens em aprendizado de máquina que minimiza o uso de *features* projetadas manualmente, vêm demonstrando um grande sucesso em diferentes tare-

fas advindas de múltiplas áreas [17–19]. Abordagens de DL normalmente aprendem *features* diretamente do dado bruto com um minimo ou nenhuma intervenção humana, o que resulta em um sistema que pode adaptar-se facilmente a novos conjuntos de dados (seção 1.1.7).

Nesse trabalho, nós propormos uma abordagem baseada em redes convolucionais para melhorar *virtual screening* baseado em estrutura. A abordagem usa resultados de simulações de *docking* como entrada para uma rede neural profunda (*Deep Neural Network*), nomeada como DeepVS, que aprende automaticamente a extrair *features* relevantes a partir de informações básicas como tipos de átomos, cargas atômicas parciais, distância entre átomos presentes em um complexo proteína-ligante (seção 1.3.2.1). DeepVS aprende *features* abstratas que são adequadas para discriminar ligantes ativos de *decoys* (seção 1.3).

Nós avaliamos a DeepVS usando o bando de dados DUD (*Directory of Useful Decoys*) (seção 1.3.3.1), que contém 40 receptores diferentes. Em nossos experimentos, usamos a saída de dois programas de *docking*: AutodockVina1.1.2 e Dock6.6 (seção 1.3.3.3). Adicionalmente, reportamos resultados usando um subconjunto do banco de dados DUD-E (*Directory of Useful Decoys - Enhanced*). Para ambos os bancos de dados, a DeepVS possui uma performance melhor quando comparada com os programas de *docking* molecular. Além disso, se compararmos os nossos resultados com sistemas previamente reportados na literatura, a DeepVS registra o estado-da-arte com o melhor valor de AUC reportado.

1.1.2 Métodos de *Virtual Screening*

A abordagem *virtual screening* funciona como um filtro (ou pré-filtro) que consiste na seleção de potenciais compostos ativos para um determinado alvo farmacológico [3] tendo como base algum critério predefinido (Figura 1.1).

Em outras palavras, *virtual screening* é uma metodologia que ranqueia um conjunto de compostos levando em consideração muitas vezes um *score*. Quando essa metodologia é usada de maneira bem sucedida gera uma lista de compostos ranqueados, de modo que no topo da lista se concentram os compostos com maior probabilidade de possuir atividade para um determinado receptor ou conjunto de receptores. Os ligantes ranqueados pelo o *virtual screening* seguem então para as etapas experimentais [20].

Tradicionalmente técnicas de *virtual screening* podem ser divididas em dois tipos: baseado no ligante, usado quando não se dispõe da estrutura tridimensional do receptor (seção 1.1.3); e baseado na estrutura ou no receptor, que é utilizada quando a estrutura tridimensional do receptor já foi resolvida de forma experimental ou computacionalmente modelada (seção 1.1.4) [12, 13].

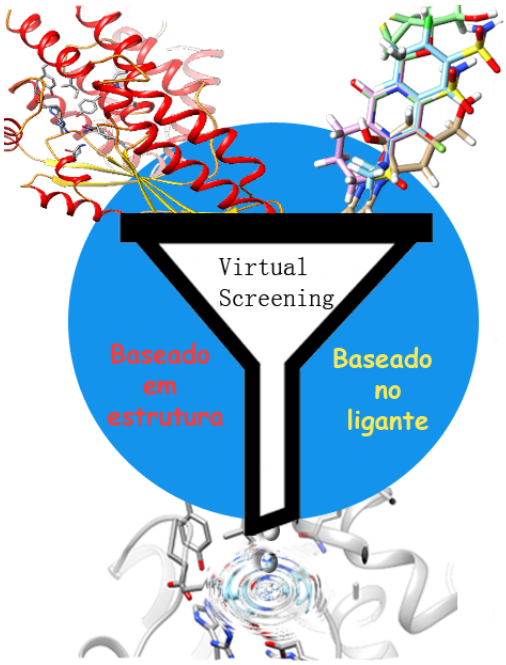


Figura 1.1: **Analogia à metodologia de *Virtual Screening***. *Virtual Screening* funciona basicamente como um grande filtro que tem por o objetivo analisar computacionalmente uma grande quantidade de compostos e selecionar, de acordo com algum critério predefinido, compostos mais ativos para um determinado alvo farmacológico. O passo inicial para o *Virtual Screening* é a obtenção de um conjunto de ligantes a partir da busca em banco de dados específicos; O segundo passo é optar por uma das duas abordagens de *Virtual Screening*.

1.1.3 *Virtual Screening* Baseado no Ligante

A estratégia de *virtual screening* baseado no ligante (*Ligand-Based Virtual Screening*-LBVS) consiste na análise de similaridade de propriedades estruturais e físico-químicas de ligantes conhecidamente ativos para predizer atividade de ligantes com características semelhantes [13, 21, 22], baseado no conceito de similaridade molecular, no qual moléculas com estruturas similares podem vir a exercer atividades biológicas similares [23].

As técnicas usadas em LBVS levam em consideração o uso descritores e têm como objetivo capturar as características e propriedades de uma determinada molécula através da sua estrutura molecular. Esses descritores podem ter variados níveis de complexidade mas possuem em comum o atributo dimensionalidade, que é a representação molecular a partir da qual os descritores são computados [24]. Dessa forma, métodos de LBVS podem ser classificados de acordo com a dimensão na qual a molécula está representada, como: unidimensional (1D), bidimensional (2D) e tridimensional (3D) [24, 25].

Os métodos 1D estão usualmente relacionados a descritores de representações unidimensional e são geralmente empregados para selecionar banco de dados específicos usando filtros moleculares ou fisioquímicos, a exemplo do conjunto de dados *drug-like* e *lead-like*. Os exemplos mais comuns de descritores unidimensionais

são os que levam em consideração propriedades fisioquímicas e moleculares globais tais como peso molecular, LogP, candidatos a ligações de hidrogênio, dentre outros [24, 26, 27].

Métodos que computam similaridade entre descritores derivados de estruturas bidimensionais são conhecidos como métodos 2D [26]. Um dos métodos 2D mais comuns é o *Molecular Fingerprints* (FP), que consiste em gerar um vetor de bits que contém a informação da presença ou ausência de *features* moleculares adivindas de um composto [22] (Figura 1.2). Para computar a similaridades entre estruturas moleculares os vetores de *bits* são comparados usando coeficientes de similaridade como o coeficiente de Tanimoto [22, 26, 28, 29]. O coeficiente de Tanimoto é determinado pelo número de características químicas que são comuns em ambas as moléculas, por exemplo o número de *bits* positivos comuns dado duas sequências em comparação com o número de características químicas que estão presentes nessas moléculas, número total de *bits* positivos [26, 28].

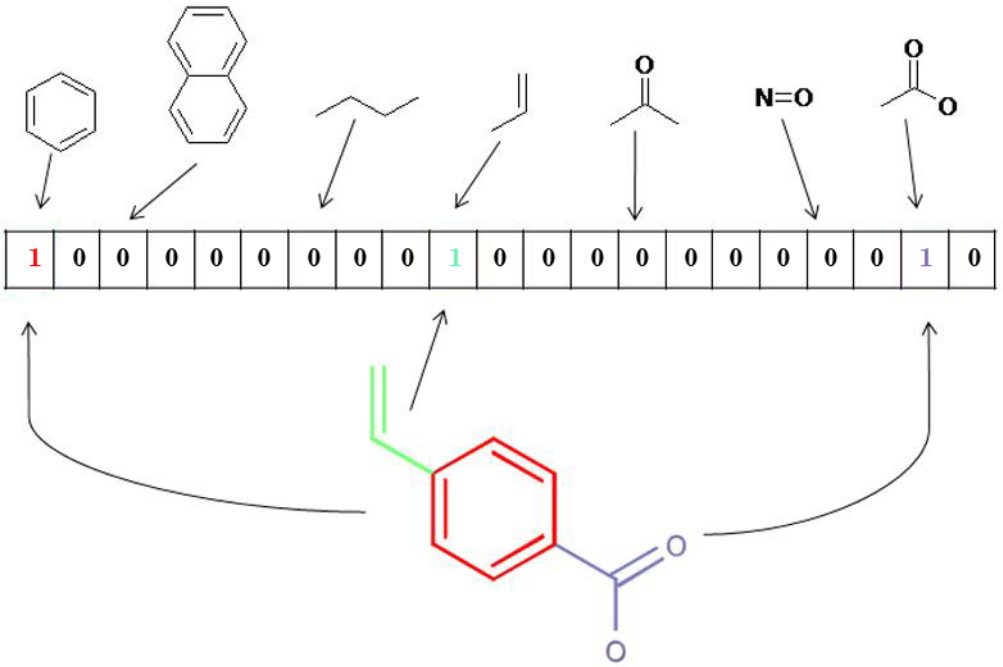


Figura 1.2: ***Molecular Fingerprints* são utilizadas para representar a estrutura de cada molécula do fármaco**. A ausência e presença das substruturas de um composto é representada em uma sequência de *bits* similar à metodologia de *Molecular Fingerprints*. Fonte: Cao *et al.* (2013) adaptado por Pereira, J.C.

Métodos 3D são considerados com melhor performance se comparados a métodos 2D. Isso ocorre por que métodos 3D possuem a capacidade de realizar *scaffold hopping* [26], como por exemplo a capacidade de identificar moléculas que realizam as mesmas interações com o receptor porém possuem estruturas químicas diferentes [21, 26]. O método 3D mais popular é o baseado em farmacóforos [30]. Farmacóforos foram primeiramente mencionados em 1909 [30] como a descrição de uma molécula biologicamente ativa que envolve (phoros) as características necessárias

responsáveis pela a atividade daquela molécula ou composto (pharmacon) [30, 31]. Em *virtual screening*, farmacóforos são utilizados para pesquisas de moléculas em banco de dados que possuam o melhor arranjo geométrico o que potencializaria a chance dessa molécula ser uma molécula ativa para um determinado alvo farmacológico [26, 31]. Exemplos de programas que realizam *virtual screening* baseado em farmacóforos são ARDAs [32], primeiro programa a usar *virtual screening* baseado em farmacóforos para identificação de agentes contra o câncer de próstata e Silicos-it [33] uma alternativa gratuita e eficiente para *virtual screening* baseado em farmacóforos.

Em ordem de aprender a identificar ou a prever descritores dentro do conceito de similaridade molecular, abordagens de aprendizado de máquina vêm sendo extensivamente usadas nos últimos anos [22], como: ID3 [34], máquinas de vetores de suporte (*Support Vector Machines*-SVM) [35], máquinas de aprendizado linear (*Linear Learning Machine*) [36] e classificador Bayesiano (*Bayesian classifier*) [37]. Em anos recentes *deep learning*, têm despertado o interesse da comunidade científica e de grandes industrias farmacêuticas, a exemplo da empresa Merck [38], com o objetivo de encontrar o melhor conjunto de descritores para um determinada tarefa, como por exemplo reposicionamento de fármacos (seção 1.1.9).

1.1.4 *Virtual Screening* Baseado na Estrutura

Métodos baseados em estrutura (*Structure-Based Virtual Screening* - SBVS) ou métodos baseados em receptor (*Receptor-Based Virtual Screening* - RBVS) são utilizados quando a estrutura tridimensional do receptor foi resolvida experimentalmente [39] ou modelada computacionalmente [40, 41]. Uma vez a estrutura 3D do receptor encontra-se disponível em bancos de dados de estruturas, como por exemplo o PDB (*Protein Data Bank*) [42], é então escolhida para as demais etapas de *virtual screening* aquela que possui a melhor resolução [31].

O conhecimento prévio do sítio de ligação é fundamental para metodologias de SBVS [31]. Estudos relacionados ao sítio de ligação no receptor envolvem tanto suas características físico-químicas quanto suas características estruturas [43], tais como estado de protonação dos resíduos envolvidos em interações ligante-receptor; estado de protonação e a forma enantiomérica biologicamente ativa do ligante [44].

Metodologias que envolvem estratégias de SBVS são mais produtivas e informativas se comparadas com metodologias que envolvem estratégias de *virtual screening* baseado no ligante [31]. Isto está relacionado ao fato que estratégias baseadas em SBVS exploram o reconhecimento molecular entre um ligante e um receptor alvo para selecionar moléculas que se ligam fortemente aos sítios ativos contidos em alvos biologicamente relevantes. Dessa forma, SBVS se caracteriza como uma metodologia aberta que permite a identificação de ligantes estruturalmente novos que podem fazer interações semelhantes às dos ligantes conhecidos ou podem ter diferentes interações com outros resíduos no sítio de ligação [45]. Este último procedimento não é

possível em estratégias que envolvem *virtual screening baseado* no ligante, pois esse tipo de metodologia é estruturada no conceito de similaridade molecular [23].

As estrategias mais utilizadas em SBVS são: (1) estudos de farmacóforos para o sítio de ligação ou *c-pharmacophore*, consiste em desenvolver hipóteses relacionadas ao sítio de ligação do receptor usando farmacóforos advindos do complexo proteína-composto [46]. O princípio básico de farmacóforos parte da premissa que grupos químicos relacionados espacialmente e geometricamente podem realizar interações semelhantes em um dado receptor e consequentemente podem gerar atividades biológicas semelhantes [47]. Normalmente, são utilizadas informações provenientes de funções de pontuação ou de pontuações de contato para categorizar interações, como por exemplo energia de ligação, entre o receptor e o composto de forma a criar *features* que representem essas interações as quais constituem o modelo do farmacóforo [46, 47]. (2) baseado em *docking* molecular (*Docking-based Virtual Screening* - DBVS). Essa abordagem de SBVS é utilizada para predizer o modo de ligação e a afinidade de ligação de um conjunto de compostos dado um determinado receptor [44] (ver seção 1.1.4.1).

Abordagens de SBVS que utilizam farmacóforos são computacionalmente mais eficientes se comparadas com abordagens de SBVS baseado em *docking* molecular. Isso porque, metodologias de *docking* molecular necessitam realizar inúmeras avaliações de energia que em muitos casos envolvem metodologias computacionalmente caras como campos de força [47]. Porém, abordagens de *docking* são mais efetivas se comparadas, a métodos baseado em farmacóforos. Ou seja, abordagens de *doking* conseguem distinguir de forma mais precisa ligantes e *decoys* dado um conjunto de compostos e um receptor específico [4, 13, 44].

1.1.4.1 *Virtual Screening* Baseado em *Docking* (Docking-based Virtual Screening - DBVS)

Em estratégias de *virtual screening* baseado em *docking* (*Docking-based Virtual Screening* - DBVS), a seleção de ligantes mais ativos em um banco de compostos é feita mediante o *docking* de cada composto contra o receptor alvo utilizando para isso programas de *docking* molecular [1, 12].

Uma abordagem bem sucedida de DBVS deve envolver as seguintes etapas (Figura 1.3): (1) seleção de um conjunto de compostos em banco de dados de pequenas moléculas e preparação dos ligantes selecionados; (2) estudos relacionados ao sítio de ligação do receptor e preparação do receptor para receber a etapa de *docking*; (3) *docking* molecular inclui investigação e predição da orientação conformacional do composto no sítio de ligação do receptor e predição de afinidade ligação (energia livre de ligação) em um complexo proteína-composto utilizando funções de pontuação; (4) analises pós-*docking* envolvem a seleção de um pequeno número de compostos que serão utilizados em ensaios experimentais, um exemplo de análise pós-*docking* é a

inspeção visual por meio de mapas de interação [1, 4, 12, 15, 43, 45].

A obtenção de um conjunto de ligantes a ser ranqueado é o primeiro passo para a metodologia de DBVS [1]. Existem diversos tipos de bancos de dados de pequenas moléculas disponíveis para *virtual screening*, como por exemplo: PubChem [48], ChEMBL [49], NCI Set [50], ChemSpider [51], MDDR [52] e o mais popular ZINC [53]. Esses bancos de dados podem conter milhares a milhões compostos, podem ser de acesso público ou de uso comercial, para fins específicos ou de amplo uso.

O conteúdo e a qualidade do banco de dados utilizado para seleção do conjunto de composto pode ser crucial para o sucesso das demais etapas de DBVS [4]. Para otimizar a obtenção do conjunto de ligantes para DBVS são utilizados desde filtros baseados em características físico-químicas do composto [27] a estratégias de *virtual screening* baseado no ligante [25] (seção 1.1.5). A etapa seguinte a seleção do conjunto de compostos é a etapa de preparação dos compostos que engloba a determinação do estado de protonação de cada composto, identificação da flexibilidade das ligações químicas dos compostos, possíveis formas enantioméricas e estado tautomérico dos compostos [43].

A etapa de estudo do sítio ativo e preparação do receptor se inicia com a obtenção da estrutura tridimensional da proteína. Estruturas 3D de proteínas podem estar disponíveis em bancos de estruturas de proteínas resolvidas experimentalmente utilizando dentre outras técnicas a difração de raio-X e espectroscopia de ressonância magnética nuclear, como por exemplo o *Protein Data Bank* (PDB) [42] ou em bancos de estruturas de proteínas modeladas computacionalmente utilizando dentre outras técnicas a predição de estrutura baseada em modelagem comparativa, como por exemplo o ModBase [54].

O segundo passo é a identificação do sítio de ligação na proteína alvo. Muitas vezes o sítio de ligação da proteína não é conhecido. Desse modo, estratégias para predição do sítio alvo foram desenvolvidas levando em consideração três informações distintas para inferir a localização de sítios na superfície da proteína: estrutura da proteína; informação evolutiva (alinhamento de sequências); informação advinda do ligante e do substrato [55].

Após a identificação do sítio alvo na proteína. O passo seguinte é realizar estudos relacionados as propriedades físico-químicas do receptor, estudos como esse contribuem para uma correta preparação da proteína. A preparação da proteína constitui uma etapa de grande importância em estratégias que utilizam a metodologia de DBVS [43]. Erros associados ao sítio de ligação como ausência de resíduos; sobreposição de átomos; apresentação de duas ou mais conformações para um resíduo ou conjunto de resíduos, como por exemplo formas tautoméricas da histidina; ligações inexistentes entre resíduos; um estado de protonação errado para resíduos como cisteína, glutamato, aspartato e histidina podem resultar em predições de conformações de compostos incorretas durante a etapa de *docking* [20, 56–58].

A preparação da proteína e seu sítio alvo está diretamente ligada com o tipo

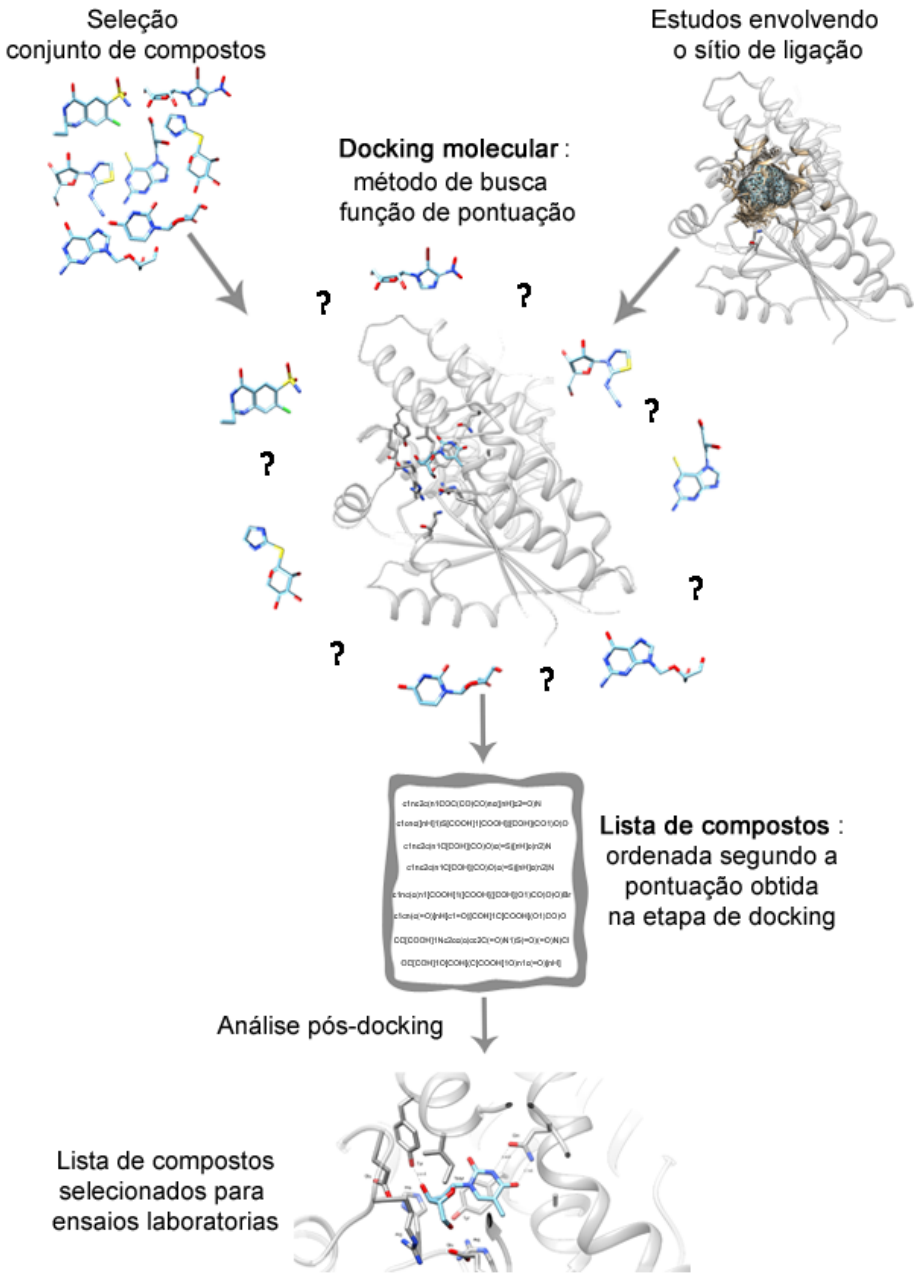


Figura 1.3: **Desenho esquemático dos processos de DBVS dado um alvo específico**. As etapas de DBVS compreendem: seleção de um conjunto de compostos em bancos de dados de pequenas moléculas e subsequentemente preparação desses compostos; estudos envolvendo o sítio ativo do receptor alvo, como estado de protanação dos resíduos e a correção de conformações incorretas que envolvem a etapa de preparação da proteína; na etapa de *docking* cada composto do conjunto selecionado é docado no sítio de ligação do receptor através de um programa de *docking* molecular, no qual modelos computacionais de interação testam as possibilidades de encaixe de modo a atingir o melhor estado de complementaridade. Funções de pontuação são utilizadas para avaliar a adequação entre o composto acoplado e o receptor; o resultado da etapa de *docking* é uma lista de compostos ordenados segundo sua pontuação; análise pós-*docking* envolvem desde inspeção manual dos compostos utilizando mapas de interação ao uso de programas ou funções de rescoring.

de programa de *docking* a ser utilizado [43]. Normalmente a preparação da proteína envolve adição de átomos de hidrogênio, checagem de possíveis sobreposições entre átomos, definição do correto estado de protonação e ionização dos resíduos pertencentes ao sítio de ligação, correção de tautômeros no sítio de ligação, seleção de águas conservadas e adicionalmente o uso de programas como o Whatcheck [59] ou ProCheck [60] para checar erros estruturais contidos na forma cristalográfica do receptor, tais como a posição de oxigênios e a orientação de anéis de histidina [4, 43, 58].

A etapa de *docking* molecular utiliza algoritmos para predizer a conformação e orientação (pose) do conjunto de compostos no sítio ativo do receptor. [12, 45]. De uma forma geral, a metodologia de *docking* empregada em DBVS pode ser dividida em dois passos: no primeiro, modelos computacionais de interação testam as possibilidades de encaixes ligante-receptor; no segundo, é feito o teste de adequação entre o composto acoplado e o alvo usando um ou mais expressões matemáticas (que podem ser chamadas de "função de pontuação"). Adicionalmente, outras funções de pontuação mais sofisticadas e com um gasto computacional maior podem também ser utilizadas para obter uma maior acurácia na predição do modo de ligação e na predição de afinidade entre o ligante e o receptor [4, 6, 12, 61].

Os métodos de busca podem ser divididos em três categorias básicas: métodos sistemáticos tais como construção incremental, busca conformacional e banco de dados; métodos estocásticos, como Monte Carlo, Algoritmos Evolucionistas e *Simulated Annealing*; e métodos determinísticos, como dinâmica molecular e métodos de minimização de energia [12].

Em uma busca sistemática da conformação e orientação do composto todos os graus de liberdade da molécula podem ser explorados, utilizando o conceito de "explosão combinatória". Nesse método, um conjunto de valores é estabelecido para cada grau de liberdade do composto de forma a efetuar uma exploração combinatória de todos os graus de liberdade da molécula durante a busca sistemática [12].

A busca estocástica utiliza alterações aleatórias nos graus de liberdade de um composto ou em uma população de compostos. Nesse caso, um composto dado como entrada para o processo de *docking* é avaliado com base em uma função de probabilidades pré-definida [12]

Em métodos determinísticos a abordagem de dinâmica molecular é o método mais popular, pois leva em consideração além dos graus de liberdade do composto a influência do solvente durante a busca [12, 62]. Uma das desvantagens do método está relacionado ao gasto computacional e à incapacidade de atravessar barreiras de alta energia dentro de períodos de simulação viáveis [62].

Como métodos de dinâmica molecular muitas vezes não conseguem atravessar barreiras de alta energia, esses métodos ficam presos em mínimos locais da superfície de energia do sistema [12, 62]. Algumas alternativas podem ser utilizadas para tentar superar o problema dos mínimos locais. Um exemplo disso seria, aumentar a temperatura da simulação, simular diferentes partes do sistema proteína-composto

a diferentes temperaturas, suavizar a superfície de energia potencial, o que permite a exploração completa da superfície de energia modificada e iniciar os cálculos de dinâmica molecular com diferentes conformações do composto [12, 62].

Métodos que envolvem minimização de energia, diferentemente dos métodos de dinâmica molecular, não são utilizados ou raramente são utilizados de forma individual como um método de busca porque alcançam apenas os mínimos locais de energia. Normalmente, métodos de minimização de energia são usados em conjunto com outros métodos de busca, como por exemplo o método de busca estocástica Monte Carlo [12, 63].

Na metodologia de *docking* molecular as funções de pontuação devem ser capazes de classificar dentro de um grande conjunto de pequenas moléculas conhecidas os compostos que possuam maior afinidade de ligação para um alvo específico durante as etapas de DBVS. Ou seja, devem distinguir dentro de um mesmo conjunto de dados ligantes ativos de *decoys*, que são moléculas que possuem características físicas semelhantes aos ligantes ativos, mas com topologia diferente o que resulta em uma possível inatividade da molécula, essa inatividade pode ser comprovada experimentalmente. [5].

As funções podem ser classificadas, de uma forma geral, em três tipos básicos de acordo com o modo que elas são derivadas: (1) baseado em campo de força, que são funções desenvolvidas baseadas em interações físicas em nível atômico incluindo interações de *van der Waals* (vdW), interações eletrostáticas e termos associados à torção das ligações químicas [5]. Esse método é limitado pela sua amostragem, pela precisão do campo de força usado e pela representação ou parametrização [1, 5, 12, 64]; (2) empírica: são funções que estimam a afinidade de ligação de um complexo proteína-composto baseadas em um conjunto ponderado de termos de energia. As principais dificuldades de aplicação estão relacionadas ao conjunto de treino, a dificuldade de controlar a dupla contagem em funções com muitos termos energéticos e a parametrização [1, 5, 16]; (3) baseados em conhecimento: funções baseadas em analises estatísticas entre os pares de átomos dos complexos proteína-composto resolvidos experimentalmente. A principal desvantagem desse tipo de função é que seu resultado está ligando diretamente à qualidade do seu conjunto de treino, podendo estar mal representado, o que torna o uso dessas funções restrito [4, 5, 16].

Diferentes tipos de funções de pontuação possuem em comum o mesmo problema que está relacionado à capacidade de representação ou parametrização de ligações [12, 15, 20, 65]. As funções de pontuação são conhecidas por serem pouco eficientes na predição de afinidade de ligação, o que constitui um gargalo na metodologia de *docking* e consequentemente para DBVS [14, 66]. Elas são incapazes de identificar de forma confiável conformações de ligantes ativos comparados a conformações de *decoys*. Assim, é mais fácil recuperar o modo de ligação mais provável durante a busca conformacional do que atribuir uma pontuação de baixa energia para uma pose correta [15]. Isso pode estar relacionado ao fato que funções de pontuação,

quando criadas, foram otimizadas para realizar comparações e cálculos de forma rápida e viável para análise de um grande número de compostos o que comprometeu de certa forma, a sua acurácia [65, 66]. Geralmente funções de pontuação atribuem um conjunto comum de pesos aos termos individuais de energia que contribuem para a pontuação global de energia. Além disso, as funções de pontuação que geram falsos negativos levam em consideração apenas interações individuais, de forma a predizer a afinidade de ligação proteína-composto a partir de uma combinação linear de termos de energia individuais, excluindo os efeitos cooperativos dessas ligações [5, 15].

Funções de pontuação caracterizam atualmente o problema central na metodologia de *docking*. Elas não são capazes de estimarem por si só a afinidade de ligação e, portanto classificar compostos docados. Dessa forma, analises pós-*docking* são necessárias, como por exemplo a inspeção visual dos compostos classificados como possíveis ligantes ativos, ou seja, aqueles que receberam maior valor de pontuação durante a etapa de *docking* (Figura 1.4). Esse problema se torna um desafio para humanos quando é preciso inspecionar manualmente milhares de poses de ligantes.

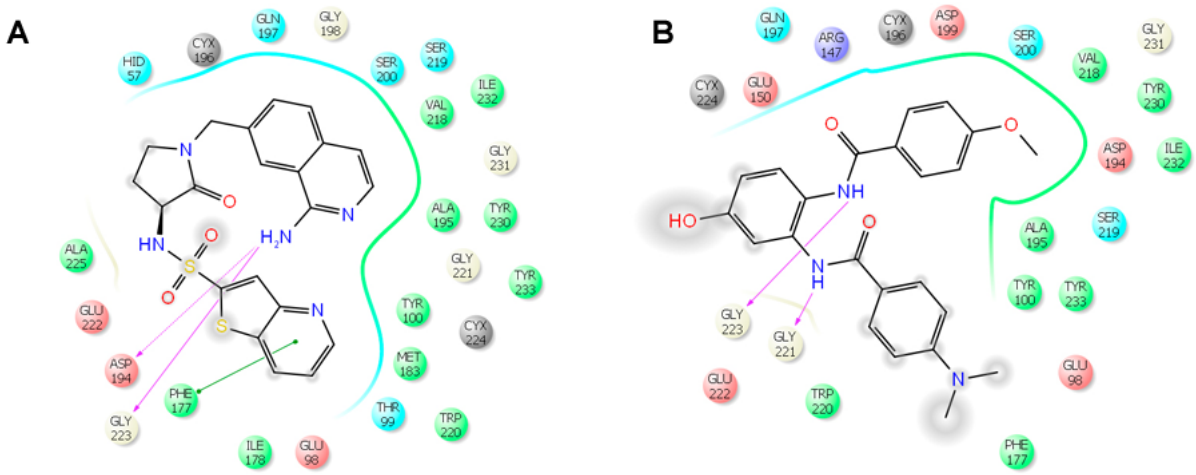


Figura 1.4: **Inspeção visual dos compostos.** Esquema de mapas de interação utilizados para inspeção manual de um composto ranqueado para a proteína Fxa (ID_PDB: 1F0R). Em **A** é retratado o mapa de interação para o composto nativo da proteína. Em **B** é retratado o mapa de interação para um composto ranqueado utilizando a metodologia de *docking*. Os cientistas utilizam esse mapas para comparar os tipos de ligações realizadas pelo o ligante nativo e o composto ranqueado de forma a descartar ou manter o composto ranqueado nas etapas de análise laboratorial. Para desenvolvimento das figuras foi utilizada a versão gratuita do *software* Maestro Schrodinger, disponível em $https://www.schrodinger.com/freemaestro/$.

Atualmente, diversos grupos de pesquisa estão trabalhando para amenizar essa problemática, sendo que os melhores resultados são obtidos usando abordagens baseadas em técnicas de aprendizado de máquina (Machine Learning - ML) [66, 67]. Aprendizado de máquina é uma área multidisciplinar que combina ideias neurociência; biologia; estatística; matemática e física para tornar o computador capaz de aprender [68]. Nas duas últimas décadas, algoritmos de aprendizado de máquina proporcionaram melhoras significativas em diversas áreas, resolveram problemas declarados insolúveis ou com respostas pouco reveladoras, como: (1) problemas relaci-

onados à mineração de dados (*data mining*), que consiste na análise automática de uma grande quantidade de dados na busca de regularidades implícitas que possam ser úteis; (2) problemas pouco ou ainda não entendidos por humanos como reconhecimento automático de face; (3) problemas dinâmicos cuja solução necessita adaptar-se constantemente a mudanças, como um sistema que proporcione mobilidade urbana eficiente [18, 68, 69].

Um levantamento feito por Arciniega & Lange (2014) [2], comparando 32 estratégias para *virtual screening*, mostrou que seis das dez melhores estratégias são baseadas em Redes Neurais (*Neural Network* - NN). O conjunto de dados do *Directory of Useful Decoys* (DUD) foi usado como o conjunto de dados de referência na pesquisa de Arciniega & Lange (2014) para comparação das demais estratégias [2]. O método NNScore de Durrante & McCammon (2010) [14], que usa redes neurais combinadas ao programa de *docking* AutodockVina [70] é o segundo melhor resultado para *virtual screening*, e apresenta o melhor resultado usando programas de livre acesso, ficando à frente de programas comerciais como o Glide (utilizando) [71] [2].

1.1.5 Métodos para Melhoramento de *Virtual Screening* usando Redes Neurais Tradicionais

Os métodos descritos nessa seção possuem em comum o uso de redes do tipo *multilayer perceptron*, que é um tipo de rede *feed-forward* (alimentação direta) (item 1.1.6). Designamos redes neurais tradicionais ou rasas como redes neurais que possuem topologia com somente uma camada escondida e *features* criadas utilizando técnicas da engenharia de *features*(Figura 1.5).

Redes Neurais (Neural Networks - NN) tradicionais são amplamente usadas para melhorar o resultado do *virtual screening* em cada uma de suas etapas, como:

1. *seleção guiada de compostos*, a seleção de compostos em bancos e/ou repositório de pequenas moléculas é a primeira etapa na metodologia de *virtual screening.* Alguns trabalhos vem sendo desenvolvidos para aperfeiçoar essa etapa, como a pesquisa desenvolvida por Sadowski & Kubinyi (1998) [72], onde os autores construíram uma rede que classifica compostos drogáveis (*drug-like*) de compostos não drogáveis (*nondrug-like*) baseado na semelhança de suas características, demonstrando que composto selecionados por seleção guiada usando critérios de semelhança pode aumentar significativamente a porção de compostos biologicamente ativos se comparado com uso de uma seleção aleatória.

2. *funções de pontuação*, nessa etapa estão concentrados os maiores esforços para melhoramento do resultado de *virtual screening* e são os métodos usando redes neurais que possuem os melhores resultados. A exemplo os trabalhos de: (i) Durrant & McCammon (2010) [14] com o NNScore1 que possui segundo melhor resultado para Virtual Screening segundo levantamento Arciniega & Lange

(2014) [2]; (ii) Durrant & McCammon (2011) [61] a com o NNScore2, ambas metodologias NNScore1 e NNScore2 possuem os melhores resultados de rescoring em virtual screening usando um mesmo conjunto de dados o DUD; e recentemente (iii) o trabalho de Ashtawy & Mahapatra (2015) [73] BgN-Score e BsN-Score para um diferente conjunto de dados o PDBbind. O fato da maior quantidade trabalhos usando redes neurais estarem concentrados na etapa de funções de pontuação pode estar relacionado à dificuldade que as funções de pontuação disponíveis para uso nas metodologias de *docking* possuem para distinguir ligantes ativos de *decoys* [5, 15, 65, 66].

3. *reclassificação*, é a etapa de reclassificação do resultado final da classificação de ligantes obtida mediante o emprego da metodologia de *virtual screening*. Difere dos métodos usados para melhorar funções de pontuação (*rescoring*) no que diz respeito ao tipo de treinamento da rede. Redes para problemas de classificação são treinadas usando *features* que usam características do *docking* associadas a moléculas ativas, o que distingue uma reclassificação (*reranking*), enquanto os métodos de *rescoring* treinam a rede usando *features* relacionados às características de interação de um complexo proteína-ligante [2]. Os melhores resultados foram reportados para DDFA-ALL, DDFA-RL, DDFA-ADV, DDFA-AD4 respectivamente, que são derivações da rede DDFA produzida por Arciniega & Lange (2014) [2] associados a diferentes programas de docking: Autodock4.2 (AD4) [74], Autodockvina1.2 (ADV) [70] e RosettaLigand3.4 (RL) [75].

1.1.6 Uma Introdução a Rede Neurais

Redes neurais (Neural Networks - NN) foram inspiradas, em parte, pela observação de sistemas de aprendizado biológico mais complexos, como o cérebro [69]. De forma didática, uma NN seria um grupo de neurônios interconectados (Figura 1.5). Ou seja, uma rede neural pode ser considerada uma técnica de processamento de dados que relaciona algum tipo de entrada de informação a uma saída de dados [69, 76, 77].

O modo como redes neurais são conectadas pode ser chamado de “arquitetura”. Existem diferentes tipos de arquiteturas. A mais comum em redes neurais possui (Figura 1.5): (1) uma camada de entrada (*input layer*), na qual *features* extraídas a partir dos dados alimentam as unidades de entrada (*units*); (2) uma camada escondida (*hidden layer*) na qual o dado é processado; e uma camada de saída (*output layer*) que retorna um valor de saída, sendo que esse valor normalmente pertence ao intervalo [0, 1]. Diversos outros tipos de arquiteturas foram propostas ao longos dos anos. Quando a NN possui duas ou mais camadas escondidas (*hidden layers*) dizemos que a rede é profunda (*deep neural network*) [18, 78].

As redes neurais também são normalmente divididas em *feed-forward* (alimentação para frente ou alimentação direta) e redes recorrentes (*recurrent neural network*)

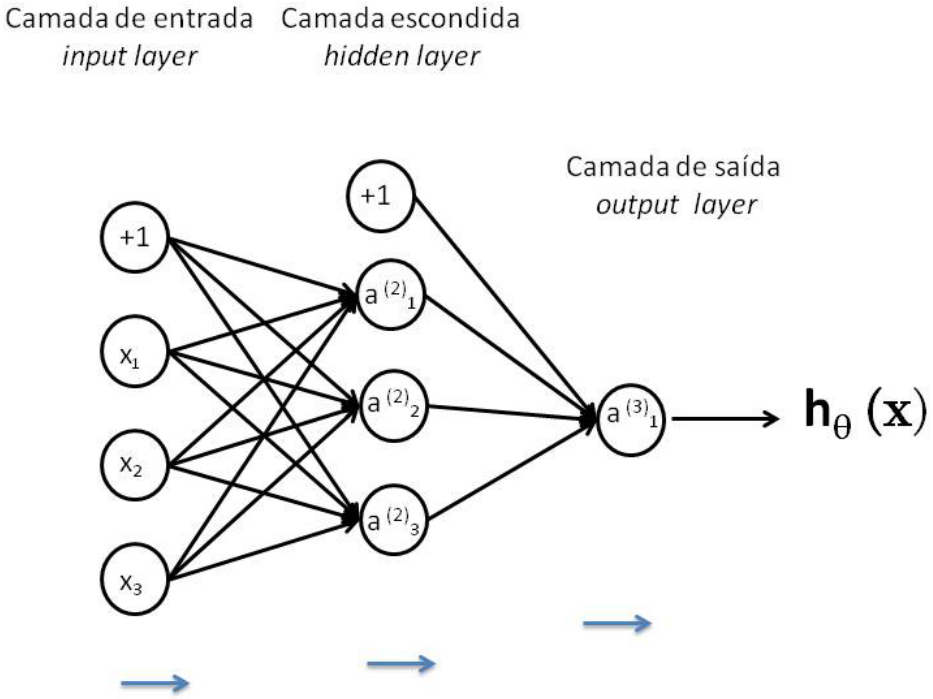


Figura 1.5: **Arquitetura comum em trabalhos usando redes neurais tradicionais para melhoramento de *Virtual Screening***. Uma rede do tipo *multilayer perceptron* com propagação do tipo *feed-forward* (setas em azuis), incluindo apenas uma camada de entrada, uma camada escondida e uma camada de saída. Onde x_1, x_2, x_3 representam as unidades da camada de entrada; $a_1^{(2)}$, $a_2^{(2)}$, $a_3^{(2)}$ representam as unidades da camada escondida e $a_1^{(3)}$ a unidade na camada de saída; e $h_\theta(x)$ representa o score de saída da rede.

[79].

Redes do tipo *feed-forward* são aquelas no qual o dado dentro da rede (ou sinal) percorre sempre em apenas uma única direção, da camada de entrada (*input layer*) para a camada de saída (*output layer*) (Figura 1.5). Uma conexão entre duas unidades possui um valor numérico (peso) representando a influência da unidade de entrada para a unidade de saída. Assim, os sinais de entrada são linearmente combinados com os vários pesos resultando em vários sinais de entrada para a segunda camada. Estes sinais de entrada são então passados adiante com o uso de uma função de ativação para produzir sinais de saída nas unidades da segunda camada [69, 78].

Redes recorrentes (*recurrent neural network*) possuem conexões do tipo *feed-forward* e adicionalmente unidades (*units*) com autoconexões ou conexões com unidades de camadas anteriores. Essa ação de recorrência atua como uma memória de curto prazo que permite que a rede lembre o que aconteceu em camadas anteriores (Figura 1.7). Historicamente, redes neurais do tipo recorrente têm sido menos influentes do que redes neurais do tipo *feed-forward*. Isso se deve em parte porque rede recorrentes são normalmente mais difíceis de se treinar. Porém, redes recorrentes são mais semelhantes ao modelo funcional cerebral se comparadas as redes do tipo *feed-forward*, o que torna esse tipo de rede interessante. Redes recorrentes têm ganhado

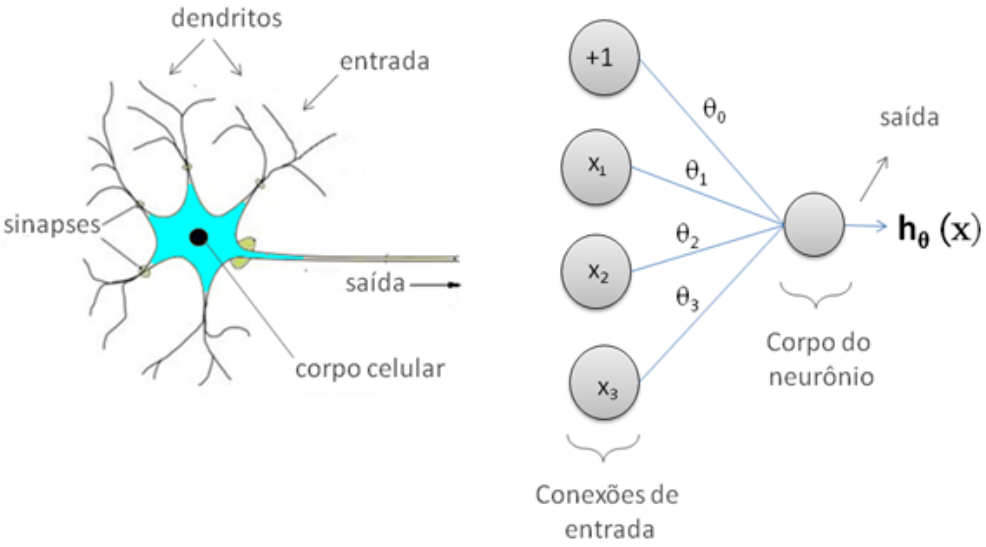


Figura 1.6: **Analogia de um neurônio artificial de NN a um neurônio de um sistema biológico**. A informação é dada a neurônios artificiais em uma NN por vias de entrada, ele fará alguns cálculos e dará um valor pela sua via de saída. Semelhante a conexões de neurônios em sistemas biológicos. Onde x_1, x_2, x_3 representam as unidades da camada de entrada; θ_0, θ_1, θ_2, θ_3 representam os pesos; e $h_\theta(x)$ calcula o score de saída da rede.

mais espaço nos últimos anos devido ao uso de novas estratégias de treinamento [80].

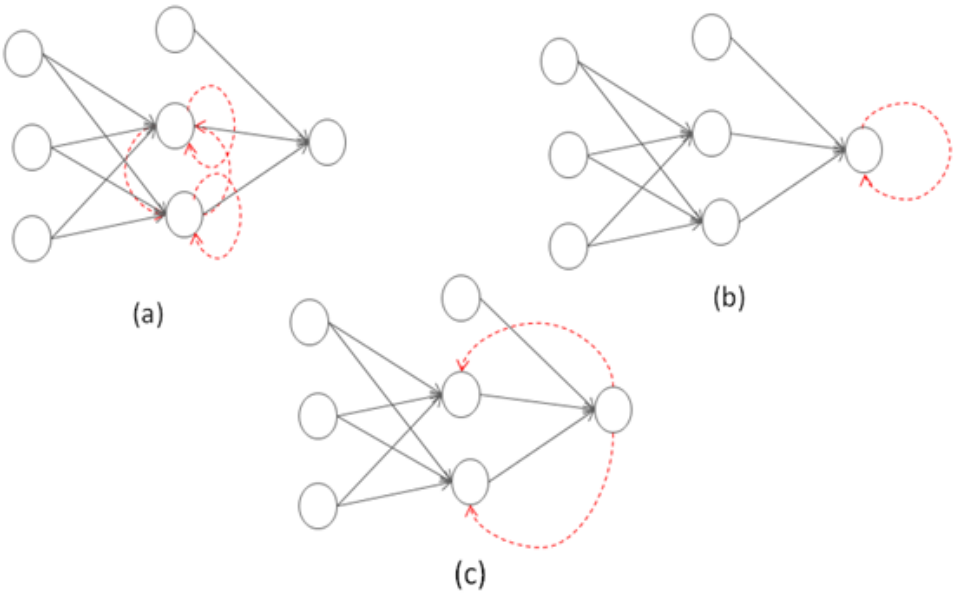


Figura 1.7: **Exemplos de um Multilayer Perceptron (MLP) com diferentes conexões de recorrência parcial.** Conexões do tipo recorrente são mostradas com setas tracejadas em vermelho: (a) autoconexões na camada escondida, (b) autoconexões na camada de saída, e (c) conexões da camada de saída para camada escondida. Combinações dessas conexões são possíveis. Fonte: Alpaydin (2014) adaptado por Pereira, J.C.

As redes neurais podem ser treinadas de diversas maneiras. A mais comum é

usando Gradiente Descente combinado ao algoritmo *Backpropagation* [81]. O *Backpropagation* é um algoritmo de aprendizagem para ajustes de parâmetros de uma rede neural dado um conjunto de treino específico. *Backpropagation* consiste em propagar o gradiente do erro da camada de saída para as demais camadas, ou seja, de trás para frente, vindo daí o nome *Backpropagation* que é uma abreviatura para *backward propagation of errors*. No algoritmo de *Backpropagation* o erro é computado da camada de saída para as demais camadas. É importante ressaltar que o erro não é associado à primeira camada, por que essa camada corresponde à camada de entrada que é constituída de *features* básicas obtidas a partir do conjunto de treino, e dessa forma não pode haver erro associados a elas [18, 19, 69, 78, 81].

Existem diversos tipos de sistemas de redes neurais, dentre eles:

(1) *Perceptron* [82] foi a primeira NN criada, possuindo certa semelhança com um neurônio humano (Figura 1.6). Este tipo é baseado em uma unidade (*unit*) que recebe um vetor de entradas (*inputs*) reais, calcula uma combinação linear destas entradas, e retorna como saída (*outputs*) 1 se o resultado é maior que algum limiar e 0 para o resultado contrário. Um único *Perceptron* pode expressar apenas decisões lineares, o que torna essa rede capaz de aprender representações de funções do tipo booleano AND, OR, $NAND$ e NOR, mas torna-se incapaz de resolver funções do tipo XOR, onde o valor é verdadeiro, se e somente se $x1 \neq x2$, envolvendo dessa forma uma solução não linear [69, 78, 82].

(2) *Multilayer Perceptron* (MLP) combinado ao algoritmo *Backpropagation* resolveu o problema do *Perceptron* com relação a soluções que envolvem não linearidade. O MLP é capaz de expressar uma diversa variedade de decisões não lineares. Esse tipo de rede pode ter uma ou mais camadas escondidas. Isso é possível porque camadas escondidas com seus próprios pesos podem ser dadas como entrada para a próxima camada escondida, calculando assim funções não lineares da primeira camada escondida e implementando funções mais complexas para serem dadas como entrada para a próxima camada escondida. Isso é possível graças à possibilidade de treino do MLP com o uso do gradiente descendente somado ao *Backpropagation* [18, 19, 78].

(3) Na rede convolucional (*Convolutional Neural Networks* - CNN ou ConvNet) [83] usa-se uma operação matemática chamada “convolução”. De forma simples, convolução é uma operação envolvendo duas funções com argumentos de valores reais. Redes convolucionais são redes neurais que usam uma convolução no lugar de uma multiplicação convencional de matrizes. Em redes convolucionais o primeiro argumento é referido como a entrada (matriz de dados) e o segundo argumento é definido como núcleo (usualmente uma matriz de parâmetros para serem aprendidos) e a saída é definida como um mapa de atributos (*features map*). Esse tipo de rede é normalmente utilizado em processamento de dados com uma topologia do tipo *grid* conhecida. Por exemplo, dados de series temporais podem ser pensados como uma *grid* unidimensional que recolhe amostras em intervalos regulares, ou dados de imagem podem ser pensados como uma *grid* bidimensional de pixels [18, 78, 83].

1.1.7 Uma introdução a *Deep Learning*

As estratégias de aprendizado de máquina tradicionalmente utilizadas dependem da forma como os dados do problema são representados. Nas abordagens de *virtual screening* com aprendizado de máquina, a fim de alimentar o algoritmo de aprendizado de máquina com informação que seja importante para a tarefa, os pesquisadores normalmente analisam a saída do programa de *docking* para identificar *features* manualmente. Embora este processo possa ser eficaz até certo ponto, a identificação manual de *features* é um processo árduo, e não pode ser aplicada em larga escala resultando na perda de informações cuja importância não é de fácil descoberta por parte dos pesquisadores. O que pode levar a um conjunto de *features* que não explicam toda a complexidade do problema [1, 4, 18, 78].

Uma alternativa para resolução do problema de extração manual de *features* é o uso de estratégias de *deep learning*. No aprendizado com *deep learning* várias camadas da rede são utilizadas para aprender diferentes níveis de representação dos dados de entrada (Figura 1.8) [18, 19, 78, 84, 85].

As estratégias de *deep learning* geralmente usam redes neurais artificiais que aprendem representações distribuídas dos dados de entrada. Em tais representações cada neurônio da rede participa da composição de diferentes conceitos. Uma das abordagem de maior sucesso da área de inteligência artificial (*Artificial Intelligence* - AI), *deep learning* permite que os computadores aprendam com a experiência e compreendam o mundo em termos de uma hierarquia de conceitos, no qual cada conceito é definido baseado na sua relação com conceitos mais simples [18, 78, 84].

Deep learning envolve múltiplos níveis de aprendizado de *features*, correspondendo a diferentes níveis de abstração. Aprendizado de *features* usa aprendizado de maquina não só para descobrir como a *feature* está mapeada como também para entender o que é a própria *feature*. *Features* aprendidas muitas vezes resultam em um desempenho melhor do que *features* que são extraídas de forma manual, e de certa forma, permitem que os sistemas de inteligência artificial possam se adaptar rapidamente a novas tarefas, com a mínima intervenção humana. Algoritmos de aprendizado de *features* podem descobrir um bom conjunto de atributos para uma tarefa simples em questão de minutos, ou uma tarefa complexa que pode levar de horas a meses [18, 78].

Nos últimos anos, abordagens usando *deep learning* têm mudado o campo de aprendizado de máquina e influenciado a nossa capacidade em entender como funciona a percepção humana sobre o mundo. Isso tem revolucionado áreas como reconhecimento de voz, compreensão de imagem, análise de sentimentos em textos e tradução automática de textos, além de despertar o interesse de grandes empresas como Google, Facebook, IBM, Microsoft, NEC, Baidu e outras na qual parte dos seus serviços e produtos desenvolvidos recentemente estão baseados em métodos que usam *deep learning* [18, 78, 84].

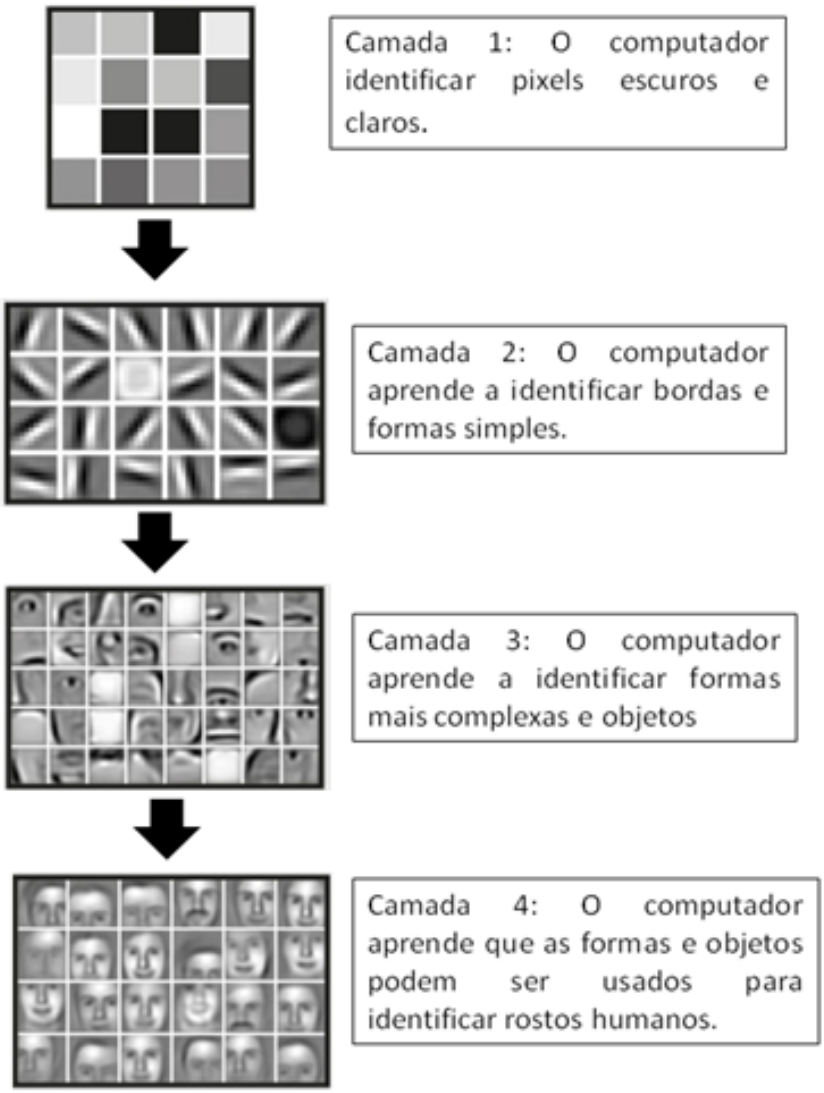


Figura 1.8: **Tarefa de reconhecimento facial usando *deep learning***. Estratégias de *deep learning* usam redes neurais com múltiplas camadas onde camadas mais profundas são capazes de aprender *features* mais abstratas e reconhecer padrões cada vez mais complexos. Fonte: Jones (2014) adaptado por Pereira, J.C.

1.1.8 Redes Neurais Convolucionais

Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs) ou ConvNets [86] foram desenhadas para trabalhar especificamente com dados que possuem matrizes multidimensionais [18, 83]. Um exemplo de matrizes multidimensionais são imagens coloridas, no qual são compostas por matrizes de pixels 2D em três níveis de cores [18].

O nome convolucional faz alusão à operação matemática convolução. A operação matemática de convolução envolve duas funções cuja saída é uma terceira função [80]. Por exemplo, utiliza-se para medir a localização de um objeto móvel no espaço por meio de um sensor. Esse sensor gera uma única saída $x(t)$, a posição do objeto no tempo t, onde x e t são valores reais. Suponha que o sensor usado gere

ruído durante a medição. Nesse caso, a média de todas a medições é utilizada para obter o menor ruído nos dados. Pesos são utilizados para diferenciar medições recentes de medições antigas, assim temos a função de pesos $w(a)$, onde a é a idade da medição. Quando aplicamos uma operação de média ponderada para cada momento, obtemos uma nova função s que fornece uma estimativa da posição do objeto [80], segundo:

$$s(t) = \int x(a)w(t-a)da \tag{1.1}$$

Esse tipo de operação matemática é conhecida como convolução, é representada pelo o sinal de $*$:

$$s(t) = (x * w)(t) \tag{1.2}$$

De um modo geral, a operação de convolução é definida para quaisquer funções para as quais a integral acima é definida e pode ser usada para outros propósitos além de tomar médias ponderadas [80].

A arquitetura da ConvNets consiste basicamente em quatro passos: (1) Camada convolucional (*Convolutional Layer*); (2) Camada do detector (*Rectied Linear* - RELU); (3) Camada *pooling* (*Pooling Layer*); (4) Camada totalmente conectada (*Fully-Connected Layer*) [18, 80, 83, 85] (Figura 1.9).

A camada convolucional (*Convolutional Layer*) é utilizada em arquiteturas cuja entrada possui tamanho variável, como por exemplo reconhecimento de caracteres manuscritos e classificação de imagens [18, 80, 83]. As características centrais que compreendem a camada convolucional são conexões esparsas (*sparse connectivity*), parâmetros compartilhados (*parameter sharing*) e representações equivariantes (*equivariant representations*) [80].

Em redes neurais totalmente conectadas como *multilayer perceptron* todos os neurônios (ou núcleos) de uma determinada camada, como por exemplo a camada de entrada, interagem com todos os neurônios da próxima camada (Figura 1.10). Porém, redes convolucionais utilizam o que chamamos de conexões esparsas (também conhecido como interações esparsas ou pesos esparsos), no qual neurônios de uma camada irão interagir com apenas um conjunto (*kernel*) de neurônios pertencentes à próxima camada [18, 80, 83] (Figura 1.10.)

Uma das vantagens de usar conexões esparsas está relacionado à redução do tamanho da informação de entrada [18, 80]. Tomamos a figura 1.9 como um exemplo, onde se supõe que um grupo de herpetólogos utilizou uma ConvNet para distinguir tipos de cobras utilizando apenas imagens. A ConvNet recebe como entrada a imagem de uma cascavel (*Crotalus durissus*) constituída de uma matriz 28 X 28 de intensidade de pixels. Suponha que cada neurônio na camada de entrada representa um pixel da imagem. Cada neurônio da próxima camada receberá como entrada a representação

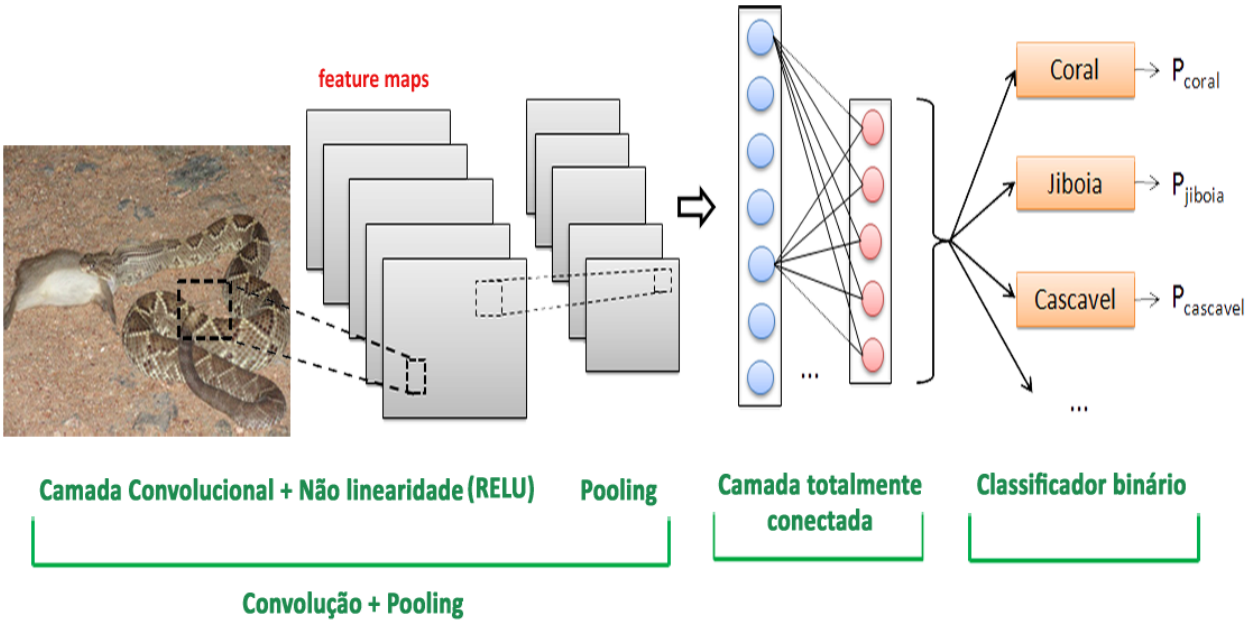


Figura 1.9: **Arquitetura básica de uma ConvNet**. Os componentes típicos encontrados em uma Rede Neural Convolucional. Camada de entrada, que contém imagens formadas por pixels; Camada convolucional, que executa várias operações de convolução em paralelo para produzir um conjunto de ativações lineares; Camada detector para inserir uma não linearidade, por exemplo RELU (*rectied linear*); Camada *Pooling* com o objetivo de modificar a saída para as próximas camadas, como por exemplo fixar um tamanho de saída; Camada totalmente conectada, como por exemplo um *multilayer perceptron* - MLP; Camada de saída que pode ser um classificador binário.

de um pequeno conjunto de 4 X 4 pixels, total de 16 pixels, provenientes da camada de entrada. Esse pequeno conjunto é movido pixel a pixel da direita para esquerda e de cima para baixo. Ou seja, a imagem será representada para as próximas camadas sempre em um conjunto menor de pixels se comparado à camada anterior (Figura 1.11). A quantidade de pixels que o conjunto é movido é denominado *stride*. Nesse caso o *stride* é igual a 1, porém há modelos em que o tamanho do *stride* é maior [83].

Geralmente o uso de conexões esparsas reduz o uso de memória do modelo e melhora sua eficiência estatística [18,80]. Por exemplo, ao se processar uma imagem, a imagem de entrada pode conter milhares ou milhões de pixels, porém características importantes podem ser representadas por apenas um conjunto pequeno de pixels. No caso da imagem da cascavel, o conjunto de pixels que representa o guizo da cobra é crucial para diferencia-lá de outros tipos de cobras (Figura 1.9).

Uma outra vantagem do uso de camadas convolucionais é a possibilidade de compartilhar parâmetros (pesos). Em outras palavras, o mesmo conjunto de pesos é aplicado para cada conjunto da matriz de entrada (Figura 1.12), o que difere das redes neurais tradicionais onde cada elemento da matriz de pesos é utilizado somente uma

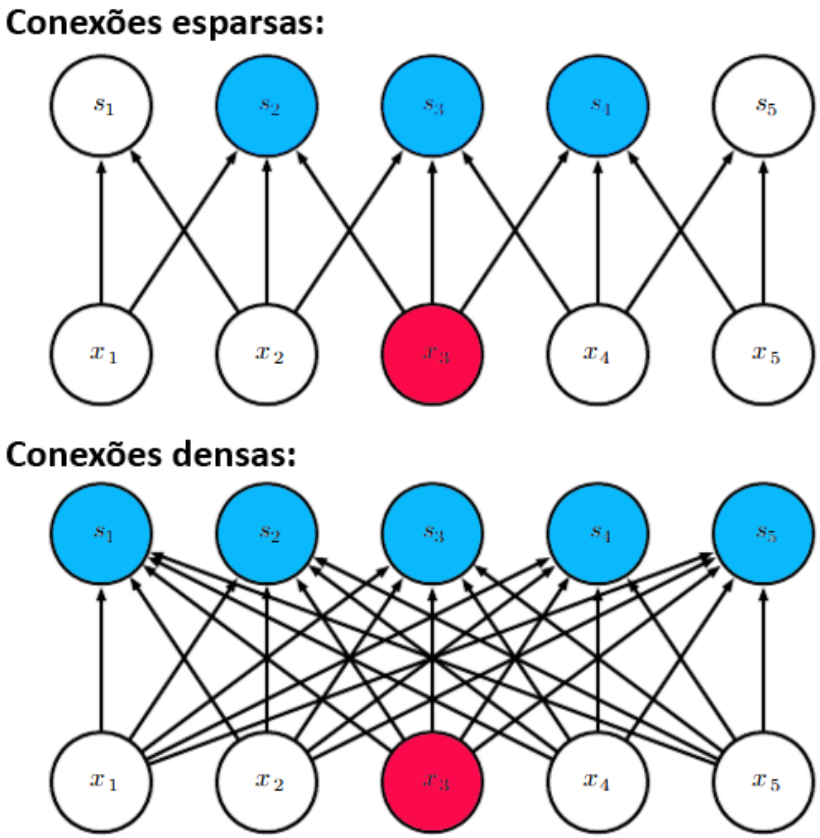


Figura 1.10: **Conexões Esparsas vs Conexões Densas** Em conexões densas o neurônio x_3 interage com todos os neurônios na camada s, ou seja, uma unidade na camada x altera todas as unidades na camada s. Em conexões esparsas o neurônio x_3 interage com um conjunto definido de neurônios da camada s, no caso um conjunto de três neurônios. Fonte Goodfellow *et al.* (2006) adaptado por Pereira J.C.

vez para calcular a saída de uma camada (Figura 1.6) [80].

Quando a camada convolucional usa parâmetros (pesos) compartilhados significa que ao invés de ser aprendido um conjunto separado de parâmetros para cada localidade, como ocorre em redes neurais tradicionais, a camada aprenderá um conjunto de parâmetros para toda a extensão do objeto de entrada [80]. Por isso, redes neurais convolucionais são consideradas drasticamente mais eficientes do que a multiplicação matricial densa em termos dos requisitos de memória e da eficiência estatística [18, 80]. Por exemplo na figura 1.12, a matriz de pixels de tamanho 28 X 28 é dada como entrada para uma ConvNet qualquer, um conjunto de pesos de tamanho 4 X 4 é aplicado (filtro 1), o resultado dessa operação é conhecido como *feature map*. Nesse caso cada *feature map* resultante possui 16 parâmetros compartilhados (4 x 4 =16) mais uma *bias* compartilhada, em um total de 17 parâmetros. Suponha que essa ConvNet possua 30 *feature maps*, o total de todos os parâmetros da camada convolucional seria 510 (30 x 17 = 510). Porém, caso fosse utilizada uma rede neural totalmente conectada (*full layer connected*) para a mesma entrada de dados de tamanho 784 (28 X 28), com um mínimo de 40 parâmetros e consequentemente 40 bias teríamos um total 31.400 parâmetros (784 x 40 $+$ 40 = 31.400).

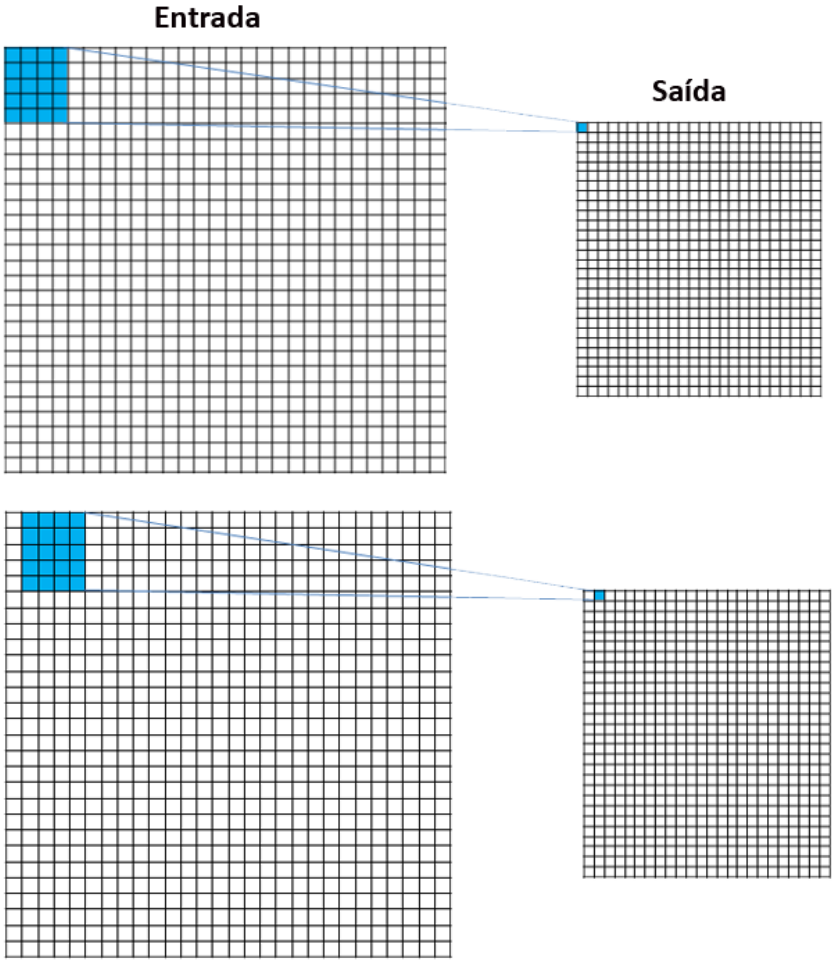


Figura 1.11: **Representação de uma imagem em uma matriz de 28 X 28 pixels usando conexões esparsas**. A representação de um conjunto de 16 pixels é dada como entrada para a camada de saída usando *stride*=1. O subconjunto irá se mover um pixel para direita e para baixo. Note que a camada de saída 625 unidades é menor que a camada de entrada 784 unidades.

Parâmetros compartilhados além de reduzir uso de memória e produzir maior eficiência da rede também possuem a propriedade de equivariância que consiste na invariância de uma função, dada uma mudança em uma entrada a saída irá transmitir essa mudança da mesma forma [80, 83]. Tomando o mesmo exemplo da ConvNet que classifica cobras recebendo como entrada imagens, no caso da cobra cascavel uma das características mais importantes para distingui-lá de outras cobras é a presença de um gizo no final de sua cauda. Suponha que a ConvNet recebe como entrada imagens em que o gizo da cascavel pode estar em diferentes posições (Figura 1.13), a rede deverá ser capaz de pontuar de forma mais expressiva essa característica na imagem independente de sua posição no espaço.

Camadas *Pooling* (*Pooling layers*) usam uma função do tipo *Pool* para alterar a saída das camadas convolucionais de forma que essa saída seja adequada para a próxima camada da ConvNet [80, 83]. Uma função *Pooling* substitui a saída por

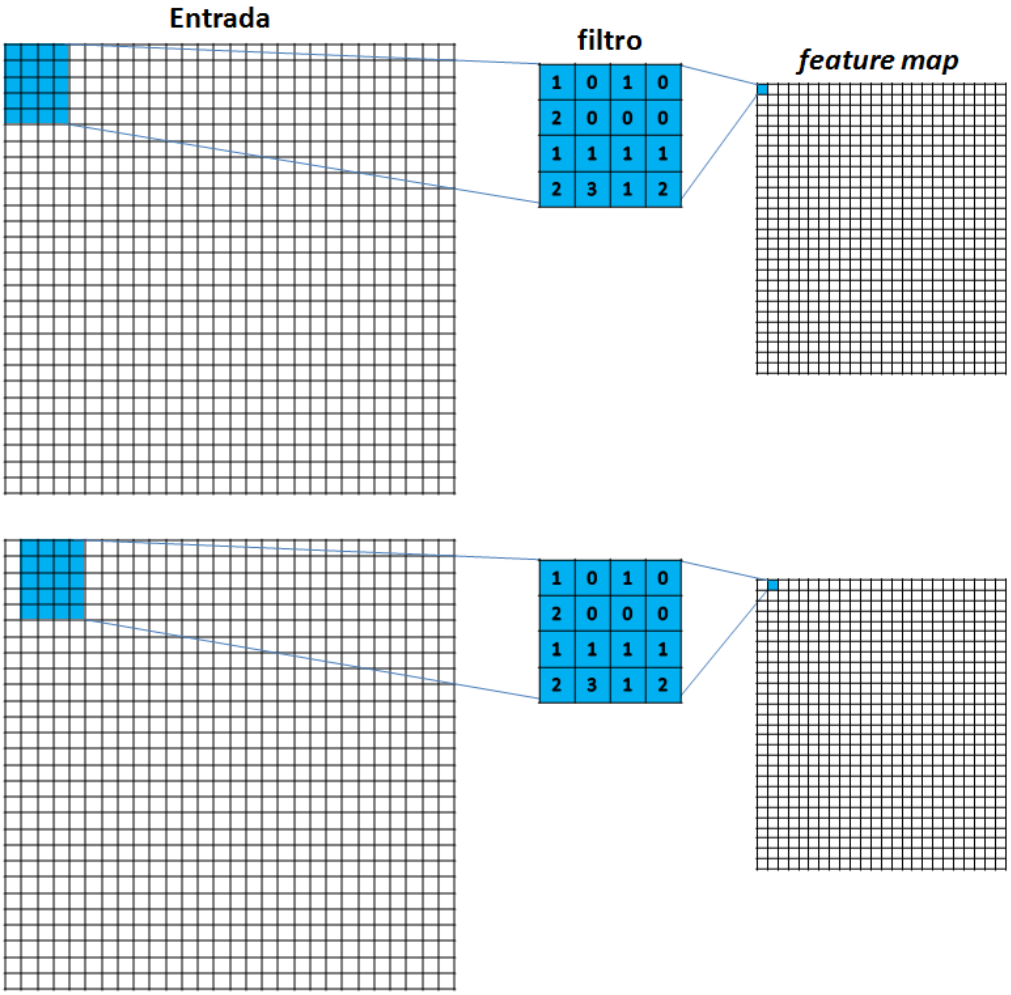


Figura 1.12: **Representação de uma imagem em uma matriz de 28 X 28 pixels usando pesos compartilhados**. A representação de um conjunto de 16 pixels é dada como entrada para a camada de saída usando *stride*=1. A cada conjunto é aplicado um filtro que corresponde a um conjunto de pesos de tamanho 4 X 4. O filtro percorre toda a extensão da matriz 28 X 28 pixels. A saída desse operação é chamada de *feature map*. Em redes convolucionais são utilizados mais de um filtro e consequentemente é gerado mais de um *feature map*.

um resumo estatístico de saídas próximas. Dessa forma, ConvNets podem processar entradas de tamanhos distintos, como por exemplo imagens de tamanhos diferentes.

Funções *Pool* diminuem o tamanho da representação, além de diminuir a quantidade de parâmetros computados em uma rede. Isso é possível por que funções *Pool* extraem somente a informação importante proveniente das camadas convolucionais [18]. Como exemplo, a operação de *max pooling* [87] reporta o valor máximo da saída dentro de uma vizinhança retangular [80]. Tomemos como exemplo a Figura 1.14, a operação de *max pooling* é aplicada de forma independente para cada *feature map*. Um filtro de tamanho 2 X 2 percorre cada *stride* de tamanho 2 para cada um dos *feature maps* de forma a computar o valor máximo para cada conjunto. Ou seja, o conjunto de pixels da imagem que recebeu uma pontuação maior será selecionado e consequentemente pode pertencer a uma característica importante para

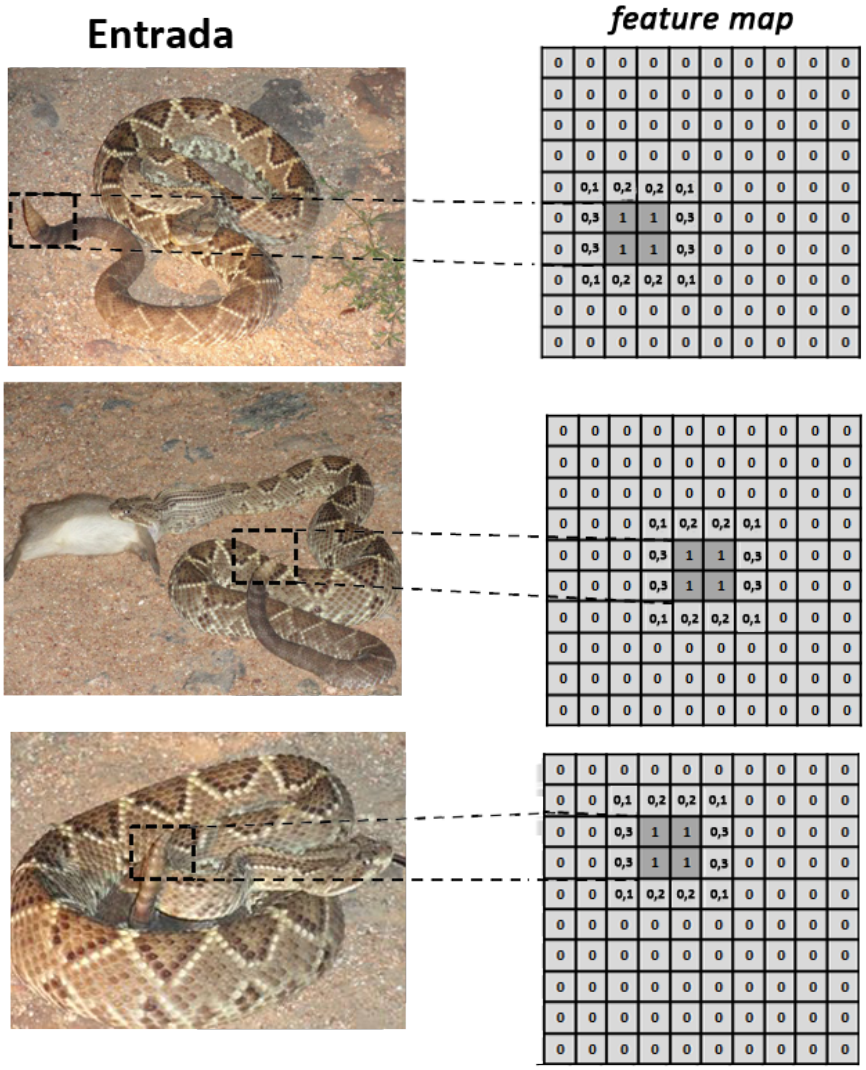


Figura 1.13: **Propriedade de equivariância em camadas convolucionais**. Figura esquemática da propriedade de equivariância em redes convolucionais. Uma ConvNet recebe como entrada diferentes fotos de uma cascavel. Como a camada convolucional usa pesos compartilhados para toda a extensão da imagem, ela é capaz de inferir uma pontuação elevada para o gizo da cobra independente de sua localização no espaço.

diferenciar aquele objeto de outros. A exemplo do gizo como indicativo do tipo de cobra no caso a cascavel na Figura 1.9.

Tipos de operações *poolings* mais utilizadas em redes convolucionais são: 1) *max pooling*; 2) L^2 *norm pooling*, que consiste na soma das raizes quadradas de cada conjunto; 3) *average pooling*, valor médio do conjunto [80, 88].

Camada detector, nessa etapa cada ativação linear é executada através de uma função de ativação não linear, com intuito de inserir uma não linearidade ao conjunto de dados. Uma das funções mais utilizadas em camadas detector é o *Rectified Linear Unit* - RELU, que é uma ativação em limiar de matriz formada por zeros [80, 89]. Algumas arquiteturas de redes convolucionais não utilizam camadas detector.

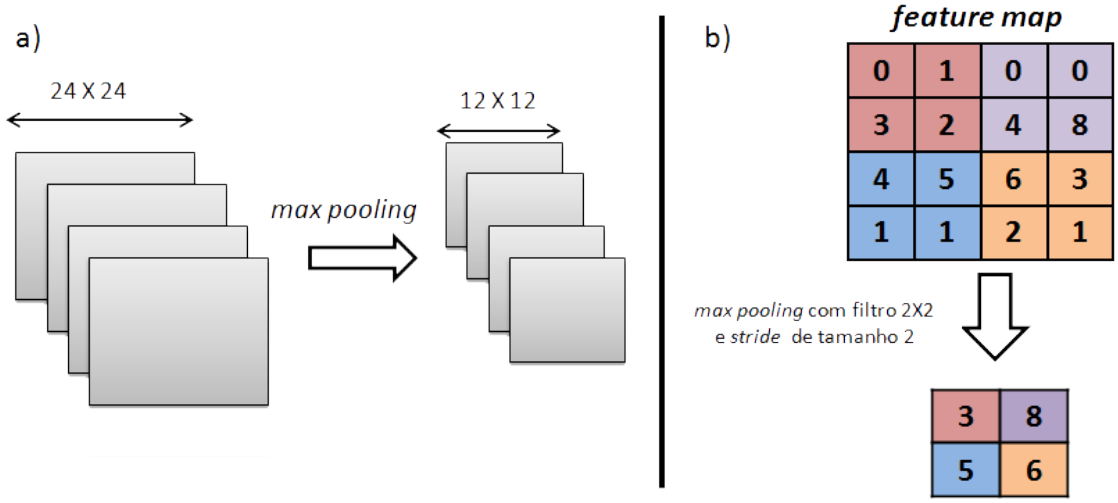


Figura 1.14: **Representação esquemática da operação *max pooling***. a) A operação de *max pooling* é aplicada para todas as *feature maps* de forma a compactar informação. b) Na operação de *pooling* um filtro de tamanho 2 X 2 é aplicado ao *feature map* em *stride* de tamanho 2, a *pool* do tipo *max* consistem em selecionar o maior valor para cada conjunto.

A camada totalmente conectada (*fully-connected layer*) é conectada ao resultado da operação de *pooling* e subsequentemente aos neurônios da camada de saída. Nesse tipo de camada os neurônios possuem conexões do tipo densa (Figura 1.10), no qual cada neurônio de uma camada totalmente conectada é conectado com todos os neurônios da camada seguinte. Um exemplo bastante comum da arquitetura de uma camada totalmente conectada é o *Multilayer Perceptron* - MLP (ver seção 1.1.6).

Redes Convolucionais vêm sendo utilizadas em larga escala desde da década de 1990, principalmente para reconhecimento de voz [90] e leitura de documentos [83]. Porém, ganhou grande visibilidade a partir da segunda década dos anos 2000 e especificamente após a competição ImageNet em 2012 cujo principal objetivo era o reconhecimento de imagens. [91]. Nessa competição os autores utilizaram uma rede convolucional somada a uma nova técnica de regularização chamada *dropout* [92], além do uso mais eficiente de GPUs (*Graphics Processing Unit*) e camadas detector RELUs (*Rectified Linear Unit*) [91]. A ConvNet proposta pelos autores analisou mais de um milhão de imagens e as distribuiu em 1.000 diferentes tipos de classes. O que rendeu vasta notoriedade tanto para os autores quanto para redes convolucionais foi a diferença em porcentagem de erro do primeiro para o segundo colocado em torno de 75% [91].

Atualmente, Redes Neurais Convolucionais (ConvNets) são amplamente utilizadas em diversas tarefas como: reconhecimento de imagens [91]; reconhecimento de sinais de trânsito [93]; reconhecimento de faces [94]; segmentação de imagens biológicas [95]; descoberta e reposicionamento de novos fármacos [96, 97]; classificação de sentimentos em texto [98, 99]; veículos autônomos [100, 101]; dentre outros.

1.1.9 Deep Learning no Processo de Descoberta de Novos Fármacos

Recentemente *deep learning* (DL) vem despertando o interesse tanto da comunidade científica quanto de grandes indústrias farmacêuticas, como uma alternativa viável para auxiliar nos processos de descoberta de novos fármacos. Um dos primeiros trabalhos que utilizaram abordagens de *deep learning* com sucesso foi o vencedor da competição oferecida pela Merck em 2012 [38] que usou estratégias de *deep learning* para resolver problemas relacionados a QSAR (*Quantitative Structure-Activity Relationships*). Dois anos depois, Dahl *et al.* [102] desenvolveram uma *multi-task deep neural network* para prever propriedades biológicas e químicas diretamente a partir da estrutura molecular do ligante.

Trabalhos recentes ligados a *multi-task deep neural network* foram empregados para prever farmacóforos direcionado a sítios de ligação [103] e na previsão de toxidade [104]. Em 2015, Ramsundar *et al.* [105] utilizaram *massivily multi-task deep neural network* associado a *fingerprints* para predizer atividade de ligantes.

A importância de DL para descoberta de novos fármacos também é ressaltada envolvendo a determinação de propriedades associadas a fármacos tais como o trabalho desenvolvido por Lusci *et al.* [106] que usaram um conjunto de *Recursive Neural Networks* para prever a solubilidade de fármacos em sistemas como água e o trabalho de Duvenaud *et al.* [107] que utilizaram *Convolutional Networks* associado a representações gráficas para aprender molecular *fingerprints*.

O nosso trabalho, até o presente momento, é o primeiro trabalho usando *Deep Learning* em específico redes convolucionais para melhorar *virtual screenng* baseado em *docking* (*Docking-based Virtual Screening*-DBVS) [96](Apêndice A). Trabalhos para melhoramento de DBVS têm usado somente redes neurais tradicionais ou rasas com *features* desenhadas por humanos (seção 1.1.5). A nossa abordagem foi citada em um trabalho recente desenvolvido por Gonczarek *et al.* [97] para melhorar DBVS. Os autores abordaram o problema processando de forma separada o ligante e o sítio de ligação no receptor, afim de gerar *fingerprints* do complexo proteína-composto usando o que os autores chamaram de uma abordagem similar a redes convolucionais. Por fim, os autores aplicaram uma rede neural totalmente conectada (MLP) (seção 1.1.6) para aprender a classificar ligantes ativos de *decoys*.

1.2 Objetivos

1.2.1 Objetivo Geral

Estabelecer um novo método para melhoramento *virtual screening* baseado em *docking* usando estratégias de *deep learning*.

1.2.2 Objetivos Específicos

- Analisar o desempenho das metodologias de *docking* para técnicas de *virtual screening* em um conjunto de dados padrão, o DUD (*Directory of Useful Decoys*);
- Monitorar o desempenho de estratégias de aprendizado (*deep learning*) para classificar ligantes ativos;
- Testar o método proposto com um subconjunto de dados externo DUD-E (*Directory of Useful Decoys: Enhanced*);
- Comparar o resultado obtido com a estratégia adotada usando *deep leaning* com outras estratégias de melhoramento de *virtual screening* baseado em *docking*.

1.3 Material e Métodos

1.3.1 Ideia Geral

O presente projeto propõe uma abordagem para melhoramento de *virtual screening* baseado em estrutura usando a abordagem de *deep learning*. Na estratégia proposta, os seguintes passos são executados (Figura 1.15):

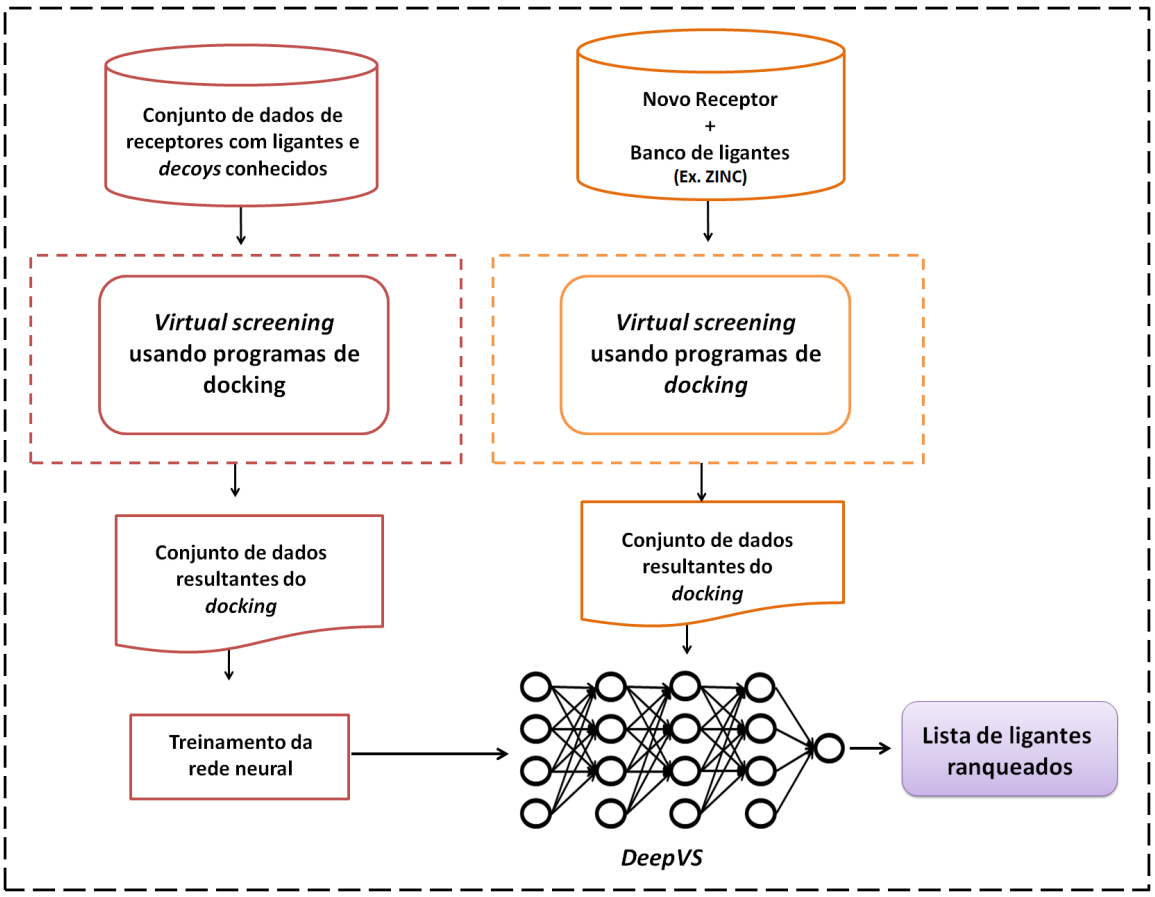


Figura 1.15: **Fluxograma da metodologia proposta no projeto.**

1. *Virtual screening* baseado em *docking* é aplicado para todos os receptores disponíveis em um conjunto onde ligantes e *decoys* de cada receptor são conhecidos;

2. A saída do *docking* (pose dos receptores e ligantes/*decoys*) para todos os receptores é utilizada para treinar uma rede neural profunda (*Deep Neural Network* - DNN). Nesse trabalho usamos uma rede neural convolucional, à qual nomeamos DeepVS;

3. Uma vez treinada, a DeepVS poderá ser usada para melhorar a qualidade do *virtual screening* de ligantes em novos receptores a partir do processamento da saída do *docking* para esse novo receptor e respectivo conjunto de compostos.

1.3.2 DeepVS

Em nossa abordagem a DeepVs recebe como entrada os dados estruturais de um complexo proteína-composto (camada de entrada) e produz um *score* capaz de distinguir potenciais ligantes de *decoys* (camada de saída). Conforme detalhado na Figura 1.16, o primeiro passo na rede consiste em extrair informação do contexto local de cada átomo do ligante (Primeira camada escondida).

O contexto de um átomo compreende dados estruturais básicos (*features* básicas) que estão relacionados a distâncias entre os átomos vizinhos, tipos de átomos, cargas atômicas parciais e resíduos associados. Subsequentemente, os valor das *feature* básicas advindos do contexto de cada átomo são convertidos em vetores de *features* que são aprendidos pela *deep neural network*. Em seguida, uma camada convolucional é empregada para sumarizar a informação advinda dos contextos de todos os átomos do composto. Essa camada gera uma representação vetorial distribuída do complexo proteína-composto r (segunda camada escondida). Subsequentemente, a terceira camada escondida da DeepVS, que é uma camada totalmente conectada (*fully connected layer*), processa r com o objetivo de gerar *features* ainda mais abstratas do complexo proteína-composto.

Finalmente, na última camada (camada de saída), a representação do complexo proteína-composto é dada como entrada para a função *softmax* que computa a probabilidade do composto ser ativo ou inativo. No Algoritmo 1, apresentamos um pseudocódigo com os passos do processo de *feedforward* executado pela DeepVS.

1.3.2.1 Contexto do Átomo

O complexo proteína-ligante resultante do procedimento de *docking* precisa ser processado para gerar a entrada para a DeepVS. A camada de entrada da rede usa informação do contexto de cada átomo do ligante. O contexto de um átomo "*a*" é definido por um conjunto de *features* básicas extraídas de sua vizinhança. Tal vizinhança consiste no próprio átomo "*a*", nos k_c átomos do ligante mais próximos de "*a*" e nos k_p átomos da proteína mais próximos de "*a*", onde k_c e k_p são hiperparâmetros que devem ser definidos pelo usuário. A ideia de usar informação dos átomos vizinhos mais próximos do complexo proteína-composto vem sendo explorada com sucesso em trabalhos relacionados à descoberta e planejamento de novos fármacos [108, 109].

As *features* básicas adquiridas a partir do contexto do átomo incluem os tipos de átomos, cargas atômicas parciais, tipos de resíduos (aminoácidos) associados e a distância calculada entre o átomo de referência e cada um dos seus vizinhos. Por exemplo, na Figura 1.17, para o nitrogênio (N_3) do ligante THM (timidina), a vizinhança com k_c = 3 e k_p = 2 consiste nos átomos N_3, H e C do ligante e dos átomos OE e CD do aminoácido Gln215 da proteína Timidina Quinase (TK - ID_PDB [42]: 1kim). Nesse caso, o contexto do átomo N_3 possui os seguintes valores para cada característica:

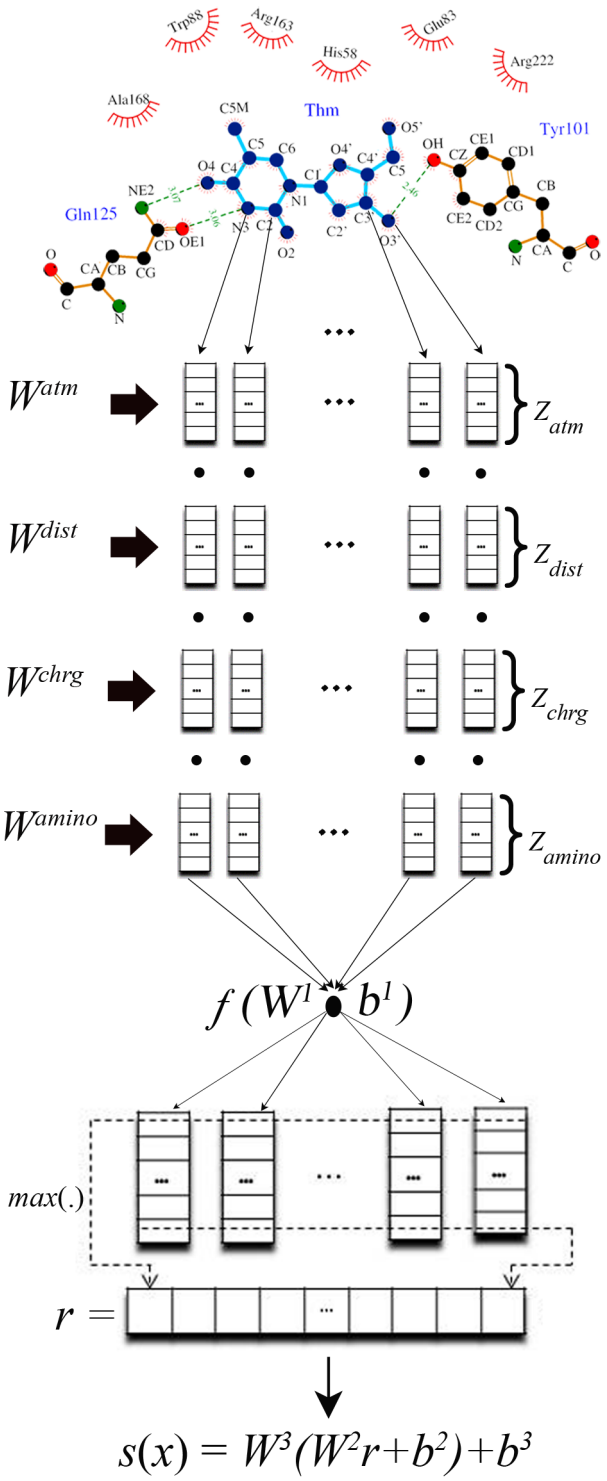


Figura 1.16: **Desenho esquemático da arquitetura da DeepVS.** Nesta figura, é usado como exemplo o ligante THM (Thymidine) complexado com a proteina TK -Thymidine kinase (ID_PDB: 1kim) átomos do ligante estão marcados em azul escuro e suas interações em azul claro.

Algorithm 1 Processo *feedforeward* utilizado na DeepVS

Input: complexo proteína-composto x, onde o composto contém m átomos
Given: parâmetros de treinamento da rede $W^{atm} \in \mathbb{R}^{d^{atm} \times |A|}$, $W^{dist} \in \mathbb{R}^{d^{dist} \times |D|}$, $W^{chrg} \in \mathbb{R}^{d^{chrg} \times |C|}$, $W^{amino} \in \mathbb{R}^{d^{amino} \times |R|}$, $W^1 \in \mathbb{R}^{cf \times |z_i|}$, $W^2 \in \mathbb{R}^{h \times |cf|}$, $W^3 \in \mathbb{R}^{2 \times |h|}$, $b^1 \in \mathbb{R}^{cf}$, $b^2 \in \mathbb{R}^h$, $b^3 \in \mathbb{R}^2$
Z = [ ]
// gera a representação do contexto dos átomos (1ª camada escondida)
for i=1 to m **do**
    z_{atm} =colunas de W_{atm} correspondentes aos tipos de átomos vizinhos ao átomo i
    z_{dist} =colunas de W_{dist} correspondentes aos tipos de átomos vizinhos ao átomo i
    z_{chrg} =colunas de W_{chrg} correspondentes aos tipos de átomos vizinhos ao átomo i
    z_{amino} =colunas de W_{amino} correspondentes aos tipos de átomos vizinhos ao átomo i
    //Representação dos contextos do átomo i
    z_i = {z_{atm}; z_{dist}; z_{chrg}; z_{amino}}
    $Z.add(z_i)$
end for
// U é inicializada com zeros
U = $[..] \in \mathbb{R}^{cf \times m}$
// camada convolucional (2ª camada escondida)
for i=1 to m **do**
    $U[:, i] = f(W^1 Z[i] + b^1)$
end for
// max pooling baseado nas colunas
$r = max(U, axis = 1)$
// 3ª camada escondida e camada de saída
score = $W^3 \left(W^2 r + b^2\right) + b^3$
// retorna o score normalizado
return $\frac{e^{score[1]}}{e^{score[0]} + e^{score[1]}}$

- *FEATURE* BÁSICA TIPO DE ÁTOMO = [N; H; C; OE; CD]
- *FEATURE* BÁSICA CARGA = [-0.2359; 0.1594; 0.3146; -0.6086; 0.6051]
- *FEATURE* BÁSICA DISTÂNCIA = [0.0000; 1.0000; 1.3382; 3.0615; 3.8989]
- *FEATURE* BÁSICA TIPO DE AMINOÁCIDO = [Gln; Gln]

As *features* básicas (tipo de átomo, carga atômica parcial, distância e resíduos associados) foram escolhidas baseadas em características bioquímicas e biofísicas de interações entre moléculas. Por exemplo, como interações entre moléculas incluem contribuições atrativas e repulsivas entre cargas elétricas parciais, grande parte das propriedades físicas e químicas está relacionada com a distribuição parcial de cargas em uma molécula ou em um grupamento. Os resíduos associados podem definir o caráter hidrofílico ou hidrofóbico da cavidade onde se liga o composto. Outro exemplo é o uso da distância como *feature*, dado que todos os termos da energia potencial de interação dependem da distância de separação dos átomos. Interações como íon-íon,

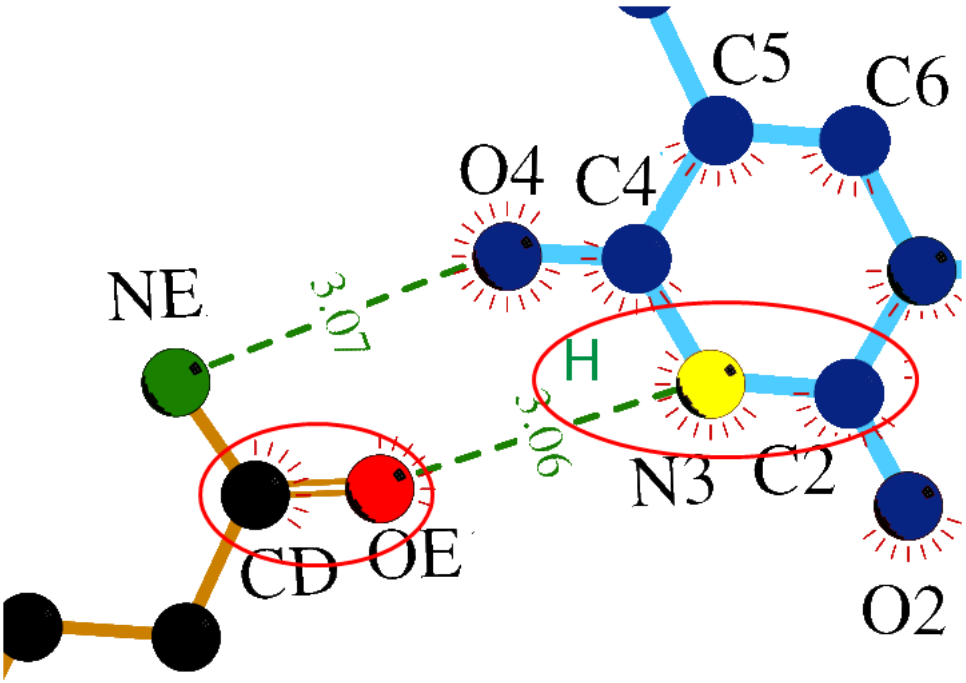


Figura 1.17: **Contexto do átomo do ligante THM (timidina).** Círculos vermelhos representam respectivamente os dois vizinhos mais próximos do átomo N_3 no ligante THM (timidina) e os dois vizinhos mais próximos do N_3 na proteína TK (ID_PDB: 1kim). Figura construída utilizando o programa LigPlot.

dipolo-dipolo, dipolo-dipolo induzido e de dispersão variam com o inverso da distância de separação entre átomos, dessa forma a *feature* distância passa a ser importante para caracterizar interações intermoleculares. Acreditamos que a DeepVS é capaz de utilizar a informação contextual básica dos átomos para gerar *features* mais abstratas e efetivas para identificação de ligantes.

1.3.2.2 Representação do Contexto do Átomo

A primeira camada escondida da rede DeepVS transforma cada valor das *features* básicas do contexto de um átomo em um vetor de números reais, o qual é normalmente conhecido na literature de *deep learning* como *embedding* [110]. Esses vetores contêm *features* que são automaticamente aprendidas pela rede. Para cada tipo de *feature* básica existe uma matriz W de *embeddings* que encapsula os vetores para os valores possíveis daquela *feature* básica. Dessa forma, as matrizes W^{atm}, W^{dist}, W^{chrg}, W^{amino} contêm respectivamente os *embeddings* das *features* tipo de átomo, distância, carga atômica parcial e resíduo associado. Essas matrizes constituem as matrizes de pesos da primeira camada escondida e são inicializadas com números aleatórios antes do treinamento da rede.

Cada coluna da matriz $W^{atm} \in \mathbb{R}^{d^{atm} \times |A|}$ corresponde ao vetor de *features* de um tipo de átomo específico, onde A é o conjunto de tipos de átomos, d^{atm} é o número de

features que são aprendidas pela rede e é um hiperparâmetro definido pelo usuário. Dado o contexto de um átomo a, a rede transforma cada valor da *feature* básica tipo de átomo no seu respectivo vetor de *features* e concatena esses vetores para gerar o vetor *representação do tipo de átomo* (z_{atm}). A Figura 1.18 ilustra a criação do vetor z_{atm} para o contexto do átomo N_3 ilustrado na Figura 1.17.

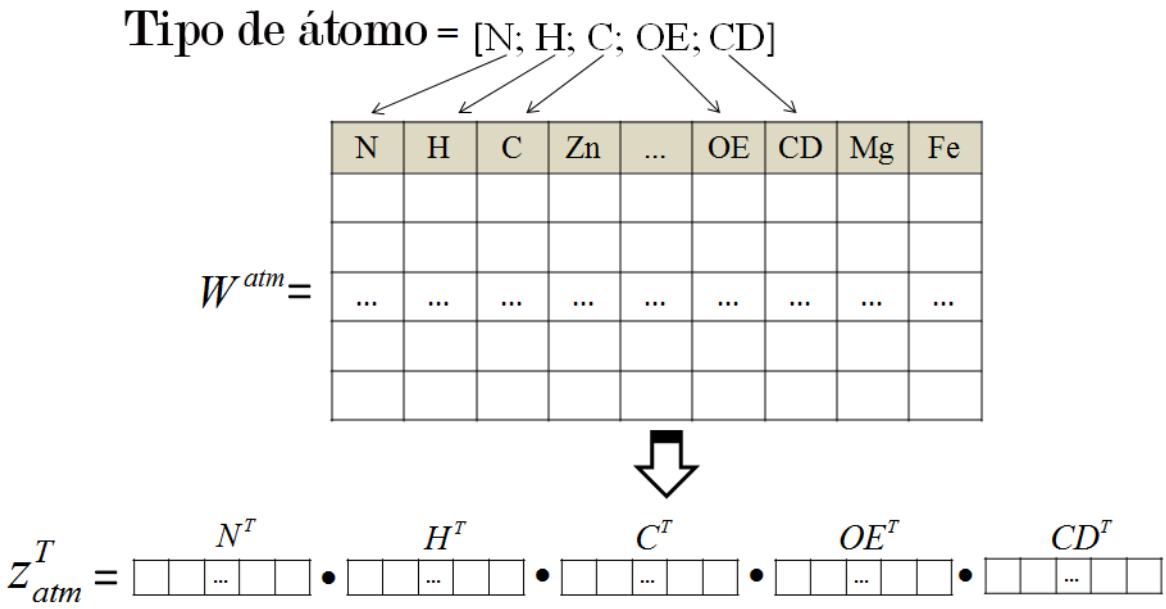


Figura 1.18: **Ilustração da construção do vetor representação do tipo de átomo (z_{atm}) usando a matriz de *embeddings* W^{atm}** para o contexto do átomo N_3 advindo do ligante THM (Thymidina). O símbolo • indica uma operação de concatenação.

Os valores da *feature* básica carga precisam ser discretizados antes de serem dados como entrada para a DeepVS. A discretização é necessária por que precisamos de um conjunto finito de valores para serem usados nas matrizes de *embeddings*. Para realizar a discretização são definidos um valor mínimo c_{min} e um valor máximo c_{max} de carga, e são construídos intervalos de 0,05 entre esses valores. Por exemplo, no caso de cargas do tipo *Gasteiger*, com $c_{min} = -1$ e $c_{max} = 1$, haverá 40 intervalos de carga possíveis. Cada valor da *feature* básica carga é mapeado para o índice do intervalo ao qual ele pertence. Na rede, a matriz $W^{chrg} \in \mathbb{R}^{d^{chrg} \times |IC|}$ contém um vetor de *features* (coluna) para cada intervalo de carga i, onde IC é o conjunto de intervalos de carga e d^{chrg} é o número de *features* que são aprendidas pela rede, onde o d^{chrg} é um hiperparâmetro definido pelo usuário. Dado o contexto de um átomo a, se transforma cada valor da *feature* básica carga no seu respectivo vetor de *features* e se concatena os vetores de *features* para gerar o *vetor representação da carga* (z_{chrg})(Figura 1.19).

De forma similar à *feature* básica carga atômica parcial os valores referentes à *feature* básica distância também são discretizados antes de serem dados como entrada para a DeepVS. Para realizar a discretização se definem um valor mínimo dt_{min} e um valor máximo dt_{max} de distância possíveis, e se constroem intervalos de $0,3$ Å entre esses valores. Por exemplo, com $dt_{min} = 0$ e $dt_{max} = 5,1$ Å obtém-se 18

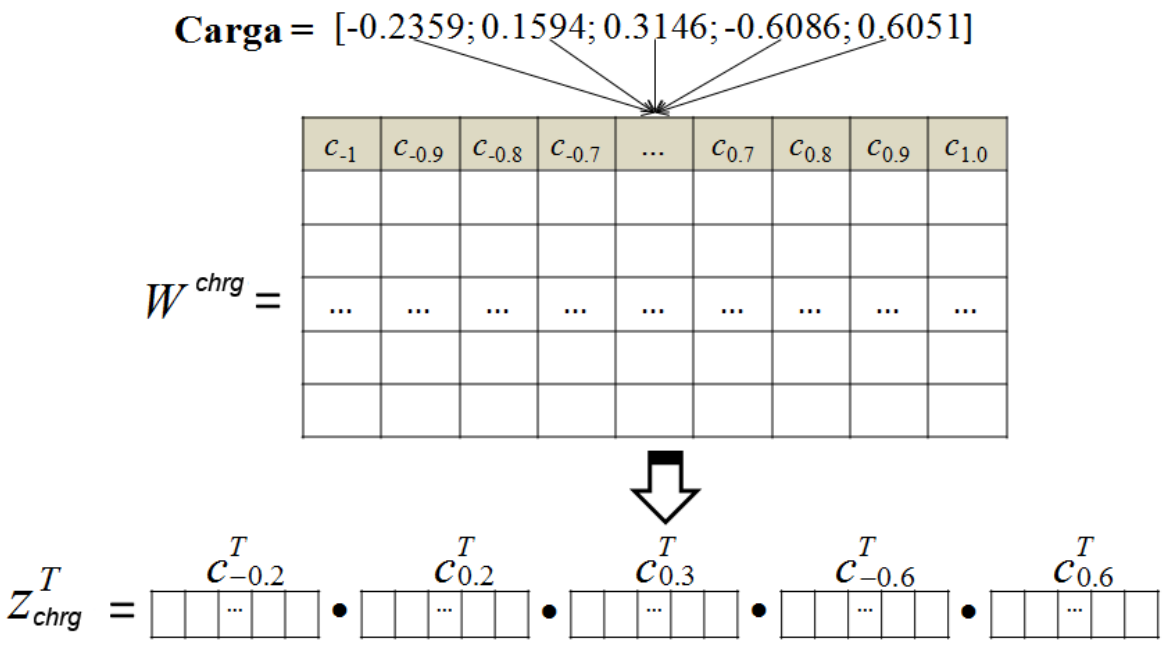


Figura 1.19: **Ilustração da construção do vetor representação da carga atômica parcial(z_{chrg}) usando a matriz de *embeddings* W^{chrg}** para o contexto do átomo N_3 advindo do ligante THM (Thymidina). O símbolo • indica uma operação de concatenação.

intervalos de distância possíveis. Cada valor da *feature* básica distância é mapeado para o índice do intervalo ao qual ele pertence. Na rede, a matriz $W^{dist} \in \mathbb{R}^{d^{dist} \times |D|}$ contém um vetor de *features* (coluna) para cada intervalo de distância t, onde D é o conjunto de intervalos de distância, d^{dist} é o número de *features* que são aprendidas pela rede e d^{dist} é um hiperparâmetro definido pelo usuário. Dado o contexto de um átomo a, cada valor da *feature* básica distância é transformado no seu respectivo vetor de *features* e esses vetores são concatenados para gerar o vetor representação da distância (z_{dist}) (Figura 1.20).

A *feature* básica tipo de aminoácido inclui o resíduo associado aos k_p átomos da proteína mais próximos do átomo "a" pertencente ao ligante. Cada coluna da matriz $W^{amino} \in \mathbb{R}^{d^{amino} \times |AM|}$ corresponde ao vetor de *features* de um tipo de resíduo presente na proteína, onde AM é o conjunto de tipos de resíduos e d^{amino} é o número de *features* que são aprendidas pela DeepVS. O procedimento para gerar a *feature* básica do tipo de aminoácido é análogo aos definidos anteriormente na seção 1.3.2.2 para as outras *features*. A Figura 1.21 ilustra a criação de z_{amino} no o contexto do átomo N_3.

Finalmente, a representação do contexto de um átomo "a" é definido como $z_a = \{z_{atm}; z_{dist}; z_{chrg}; z_{amino}\}$, que corresponde à concatenação dos vetores descritos previamente na seção 1.3.2.2. Nossa hipótese é que a partir das *features* básicas, a rede é capaz de aprender *features* mais abstratas (os *embeddings*) que são informativas com relação à discriminação entre ligantes e *decoys*. Esse tipo de estratégia, em que os valores das *features* básicas (palavras) são transformadas em vetores de *feature* mais abstratas (*embeddings* de palavras), é uma prática que tem obtido grande sucesso

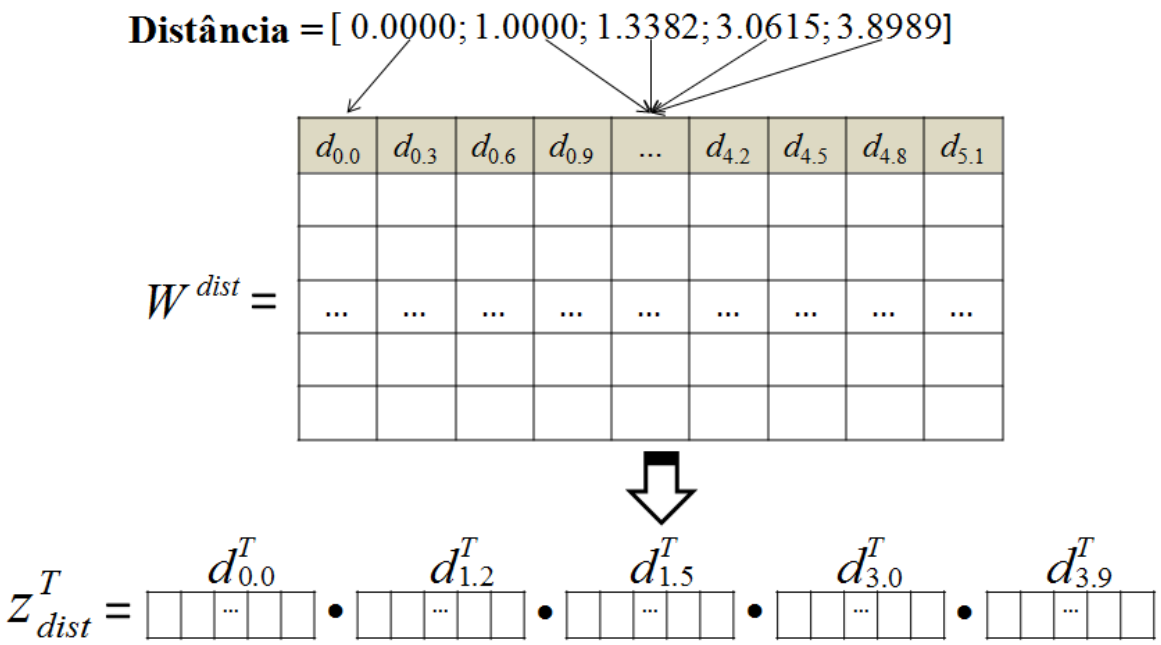


Figura 1.20: **Ilustração da construção da representação da distância entre o átomo e sua vizinhança (z_{dist}) usando a matriz de *embeddings* W^{dist}** para o contexto do átomo N_3 advindo do ligante THM (Thymidina). O símbolo • indica uma operação de concatenação.

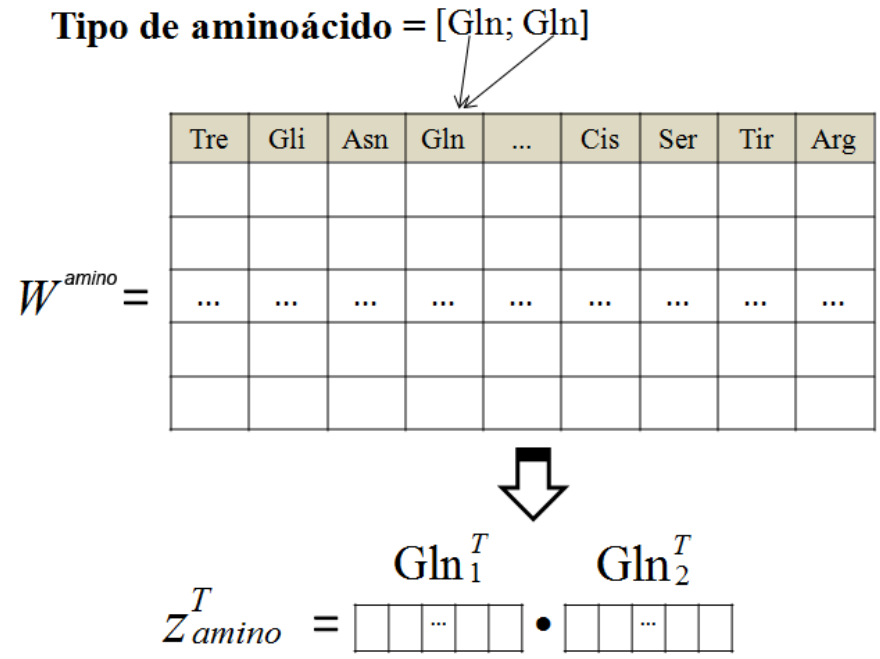


Figura 1.21: **Ilustração da construção da representação do tipo de resíduo associado a proteína (z_{amino}) usando a matriz de *embeddings* W^{amino}** para o contexto do átomo N_3 advindo do ligante THM (Thymidina). O símbolo • indica uma operação de concatenação.

para o campo de Processamento de Linguagem Natural (PLN) [98, 99, 110–112].

1.3.2.3 Representação do Complexo Proteína-Ligante

A segunda camada escondida da DeepVS consiste em uma camada convolucional, responsável por (1) extrair *features* mais abstratas a partir das representações dos contextos dos átomos do ligante, e (2) sumarizá-las em um vetor de tamanho fixo *r*. Nós nomeamos o vetor r como a *representação do complexo proteína-composto*. O vetor r é a saída da camada convolucional.

Uma das grandes motivações do uso de uma camada convolucional é a sua capacidade de lidar com entradas de tamanhos variáveis [90]. No caso da tarefa de *virtual screening*, a quantidade de átomos varia entre os compostos, logo, a quantidade de representações de contextos de átomos varia de um complexo para outro. Na DeepVS, a camada convolucional permite o processamento de complexos de diferentes tamanhos.

Dado um complexo x, cujo ligante possui m átomos, a entrada para a camada convolucional é um conjunto de vetores $\{z_1, z_2, \cdots, z_m\}$, onde z_i é a representação do contexto do i-ésimo átomo do ligante. A primeira etapa da camada convolucional consiste na geração de *features* mais abstratas a partir de cada vetor z_i usando a seguinte operação:

$$u_i = f(W^1 z_i + b^1) \tag{1.3}$$

onde, $W^1 \in \mathbb{R}^{cf \times |z_i|}$ é uma matriz de pesos da camada convolucional, b^1 é um viés (*bias term*), f é a função tangente hiperbólica e $u_i \in \mathbb{R}^{cf}$ corresponde ao vetor de *features* resultantes. O número de unidades (também conhecido como filtros) na camada convolucional, cf , é um hiperparâmetro a ser definido pelo usuário.

O segundo passo na camada convolucional, conhecido como *pooling layer*, sumariza as *features* provenientes dos vários contextos de átomo. A entrada para a camada *pooling* consiste em um conjunto de vetores $\{u_1, u_2, \cdots, u_m\}$. Para a DeepVS, nós usamos uma camada *pool* do tipo *max* (ou *max-pooling layer*), que produz um vetor $r \in \mathbb{R}^{cf}$, onde o valor do j-ésimo elemento é definido pelo valor máximo dentre os j-ésimos elementos dos vetores de entrada. Formalmente:

$$[r]_j = \max_{1 \leq i \leq m} [u_i]_j \tag{1.4}$$

O vetor resultante r para esse etapa é a representação do complexo proteína-composto (Eq. 1.4). Ao treinar a rede, o objetivo é que a mesma aprenda a gerar um vetor de *features* que sumarize a informação proveniente do complexo proteína-composto que é relevante para discriminar entre ligantes e *decoys*.

1.3.2.4 Pontuação do Complexo Proteína-Composto

O vetor r é processado por duas camadas da rede neural: uma terceira camada escondida que em princípio pode gerar *features* ainda mais abstratas, e uma camada de saída, que computa o *score* para as duas possíveis classificações do complexo: (0) ligante inativo e (1) ligante ativo. Formalmente, dada a representação r gerada pelo o complexo x, a terceira camada e a camada de saída executam a seguinte operação:

$$s(x) = W^3 \left(W^2 r + b^2\right) + b^3 \tag{1.5}$$

onde, $W^2 \in \mathbb{R}^{h \times |cf|}$ é a matriz de peso da terceira camada, $W^3 \in \mathbb{R}^{2 \times |h|}$ é a matriz de pesos da camada de saída, $b^2 \in \mathbb{R}^h$ e $b^3 \in \mathbb{R}^2$ são os seus respectivos vieses. O número de unidades na camada escondida h é um hiperparâmetro definido pelo usuário. $s(x) \in \mathbb{R}^2$ é um vetor que contém o score para cada uma das duas classes. Sejam $s(x)_0$ e $s(x)_1$ os scores para classe 0 e para classe 1, respectivamente. Transformamos essas pontuações em uma distribuição de probabilidades utilizando a função *softmax* que tem como objetivo definir a probabilidade de uma determinada classe dentro de um problema multiclasses:

$$p(0|x) = \frac{e^{s(x)_0}}{e^{s(x)_0} + e^{s(x)_1}} \tag{1.6}$$

$$p(1|x) = \frac{e^{s(x)_1}}{e^{s(x)_0} + e^{s(x)_1}} \tag{1.7}$$

As probabilidades $p(0|x)$ e $p(1|x)$ são interpretadas como probabilidades condicionais para o ligante ser considerado um *decoy* ou um ligante ativo, respectivamente, dado um complexo proteína-composto advindo previamente de um processo de *docking*.

A probabilidade dada para a classe 1 (ligante ativo) é o score usado para ranquear os ligantes durante os nossos experimentos de *virtual screening*. Quanto maior essa pontuação, maior a chance da molécula ser um ligante ativo.

1.3.2.5 Treinamento da DeepVS

Um abordagem comum para o treinamento de redes neurais é o algoritmo Gradiente Descendente Estocástico (*Stochastic Gradient Descent - SGD*) [81]. Em nosso caso, o SDG é usado para minimizar uma função de perda (função de custo) sobre um conjunto de treino *D* que contém exemplos de complexos com ligantes e *decoys* (moléculas com características semelhantes aos ligantes mas possivelmente inativos para o receptor examinado [113]). A cada iteração do algoritmo SGD, um novo complexo $(x, y) \in D$ é aleatoriamente escolhido, onde $y = 1$ se o complexo contém um

ligante ativo, ou $y = 0$, caso contrário. Em seguida, a rede DeepVS com o conjunto de parâmetros $\theta = \{W^{atm}, W^{chrg}, W^{dist}, W^{amino}, W^1, b^1, W^2, b^2, W^3, b^3\}$ é utilizada para gerar a probabilidade $p(y|x, \theta)$. Finalmente, o erro de predição é calculado usando a função de perda $-\log(p(y|x, \theta))$ e os parametros da rede são atualizados com o uso do algoritmo *Backpropagation* [81]. Formalmente, a rede é treinada usando-se SGD para selecionar um conjunto de valores de parâmetros θ que minimizam:

$$\theta \longmapsto \sum_{(x,y)\in D} -\log p(y|x, \theta) \tag{1.8}$$

Em nossos experimentos, nós aplicamos SGD com *minibatches*, o que significa que ao invés de considerarmos somente um complexo proteína-ligante em cada interação, consideramos um pequeno conjunto m_s de complexos selecionados aleatoriamente e usamos a média da predição de perda para executar o *Backpropagation*. Foi definido um conjunto de $m_s = 20$ para esse trabalho. Usamos a biblioteca *Theano* [114] para implementar a DeepVS e executar todos os experimentos reportados nesse trabalho. *Theano* é uma biblioteca *Python* que permite a compilação de funções matemáticas em linguagem *C. Theano* também permite o uso de múltiplos cores e GPUs de forma transparente para o usuário.

1.3.3 Configurações do Experimento

1.3.3.1 Conjunto de Dados

Nós usamos o conjunto de dados DUD (*Directory of Useful Decoys*) [64] como um conjunto de dados de referência para avaliar a abordagem de melhoramento de *virtual screening* baseado em *deep learning* proposta nesse trabalho. Um dos grandes motivos de usarmos o DUD como um conjunto de dados de referência está relacionado à capacidade de comparar os nossos resultados com os de outras abordagens de aprendizado de máquina propostas em trabalhos que utilizaram as mesmas condições de treino e teste [2, 14, 61].

O DUD é um conjunto dados construído para avaliar cálculos de *virtual screening* baseado em *docking*. É constituído de 40 receptores distribuídos em seis grupos biológicos distintos: receptores nucleares hormonais, quinases, serinoproteases, enzimas metalo, enzimas falato e outras classes de enzimas [64]. Possui 2.950 ligantes anotados e 95.316 *decoys* na proporção de 36 decoys para cada ligante anotado. Cada um dos 36 *decoys* foi retirado do banco de dados ZINC [115] de modo a serem semelhantes em alguma propriedade física do ligante anotado, como: peso molecular, cLogP ou em número de grupos de ligação de hidrogênio, mas diferem em sua topologia [2, 64].

A versão original do DUD possui um problema relacionado às distribuições de

cargas atômicas parciais descritas no conjunto de dados que facilita a discriminação entre ligantes e *decoys*. A média das cargas entre ligantes e *decoys* é bem distinta para alguns receptores. Por exemplo, para acetilcolinesterase a média das cargas dos ligantes é -1,68 enquanto a média das cargas dos *decoys* é 0,76, nesse caso o ranqueamento perfeitos dos ligantes e *decoys* é possível usando apenas a informação da carga atômica parcial dos compostos. Por esse motivo, usamos nos nossos experimentos a versão do DUD proposta por Armstrong e colaboradores [116] no qual todas as cargas atômicas parciais foram corrigidas.

Um outro problema relacionado ao DUD é a presença de duplicatas ou triplicatas de ligantes e *decoys*, que são repetições de um mesmo composto variando apenas em sua estereoquímica, ou seja, na sua disposição de átomos no espaço (Tabela 1.2). Segundo os autores do DUD e do DUD-E [117] é possível encontrar em ambos os bancos de dados compostos estereoquímicamente diferentes associados a uma mesma ID. Um exemplo disto consiste no ligante da proteína TK (ID_PDB [42]: 1kim) que possui mais de um representante com mesma ID ZINC04225128 (Figura 1.22). Segundo os autores [117] essas repetições no conjunto de dados estão associadas a um problema no banco de dados ZINC [53] que retorna mais de uma versão do estado de "referência" para ligantes e *decoys*. O estado de "referência" é o melhor representante único de um composto em pH 7,5 (como melhor representante único em pH 7,5 a molécula deve apresentar-se em um estado de protonação único). Para realizar os nossos experimentos selecionamos um representante único para cada ID, levando em consideração o maior valor de *score* proveniente dos programas de *docking* Autodock-Vina1.1.2 [70] e Dock6.6 [118].

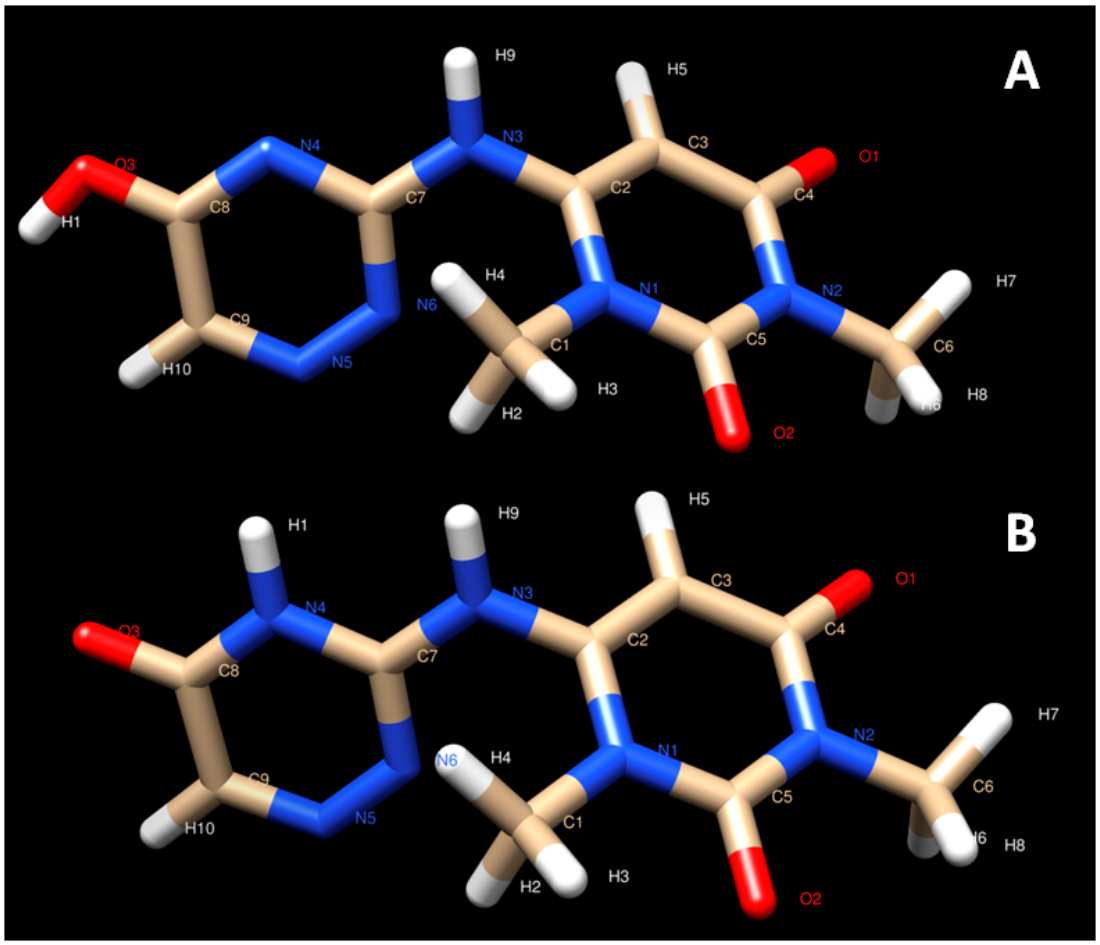


Figura 1.22: **Dois estados de protonação diferentes para mesmo ligante ID ZINC04225128 associado à proteína TK (ID_PDB: 1kim)**. Em A no ZINC04225128 o hidrogênio H1 se liga covalentemente ao oxigênio O3. Em B no ZINC04225128 o hidrogênio H1 realiza ligação covalente com o nitrogênio N4.

Tabela 1.2: **Repetições encontradas no DUD**. Valores para o banco de dados DUD do número de ligantes e *decoys*; número de ligantes e *decoys repetidos*; número total de ligantes e *decoys*; porcentagem de repetidos (ligantes e *decoys*) no conjunto de dados.

Repetições no banco de dados DUD										
ID_DUD	ID_PDB	classe biológica	Nº ligantes	Nº repetidos	total ligantes	% repetidos	Nº *decoys*	Nº repetidos	total *decoys*	% repetidos
ACE	1o86	Metalo enzimas	49	0	49	0	1.728	69	1.797	3,84
AChE	1eve	Outras enzimas	105	2	107	1,87	3.732	160	3.892	4,11
ADA	1ndw	Metalo enzimas	23	16	39	41,03	822	105	927	11,33
ALR2	1ah3	Outras enzimas	26	0	26	0	920	75	995	7,54
AmpC	1xgj	Outras enzimas	21	0	21	0	734	52	786	6,62
AR	2ao6	Nucleares hormonais	74	5	79	6,33	2.630	224	2.854	7,85
CDK2	1ckp	Quinases	50	12	72	16,66	1.780	294	2.074	14,17
COMT	1h1d	Metalo enzimas	12	0	12	0	430	38	468	8,12
COX1	1p4q	Outras enzimas	25	0	25	0	850	61	911	6,70
COX2	1cx2	Outras enzimas	349	77	426	18,08	12.491	798	13.289	6,00
DHFR	3dfr	Folato enzimas	201	209	410	50,98	7.150	1.217	8.367	14,55
EGFr	1m17	Quinases	416	59	475	14,42	14.914	1.082	15.996	6,76
$ER_{agonist}$	1l2i	Nucleares hormonais	67	0	67	0	2.361	209	2.570	8,13
$ER_{antagonist}$	3ert	Nucleares hormonais	39	0	39	0	1.399	49	1.448	3,38
FGFrl	1agw	Quinases	118	2	120	1,66	4.216	334	4.550	7,34
FXa	1f0r	Serina proteases	142	4	146	2,74	5.102	643	5.745	11,19
GART	1c2t	Folato enzimas	21	19	40	47,50	753	126	879	14,33
GPB	1a8i	Outras enzimas	52	0	52	0	1.851	289	2.140	13,50
GR	1m2z	Nucleares hormonais	78	0	78	0	2.804	143	2.947	4,8
HIVPR	1hpx	Outras enzimas	53	9	62	14,52	1.888	150	2.038	7,36
HIVRT	1rt1	Outras enzimas	40	3	43	6,98	1.439	80	1.519	5,27
HMGR	1hw8	Outras enzimas	35	0	35	0	1.242	238	1.480	16,08
HSP90	1uy6	Quinases	39	0	39	0	1.399	49	1.448	3,38
InhA	1p44	Outras enzimas	85	1	86	1,16	3.043	3.489	6.532	53,41
MR	2aa2	Nucleares hormonais	15	0	15	0	535	101	636	15,88
NA	1a4g	Outras enzimas	49	0	49	0	1.745	129	1.874	6,88
P38MAP	1kv2	Quinases	234	220	454	48,46	8.399	742	9.141	8,12
PARP	1efy	Outras enzimas	33	2	35	5,71	1.178	173	1.351	12,81
PDE5	1xp0	Metalo enzimas	51	37	88	42,05	1.810	168	1.978	8,49
PDGFrb	modelo	Quinases	157	13	170	7,65	5.625	355	5.980	5,94
PNP	1b80	Outras enzimas	25	25	50	50	884	152	1.036	14,67
PPARg	1fm9	Nucleares hormonais	81	4	85	4,71	2.910	355	3.127	11,35
PR	1sr7	Nucleares hormonais	27	0	27	0	967	74	1.041	7,11
RXRa	1mvc	Nucleares hormonais	20	0	20	0	708	42	750	5,6
SAHH	1a7a	Outras enzimas	33	0	33	0	1.159	187	1.346	13,89
SRC	2src	Quinases	162	3	165	1,82	5.801	513	6.314	8,12
thrombin	1ba8	Serina proteases	65	7	72	9,72	2.294	162	2.456	6,60
TK	1kim	Quinases	22	0	22	0	785	106	891	11,90
trypsin	1bju	Serina proteases	43	6	49	12,24	1.545	119	1.664	7,15
VEGFr2	1vr2	Quinases	74	4	88	4,55	2.647	259	2.906	8,91

1.3.3.2 Conjunto de Dados para Validação Externa

Para verificar a robustez da nossa abordagem selecionamos de forma aleatória um subconjunto do banco de dados *Directory of Useful Decoys: Enhanced* (DUD-E) [119]. O subconjunto selecionado é composto por oito receptores e seus respectivos ligantes e *decoys*. Cada receptor do DUD-E selecionado pertence a uma classe biológica distinta e não possui relação com nenhum dos receptores existentes no DUD [64]. O subconjunto selecionado foi utilizado para realizar uma validação externa da DeepVs, que compreende o treinamento da DeepVS usando o banco de dados DUD (40 receptores) e aplicação da rede treinada no subconjunto do DUD-E (8 receptores).

O banco de dados DUD-E é uma versão melhorada do banco de dados DUD que inclui novas classes biológicas como GPCRs e canais iônicos. As alterações encontradas no DUD-E incluem a redução do viés de quimiotipos e uma expansão no número de receptores de 40 para 102. Os autores do banco de dados DUD-E empregaram uma extensiva revisão de todas as cargas atômicas dos compostos e receptores que possuem um maior número de ligações com o ligante foram favorecidos.

O número de ligantes foi ampliado para 224 ligantes anotados por receptor totalizando um número de 22.886 compostos ativos identificados incrementando a diversidade de ligantes para um determinado alvo. Os receptores são divididos em oito classes biológicas distintas (Figura 1.23): quinases, proteases , receptores nucleares, CGPR, canais iônicos, citocromo P450, proteínas diversas e outras enzimas. O número de *decoys* é de 50 para cada ligante anotado. O número de falsos *decoys* (falsos *decoys* são *decoys* que possuem atividade comprovada) foi reduzido e o processo de geração de *decoys* foi extensivamente revisado.

Apesar de todos os esforços realizados para corrigir os erros encontrados no banco de dados DUD em sua nova versão o DUD-E, alguns problemas como as repetições de compostos (ligantes e *decoys*) ainda são persistentes (seção 1.3.3.1). Nesse caso, seguimos os mesmo protocolo usado para o banco de dados DUD, que consiste em selecionar um representante único para composto repetido, levando em consideração o *score* proveniente dos programas de *docking* AutodockVina1.1.2 e Dock6.6.

1.3.3.3 Programas de *Docking*

Com intuito de testar a robustez do método com relação à variabilidade dos resultados de *docking*, usamos nesse trabalho dois programas para *docking* molecular: Dock 6.6 [118] e AutodockVina1.1.2 [70]. Ambos são programas de livre acesso e amplamente utilizados em *virtual screening*. Dock6.6 oferece função de pontuação baseada em pontuação da energia e campo de força (Grid Score & Amber Score) [118,120].

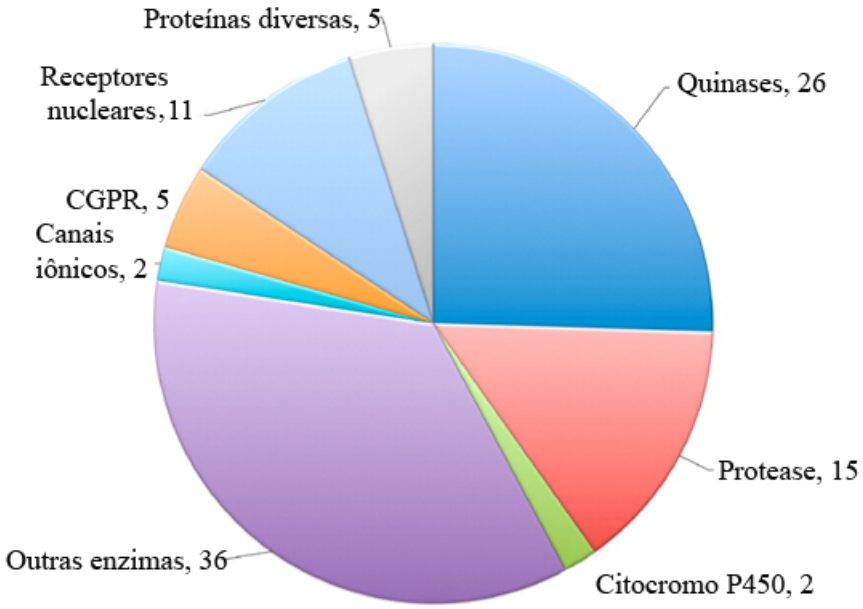


Figura 1.23: **Classificação dos receptores presentes no DUDE**. O número de 102 receptores distribuídos em oito classes biológicas. Fonte: Mysinger *et al.*, 2012 adaptado por Pereira, J.C.

AutodockVina1.1.2 fornece uma função de pontuação híbrida que combina características de funções baseadas no conhecimento e funções de pontuação empíricas (Vina score) [70].

1.3.3.4 Parâmetros Usados para *Docking* Usando Dock6.6

As estruturas dos receptores foram preparadas usando a ferramenta Dock Prep fornecida pelo o programa *Chimera* [121]. Dock Prep [122] consiste em uma interface gráfica que executa importantes funções como: remoção de moléculas de água, reparação de cadeias laterais truncadas, adiciona hidrogênios, atribui cargas atômicas parciais e escreve arquivos no formato **mol2**. Durante a preparação de todas as estruturas dos receptores as moléculas de água, ligantes e íons foram removidos e átomos de hidrogênio foram adicionados em pH fisiológico [122]. As cargas atômicas parciais foram calculadas usando o software de uso não comercial Open Babel [123] e o método usado como referência foi o *Gasteiger* [124]. Posteriormente os receptores foram salvos em formato **mol2**. Todos os compostos foram retirados diretamente da versão do DUD proposta por Armstrong e colaboradores [116] com as cargas atômicas parciais corrigidas usando o método *Gasteiger* [124].

A preparação do sítio de ligação dos receptores envolveu o uso para cada receptor de seu respectivo ligante cristalográfico. No cálculo da superfície acessível ao solvente para cada receptor, sem hidrogênios, foram gerados arquivos no formato **dms** usando a ferramente DMS pertencente ao programa *Chimera* [121]. A superfície molecular de ponto ou DMS consiste em duas partes: superfície de contato e superfí-

cie reentrante. O vetor normal é calculado em cada ponto das superfícies [125]. Para cada receptor, o arquivo em formato **dms** foi dado como entrada para ao programa SPHGEN [126] para gerar as esferas. O SPHGEN está disponível na versão padrão do programa de *docking* Dock. Trata-se de um conjunto de esferas sobrepostas para descrever a forma de uma molécula ou superfície molecular [126]. As esferas são calculadas sobre toda a superfície do receptor produzindo aproximadamente uma esfera por ponto de superfície (Figura 1.24). Subsequentemente, é realizada uma filtragem para manter apenas a maior esfera associada a cada átomo da superfície. O conjunto de esferas filtradas é então agrupado usando um único algoritmo de ligação, sendo que cada *cluster* resultante representa um possível sítio de ligação do receptor (Figura 1.25). A seleção de um único *cluster* de esferas foi realizada usando o programa *sphere_selector* fornecido pelo o Dock e para as coordenadas do ligante cristalográfico de cada receptor, as esferas selecionadas estão localizadas dentro de um raio que pode variar de 5-13.5 Å, calculado usado as coordenadas espaciais de cada ligante [127].

Com intuito de aumentar a velocidade da performance de *docking* o programa Dock6.6 usa uma *grid*. Antes de gerar a *grid* é necessário definir uma *box* que será utilizada para delimitar o local e o tamanho da *grid* (Figura 1.26). Para todos 40 receptores todas as informações referentes as coordenadas da *box* foram retiradas diretamente do banco de dado DUD [64]. No entanto, é possível gerar a *box* de forma manual usando o programa *showbox* disponível no Dock. A *grid* foi computada usado o programa Grid pertencente ao Dock e em todos os casos foram usadas as configurações padrão do Dock que consistem em: espaçamento da *grid* de 0,3 Å , uma distância de corte de 9.999, expoentes de Lennard-Jones 12-6 (atrativos-repulsivos) e uma constante dielétrica dependente da distância $4r$.

O programa de *docking* Dock possui duas formas para avaliar a energia potencial na região a ser docada: contato e energia *score*. O *score* de contato é uma soma dos contatos dos átomos pesados (exceto hidrogênio) entre o composto e o receptor. Um contato é definido como uma aproximação de dois átomos entre uma distância de corte (padrão: 4,5 Å). Se os dois átomos se aproximam o suficiente para colidir, essa interação pode ser penalizada. O *score* de energia é baseado em campo de força. Scores baseadas em campo de força são aproximações das energias de interação mecânico moleculares e é constituído de componentes eletrostáticos e Lennard-Jones:

$$E = \sum_{i=1}^{lig} \sum_{j=1}^{rec} \left(\frac{A_{ij}}{r_{ij}^{a}} - \frac{B_{ij}}{r_{ij}^{b}} + 332 \frac{q_i q_j}{D r_{ij}} \right) \tag{1.9}$$

onde cada termo é uma soma dupla sobre os átomos do ligante i e os átomos do receptor j. Nesse trabalho foi levado em consideração apenas o *score* energia. A *bump grid* foi também computada usando o programa Grid com uma margem de sobreposição de van der Waals de 0,75. A *bump grid* é um filtro para evitar colisões, no qual

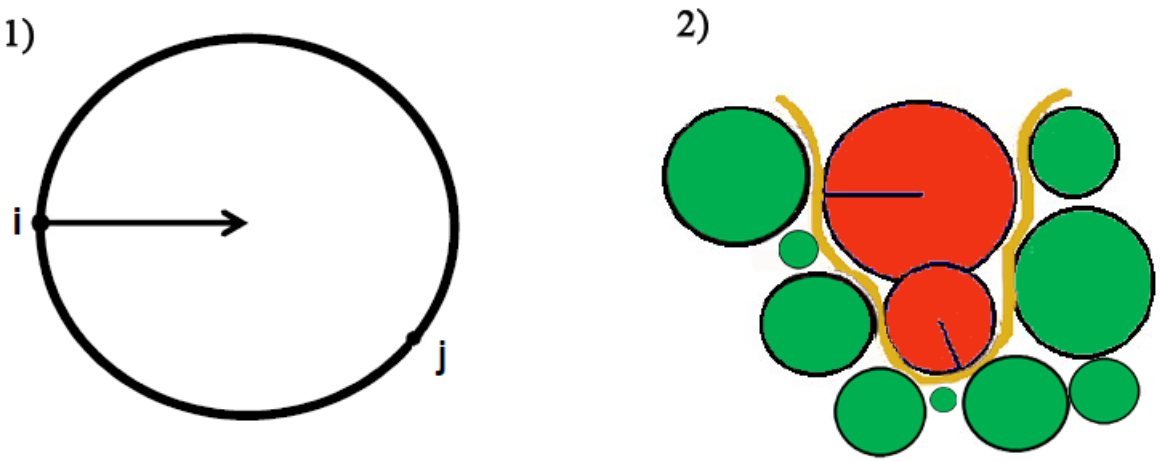


Figura 1.24: **Ilustração da geração das esferas**. 1) Cada esfera é criada de forma a tangenciar os pontos *i* e *j* da superfície, com o centro da esfera localizado na superficial normal ao ponto *i*. 2) Representação esquemática de um pequeno sítio de ligação formado por nove átomos (verde). As esferas (vermelho) são geradas usando os pontos da superfície molecular (amarelo) com seus centros situados ao longo da superfície normal (linha). Fonte: $http://dock.compbio.ucsf.edu/DOCK_6/tutorials/sphere_generation/generating_spheres.htm$ adaptado por Pereira, J.C.

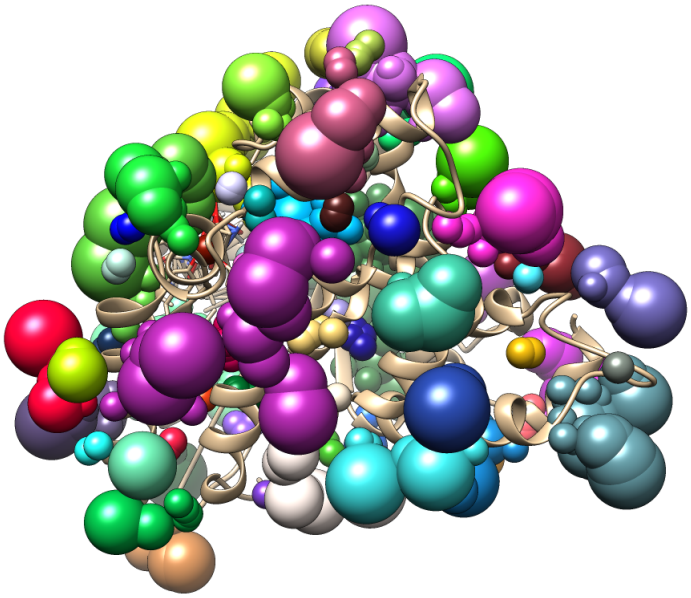


Figura 1.25: **Ilustração dos *clusters* de esferas**. Cada cor representa um *cluster* de esferas criado para o receptor TK (ID_PDB: 1kim) usando o programa SPHGEN fornecido pelo programa de *docking* Dock6.6.

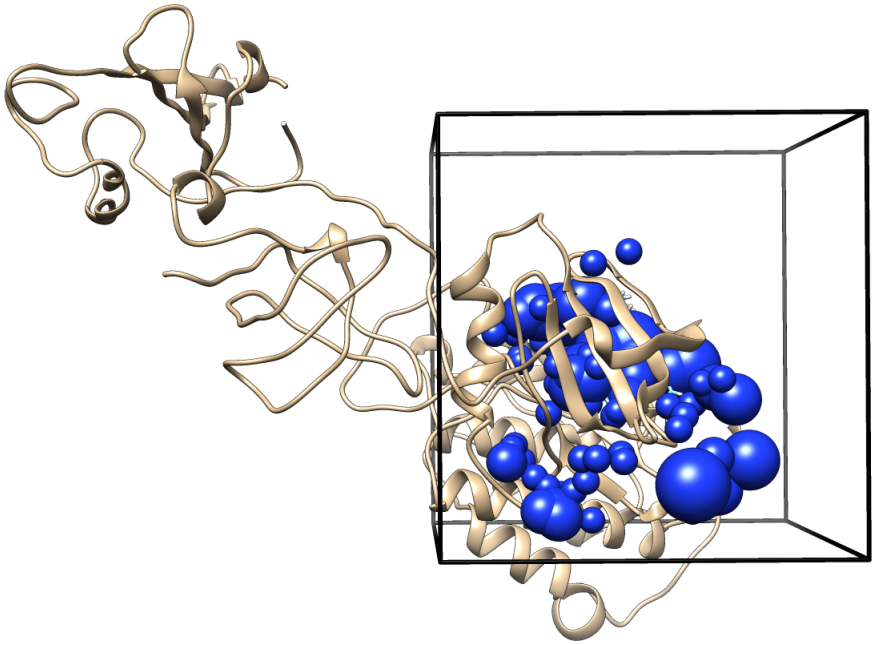


Figura 1.26: **Ilustração da *box* usada para construir a *grid***. Em azul as esferas criadas para o receptor SRC (ID_PDB: 2src), em preto a *box* retirada diretamente do DUD.

cada orientação pode ser processada de modo a evitar que átomos acoplem profundamente no interior do receptor. As orientações de átomos que passam pelo filtro de colisão são então pontuadas e/ou minimizadas usando qualquer uma das funções de pontuação disponíveis.

Rodadas de *docking* flexível foram efetuadas usando o programa Dock6.6 para todos receptores e seus respectivos ligantes e *decoys*. O programa Dock6.6 usa o algoritmo *anchor-and-grow* uma abordagem de construção incremental desenvolvida por Ewing e colaboradores [128] que consiste em fixar no sítio de ligação a maior subestrutura rígida do ligante (nomeado como âncora) e subsequentemente, cada parte flexível do ligante é construída de modo a formar um *cluster* que será minimizado, classificado e agrupado (nomeado como crescimento). Esse processo é repetido de forma a reconstruir toda a molécula. O número máximo de orientações para cada ligante foi definido como o padrão disponível no Dock6.6 que é 500 e foi escolhida apenas a melhor pose retornada para cada ligante, a pose que possui menor energia (Apêndice B). O programa de *docking* Dock6.6 possui duas funções de pontuação: a Grid Score, usada para definir a pose do composto e Amber Score utilizada como um método de reclassificação de compostos. Foi utilizada nesse trabalho apenas a função de pontuação Grid Score.

1.3.3.5 Parâmetros Usados para *Docking* Usando Autodockvina1.2

Os parâmetros utilizados nesse trabalho para o programa de *docking* Autodockvina1.1.2 [70] foram semelhantes aos parâmetros usados por outras abordagens re-

portadas na literatura que também empregaram redes neurais como a DDFA [2] e a NNScore [14, 61, 67]. Adotamos esse procedimento com o objetivo de estarmos em condições de comparar os nossos resultados com os resultados de outras abordagens de DBVS (*Docking-Based Virtual Screening*).

Os 40 receptores foram preparados usando o script *prepare_receptor4.py* disponível no programa MGLTools em específico no modulo AutoDockTools4 [74]. A ferramenta AutoDockTools4 foi especificamente desenhada para gerar entrada para os programas de *docking* AutoDock4 [74] ou AutoDockVina [70] que são arquivos em formato **pdbqt**, e pode também ser usada para visualizar os resultados de *docking* fornecidos por esses programas. Arquivos em formato **pdbqt** contêm as coordenadas atômicas dos átomos, cargas atômicas parciais e tipo do átomo reconhecido pelo AutoDock. Os átomos de hidrogênio foram adicionados nos casos em que eles não estavam presentes, usando o comando $[-A]$ "$checkhydrogens$" e as cargas atômicas parciais foram adicionadas segundo o método *Gasteiger*.

Todos os compostos foram retirados diretamente da versão do DUD proposta por Armstrong e colaboradores [116]. Os compostos foram preparados usando o script *prepare_ligand4.py* disponível no modulo AutoDockTools4 [74], todos os átomos de hidrogênios foram adicionados usando o parâmetro $[-A]$, cargas atômicas parciais foram adicionadas seguindo o método *Gasteiger* e todos os compostos foram escritos em formato **pdbqt**.

A *grid* foi definida com dimensões x, y, z igual a 27 Å. Os dados para a construção da *grid box* para cada receptor foram diretamente retirados do DUD, esses dados representam o centro de massa do ligante cristalográfico (Figura 1.27). No entanto, a *grid* pode ser definida manualmente usando a interface gráfica do modulo AutoDockTools.

Rodadas de *docking* foram efetuadas para os 40 receptores do DUD e seus receptivos ligantes e *decoys* utilizando o programa de *docking* Autodockvina1.1.2 [70]. Usamos como configuração dos parâmetros para o *docking* os seguintes valores: diferença de energia máxima entre a melhor ligação (*energy_range*) igual a 10; o número máximo de modos ligação gerados (*num_modes*) igual a 1, no qual compreende a melhor pose gerada pelo programa; a quantidade de CPUs variou de acordo com a disponibilidade de máquinas; tempo gasto aproximadamente para efetuar a pesquisa global (*exhaustiveness*) igual 16; e o valor do parâmetro semente (*seed*) correspondente a -16807. Os valores seguem as configurações padrão do Autodockvina1.1.2 com exceção do parâmetro *exhaustiveness* que foi alterado segundo a configuração usada por Arciniega & Lange [2] (Apêndice C).

O programa Autodockvina1.1.2 é um programa de livre acesso e código aberto amplamente utilizado para performance de *docking* e *virtual screening*. Sua função de pontuação combina estrategias baseadas no conhecimento e em funções de pontuação empíricas. A Vina Score foi inspirada na X-Score [129], uma função de pontuação empírica desenvolvida para estimar a afinidade de ligação dado um complexo proteína-composto. A função de pontuação leva em consideração os termos: inte-

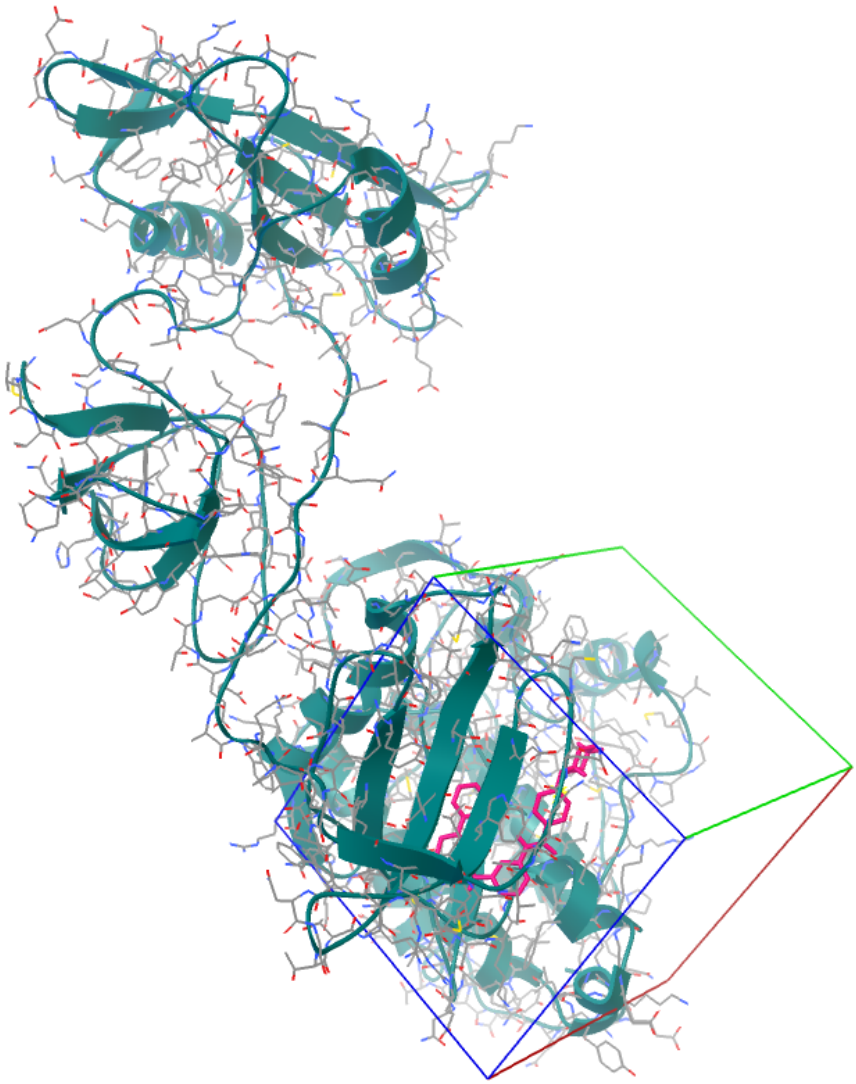


Figura 1.27: **Ilustração da *grid box* para o receptor SRC (ID_PDB: 2src)**. A *grid box* é representada em suas dimensões x, y e z nas cores vermelho, verde e azul respectivamente. O ligante ANP é representado em rosa.

rações estéricas (representadas pelos termos $gauss_1$, $gauss_2$ e repulsão), hidrofobicidade, ligações de hidrogênio e número ligações rotacionáveis entre átomos pesados do ligante (termo N_{rot}) [70]. Os pesos para cada termo utilizado pela função de pontuação está descrito na Tabela 1.3. O Autodockvina1.1.2 levou em consideração várias estratégias de otimização global estocástica e definiu o algoritmo *Iterated Local Search* [130, 131] como o algoritmo de otimização global.

Scripts implementados na linguagem de programação *python* foram utilizados para automatizar todas as etapas descritas acima como: preparação dos receptores, preparação dos ligantes e *decoys*, construção da *grid*, criação do arquivo de configuração e rodadas de *docking*.

Tabela 1.3: Pesos e termos usados pela a função de pontuação do Autodockvina1.1.2. Fonte: (Trott *et al*., 2010).

Termo	Peso
$gauss_1$	$-0,0356$
$gauss_2$	$-0,00516$
repulsão	$0,840$
hidrofobicidade	$-0,0351$
ligações de hidrogênio	$-0,587$
N_{rot}	$0,0585$

1.3.3.6 Abordagem Experimental

O desempenho do método proposto é avaliado usando a abordagem de validação cruzada *leave-one-out* com 40 proteínas pertencentes ao conjunto de dados DUD (seção 1.3.3.1). A figura 1.28 representa o processo que seguimos para realizar os nossos experimentos de validação cruzada *leave-one-out*. Essa abordagem consiste em realizar várias rodadas de treino e teste, sendo que em cada rodada um receptor é separado como conjunto de teste e os demais receptores são utilizados como conjunto de treino. A entrada para o processo de validação cruzada são os resultados individuais dos programas de *docking* aplicados aos 40 receptores disponíveis no DUD.

Para evitar distorções no resultado do desempenho da DeepVS é importante remover do treinamento os receptores que são similares ao receptor usado como teste em uma rodada específica de validação cruzada. De acordo com Arciniega & Lange (2014) [2], consideramos receptores similares aqueles que compartilham uma mesma classe biológica ou que possuem enriquecimento cruzado (*cross-enrichment*) positivo reportado. O enriquecimento cruzado ocorre quando um conjunto de ligantes demonstra enriquecimento positivo para mais de um receptor em etapas de *cross-docking* [64].

Uma vez que a rede foi treinada, subsequentemente ela foi aplicada ao receptor de teste, produzindo como resultado uma pontuação para cada um dos potenciais ligantes. Tal pontuação foi usada para ranquear os ligantes. O ranqueamento foi avaliado usando métricas bem estabelecidas como área sob a curva ROC (AUC) e fator de enriquecimento (ef) que indicam a performance do algoritmo (seção 1.3.4).

1.3.3.7 Hiperparâmetros da DeepVS

A principal vantagem no uso da validação cruzada *leave-one-out* é a possibilidade de ajustar os hiperparâmetros da rede neural sem se preocupar com *overfitting*. O *overfitting* acontece quando os parâmetros estão otimizados para obter um bom resultado no conjunto de treino, mas o modelo gerado não possui bom resultado no conjunto de teste. De fato, a validação cruzada *"leave-one-out"* é um método ade-

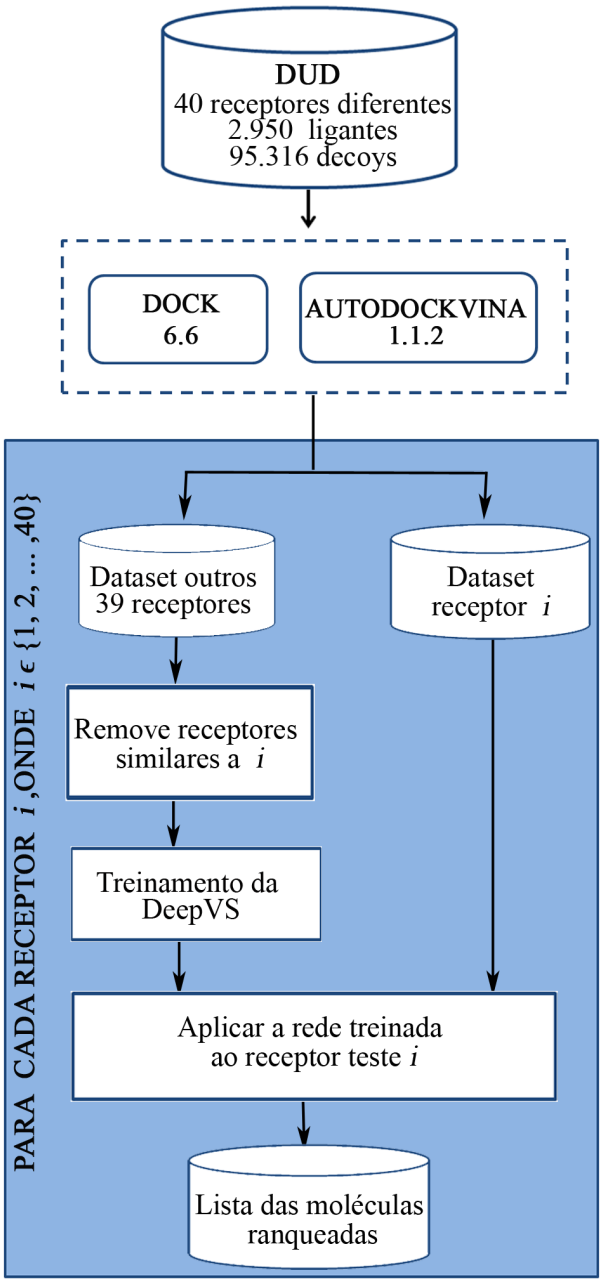


Figura 1.28: **Ilustração do processo de treinamento da DeepVS que utiliza validação cruzada do tipo *leave-one-out*.**

quado para ajustar hiperparâmetros de algoritmos de aprendizado de máquina quando o conjunto de dados é pequeno [132]. Em nossos experimentos, usamos o mesmo conjunto de hiperparâmetros para as 40 iterações de validação cruzada *leave-one-out*, o que equivale a executar 40 experimentos diferentes com diferentes conjuntos de treino/teste usando a mesma configuração para a DeepVS.

Os valores de hiperparâmetros que forneceram os melhores resultados e que foram usados em nos nossos experimentos para a saída tanto do AutodockVina1.1.2 quanto para Dock6.6, são especificados na Tabela 1.4. Note-se que, a nossa abordagem de avaliação foi mais rigorosa do que a utilizada por Arciniega & Lange (2014) [2], porque eles ajustaram os hiperparâmetros usando um conjunto *hold-out* em cada iteração do *leave-one-out*.

Tabela 1.4: Valores de hiperparâmetros para DeepVS usados durante o treinamento da rede neural

Hiperparâmetro	Descrição	Valor
d^{atm}	tamanho do *embedding* tipo de átomo	200
d^{amino}	tamanho do *embedding* tipo de aminoácido	200
d^{chrg}	tamanho do *embedding* carga	200
d^{dist}	tamanho do *embedding* distância	200
cf	# filtros da camada convolucional	400
h	# unidades da camada escondida	50
λ	taxa de aprendizado	0,075
k_c	# número de vizinhos no ligante	6
k_p	# número de vizinhos na proteína	2

1.3.4 Métricas de Avaliação

Para validar a performance da DeepVS e compara-lá com outros métodos previamente descritos na literatura, foram utilizadas duas métricas de avaliação bem estabelecidas em *virtual screening*: o fator de enriquecimento (*Enrichment Factor*-EF) e a área sob a curva ROC (*Receiver Operating Characteristic*) [133, 134]. Curvas ROC são uma forma de representar a relação entre a sensibilidade (Se) e a especificidade (Sp) ao longo de uma extensão de valores contínuos (Equações 1.10 e 1.11), que representam a taxa de verdadeiros positivos em função de falsos positivos.

$$Se = \frac{verdadeiro\ positivos}{total\ ativos} \tag{1.10}$$

$$Sp = \frac{verdadeiro\ negativos}{total\ decoys} \tag{1.11}$$

A área sob a curva ROC (AUC) representa uma quantificação da curva e faci-

lita a comparação dos resultados. A AUC é calculada dada a equação 1.12, onde N_{ativos} representa o número de ligantes, N_{decoys} representa o número de *decoys*, e $N^{i}_{decoys\_verificados}$ representa o número de *decoys* que são mais bem classificados do que i-ésima estrutura ativa [133]. Uma AUC = 0,50 indica uma seleção aleatória, enquanto que AUC igual a 1 indica uma perfeita identificação de compostos ativos (Figura 1.29).

$$AUC = 1 - \frac{1}{N_{ativos}} \sum_{i}^{N_{ativos}} \frac{N^{i}_{decoys\_verificados}}{N_{decoys}} \tag{1.12}$$

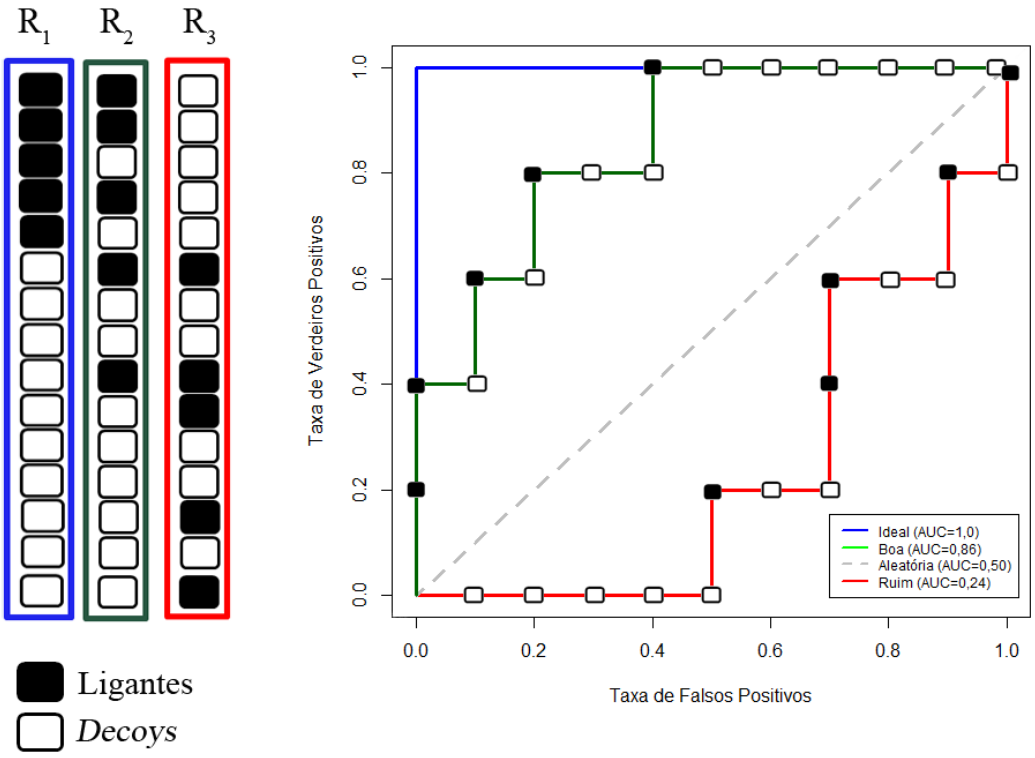


Figura 1.29: **Geração de curvas ROC**. Nessa figura usamos uma lista de 15 compostos para ilustrar a geração de três curvas ROC. Os compostos estão divididos em cinco ligantes (retângulos em preto) e 10 *decoys* (retângulos em branco). Três *rankings* com qualidades diferentes são apresentados: R_1, *ranking* ideal com AUC = 1,0; R_2, um bom *ranking* com AUC = 0,86; R_3, um *ranking* ruim com AUC = 0,24. A linha tracejada representa um *ranking* aleatório com AUC = 0,50.

Dado um conjunto de compostos ranqueados pela sua pontuação, o fator de enriquecimento em x% (Eq. 1.13) informa o quão bom é o conjunto formado pelo os top x% compostos ranqueados em comparação com um conjunto de tamanho igual selecionado de forma aleatória a partir do conjunto completo de compostos. O EF é calculado como:

$$EF_{x\%} = \frac{ativos\ de\ X\%}{compostos\ de\ X\%} \frac{total\ compostos}{total\ ativos} \tag{1.13}$$

onde, $ativos\ de\ X\%$ representa a quantidade de ligantes selecionados em uma porcentagem X do conjunto de dados, $compostos\ de\ X\%$ o número de compostos (ligantes e *decoys*) selecionados em uma porcentagem X do conjunto de dados, $total\ compostos$ número total de compostos (ligantes e *decoys*) e $total\ ativos$ número total de ligantes presentes no conjunto de dados.

1.4 Resultados e Discussão

Na seção 1.4.1, comparamos a performance de três métodos de DBVS: DeepVS (um novo método para DBVS objeto de estudo do presente trabalho) e dois programas de *docking* Dock6.6 e Autodockvina1.1.2 para o conjunto de dados DUD e discutimos a utilidade da DeepVS como um método para o melhoramento de DBVS. Na seção 1.4.2, baseado em nossos resultados, discutimos sobre a melhor abordagem de DBVS a ser adotada dado um sistema (receptor) especifico. A seção 1.4.3 apresenta resultados da sensibilidade da DeepVS com relação aos seus hiperparâmetros ($d^{atm/amino/chrg/dist}$, cf, λ, k_c e k_p), bem como resultados relacionados a sensibilidade à variação da semente de inicialização de números aleatórios. Na seção 1.4.4 discutimos quando a informação da proteína é importante para rede distinguir entre ligantes e *decoys*. A seção 1.4.5 apresenta uma comparação entre a performance da DeepVS e de outras abordagens estado-da-arte em DBVS.

1.4.1 DeepVS vs Programas de *Docking*

A rede neural DeepVS foi testada em um conjunto de mais de 100 mil complexos proteína-composto distribuídos entre 40 proteínas de seis classes biológicas distintas: nucleares hormonais, quinases, serinoproteases, metalo enzimas, folato enzimas e outras enzimas.

Utilizamos o banco de dados *Directory of Useful Decoys* - DUD que é considerado o estado-da-arte em banco de dados desenvolvidos para avaliar o desempenho de ferramentas usadas para efetuar DBVS [20, 135]. Uma das principais características desse banco de dados é que os *decoys* foram selecionados de modo a serem similares aos ligantes em suas características fisioquímicas (peso molecular; número de doadores e aceptores de ligações de hidrogênio; logP; e número de ligações rotacionais) e simultaneamente diferirem em sua topologia. Para minimizar a similaridade topológica entre ligantes e *decoys* os autores do DUD [64] utilizaram o método 2D *fingerprints*. Segundo Hawkins *et al.,* [20] da forma como esse banco de dados foi desenhado, é provável que distinguir entre *decoys* e ligantes verdadeiros no DUD é mais difícil do que fazer essa distinção em banco de dados de moléculas como o ZINC [53]. Essas características sugerem que o DUD é útil para testar de forma robusta métodos de melhoramento de DBVS como a DeepVS. Outra vantagem de utilizarmos o DUD consiste em podermos comparar os nossos resultados com trabalhos previamente publicados na literatura que utilizam abordagens de redes neurais para melhoramento de DBVS tais como DDFA [2] e NNScore [14, 61, 67].

A entrada para a DeepVS é o resultado da etapa de *virtual screening* executada usando um programa de *docking*. Nos experimentos apresentados nesse trabalho foram utilizados os programas Dock6.6 (aqui referenciado como Dock) e Autodockvina1.1.2 (aqui referenciado como ADV). Para cada receptor presente no banco de

dados DUD geramos um ranking de todos os compostos para aquele receptor baseado nos *scores* resultantes da aplicação da DeepVS para cada complexo receptor-composto. Para mais detalhes de como os *scores* são gerados ver seção 1.3.2.4.

Com o intuito de verificarmos se a DeepVS produz resultados de *virtual screening* melhores do que os dos programas de *docking*, comparamos o ranking gerado a partir da saída da DeepVS com o ranking gerado pelos programas de *docking*. O ranking dos ligantes em ambos os programas de *docking* é feito levando em consideração o maior valor retornado pela função de pontuação *Grid Score* e *Vina Score*, respectivamente. Nos nossos resultados, apresentamos quatro rankings que foram gerados usando as seguintes abordagens: (1) baseado no *score* do Dock6.6 (Dock); (2) baseado no *score* do Autodockvina1.1.2 (ADV); (3) baseado no *score* da DeepVS aplicado à saída do Dock6.6 (DeepVS-Dock); (4) baseado no *score* da DeepVS aplicado à saída do Autodockvina1.1.2. (DeepVS-ADV).

Com o objetivo de avaliar o desempenho das metodologias de *virtual screeening* empregadas usamos como métrica a área sob a curva ROC (AUC), que procura medir simultaneamente duas características importantes em *virtual screenig*: (i) a capacidade de identificar corretamente verdadeiros positivos (ligantes ativos) e (ii) descartar verdadeiros negativos (*decoys*). No qual, uma AUC igual a 0,50 representa uma seleção aleatória de ligantes e o valor de AUC igual a 1,0 reflete a identificação ideal de ligantes ativos. Como uma segunda métrica de desempenho calculamos o fator de enriquecimento (*ef*) que indica quantas vezes o melhor conjunto de compostos recomendado (ligantes) foi selecionado a partir de uma porcentagem do conjunto total de compostos (ligantes e *decoys*).

Na Tabela 1.5 reportamos para cada um dos 40 receptores do DUD os valores de AUC e fator de enriquecimento (*ef*) calculado em 1%, 5%, 10%, 20% e 50% do conjunto de dados, para o caso em que usamos o programa de *docking* Dock6.6 para executar o *vitual screening*. Adicionalmente, a média da AUC e do fator de enriquecimento ($ef_{1\%}$, $ef_{5\%}$, $ef_{10\%}$, $ef_{20\%}$, $ef_{50\%}$) também estão descrito na Tabela 1.5. Em um universo de 40 receptores apenas quatro (ressaltados em negrito) ADA (PDB_ID: 1ndw), GART (PDB_ID: 1c2t), NA (PDB_ID: 1a4g) e RXRr (PDB_ID: 1mvc) obtiveram AUC >0,70, a taxa de sucesso do programa de *docking* Dock6.6 foi de apenas 10% para o banco de dados DUD. Em 57% do banco de dados o Dock6.6 produziu uma AUC <0,50, esse resultado significa que a seleção de ligantes ativos realizada de forma aleatória poderia ser mais efetiva que a seleção gerada pelo o programa.

A AUC média para o Dock6.6 foi de 0,48 um pouco abaixo do resultado reportado para a versão 4.4 do mesmo programa, no qual obteve como valor médio de AUC igual a 0,55 [136] utilizando o mesmo banco de dados o DUD (Tabela 1.17). Os resultados para Dock6.6 poderiam ser impulsionados com emprego da função de pontuação *Amber Score* disponível no pacote AmberTools presente no Dock6.6 [137, 138].

A função de pontuação *Amber Score* é uma abordagem simples de MM-GB/SA (estima a energia de ligação) que não trata de modo explicito efeitos de entropia [139, 140]. De fato, o uso da função de pontuação *Amber Score* com o objetivo de

Tabela 1.5: Valores de AUC (ROC) e fator de enriquecimento à 1%, 5%, 10%, 20% e 50% para cada proteína depositada no DUD correspondente a performance de *virtual screening* do programa Dock6.6. Valores em negrito correspondem a AUC >0,70.

	***Virtual Screening* Dock**					
	ef$_{1\%}$	**ef$_{5\%}$**	**ef$_{10\%}$**	**ef$_{20\%}$**	**ef$_{50\%}$**	**auc**
média	6,7	3,0	2,1	1,3	0,9	0,48
ACE	2,0	1,6	1,2	0,6	0,8	0,41
AChE	1,0	0,8	0,9	0,8	1,2	0,50
ADA	13,0	7,9	5,2	3,3	1,9	**0,86**
ALR2	0,0	0,0	0,4	0,2	0,8	0,38
AmpC	4,5	1,9	1,4	1,4	1,2	0,57
AR	1,4	0,3	0,1	0,1	0,3	0,25
CDK2	24,4	4,8	3,2	2,1	1,1	0,56
COMT	0,0	10,9	5,5	2,7	1,3	0,64
COX1	0,0	0,0	0,0	0,2	0,6	0,35
COX2	0,3	0,7	1,3	1,2	1,3	0,58
DHFR	6,0	2,1	1,7	1,3	0,9	0,48
EGFr	19,9	5,6	3,2	1,9	0,9	0,49
ER$_{agonist}$	3,0	1,2	1,2	1,0	0,8	0,43
ER$_{antagonist}$	7,1	5,1	3,5	2,2	1,4	0,68
FGFrl	10,2	5,3	3,3	2,1	1,0	0,49
FXa	5,0	2,3	1,7	1,1	0,9	0,49
GART	9,2	6,6	6,7	4,5	1,9	**0,90**
GPB	0,0	0,4	0,2	0,1	0,3	0,23
GR	2,5	0,8	0,4	0,3	0,3	0,22
HIVPR	0,0	0,8	0,4	0,4	0,2	0,16
HIVRT	4,9	2,5	2,0	1,0	0,6	0,40
HMGR	5,6	1,7	0,9	0,4	0,3	0,16
HSP90	0,0	1,7	1,3	1,0	0,8	0,39
InhA	10,7	2,6	1,8	1,2	0,7	0,38
MR	6,1	1,3	1,3	1,0	0,5	0,36
NA	10,2	8,9	5,5	3,3	1,4	**0,73**
P38MAP	17,3	5,9	3,4	2,2	1,2	0,62
PARP	18,3	5,5	3,0	1,5	1,1	0,51
PDE5	5,8	2,4	1,6	0,9	0,6	0,39
PDGFrb	8,9	3,3	1,8	1,2	0,7	0,38
PNP	0,0	1,6	1,2	0,6	0,7	0,44
PPARg	2,5	1,0	0,6	0,4	0,9	0,49
PR	0,0	1,5	1,1	0,6	0,4	0,33
RXRa	20,8	5,1	3,0	2,0	1,6	**0,73**
SAHH	0,0	0,6	1,2	1,1	0,8	0,50
SRC	19,5	5,9	3,3	1,8	0,8	0,44
thrombin	1,5	1,8	2,0	1,4	1,4	0,60
TK	0,0	0,9	1,4	1,4	1,6	0,63
trypsin	6,8	3,7	2,0	1,6	0,9	0,51
VEGFr2	21,7	4,6	2,4	1,4	0,8	0,43

aprimorar o resultado de *virtual screening* para o programa Dock6.6 foi demonstrado por Kolossvary *et al*. [141]. Os autores desenvolveram experimentos para testar a performance do Dock6.6 associado a combinação de duas funções de pontuação a *Amber Score* e a *LMOD Score* (função de pontuação baseado em campo de força que explora o espaço conformacional seguindo os modos vibratórios de baixa frequência), aplicados ao banco de dados DUD. Os resultados obtidos quando empregadas as duas funções de pontuação *Amber Score* e a *LMOD Score* são superiores (AUC média = 0,60) ao reportadas nesse trabalho (AUC média = 0,48), no qual foi adotado como função de pontuação a *Grid Score* (Tabela 1.17).

Particularmente, optamos por não empregar a função de pontuação *Amber Score* em razão de três características: (1) O programa Dock6.6 possui como função de pontuação para geração da pose a *Grid Score*; (2) a função de pontuação *Amber Score* foi pensada e é atualmente utilizada apenas como uma função de reclassificação ou *rescoring* dos compostos; (3) a *Amber Score* demanda um alto custo computacional por envolver abordagens como campo de força AMBER e GAFF.

Os resultados para *virtual screening* utilizando o programa de *docking* Autodockvina1.1.2 estão apresentados na Tabela 1.6. O ADV possui um desempenho melhor se comparado com o programa de *docking* Dock6.6. Em 30% do conjunto de dados o ADV produziu uma AUC >0.70, ressaltado em negrito na Tabela 1.6. Além disso, em apenas 22,5% do banco dados o ADV reportou uma AUC <0.50 que corresponde a uma distribuição aleatória dos dados.

A AUC média para o ADV foi de 0,62 a qual está de acordo com resultados de trabalhos anteriores publicados na literatura, para teste da performance do programa ADV usando como banco de dados o DUD [131] (Tabela 1.17). Os resultados para ADV poderiam ser otimizados alterando o parâmetro *exhaustiveness* [70] para um número maior que 16. Optamos por manter as configurações do programa de *docking* ADV semelhante as configurações aplicadas para esse mesmo programa descritas em trabalhos relacionados que utilizaram redes neurais rasas para melhoramento de DBVS [2, 14, 61, 67].

Na Tabela 1.7 apresentamos o resultado da performance de *virtual screening* para cada um dos 40 receptores do DUD usando as seguintes abordagens: (1) dois diferentes programas de *docking* Dock6.6 (Dock) e Autodockvina1.1.2 (ADV); (2) a rede neural profunda DeepVS aplicada individualmente a saída dos programas Dock6.6 (DeepVS-Dock) e Autodockvina1.1.2 (DeepVS-ADV). A saída dos programas de *docking* consiste na pose (ou seja, na conformação de menor energia encontrado durante o processo de *docking* molecular) dos compostos. Um arquivo de saída é formado pelas coordenadas espacias dos átomos formadores do composto, suas respectivas cargas atômicas e o tipo de átomos presente no composto. Cada complexo proteína-composto gerado durante a etapa de *docking* é processado pela DeepVS separadamente. Para cada sistema, reportamos a AUC ROC e o fator de enriquecimento (*ef*) a 2% e 20%. Também reportamos o valor de fator de enriquecimento máximo (ef_{max}), que consiste no valor máximo de fator de enriquecimento encontrado em uma lista de

Tabela 1.6: Valores de AUC (ROC) e fator de enriquecimento à 1%, 5%, 10%, 20% e 50% para cada proteína depositada no DUD correspondente a performance de *virtual screening* do programa Autodockvina1.1.2. Valores em negrito correspondem a AUC >0,70.

	***Virtual Screening* ADV**					
	$ef_{1\%}$	**$ef_{5\%}$**	**$ef_{10\%}$**	**$ef_{20\%}$**	**$ef_{50\%}$**	**auc**
média	7,8	4,1	2,9	2,0	1,4	0,62
ACE	6,0	2,0	1,2	1,4	0,7	0,38
AChE	2,9	4,2	3,6	3,0	1,5	0,68
ADA	0,0	0,0	0,0	1,1	0,9	0,47
ALR2	4,0	7,0	4,2	2,5	8,2	0,70
AmpC	0,0	0,0	0,0	0,2	0,3	0,23
AR	16,2	8,9	5,7	3,5	1,5	**0,74**
CDK2	12,2	6,4	4,0	2,1	1,4	0,66
COMT	20,0	3,6	1,8	1,4	0,5	0,41
COX1	15,5	6,4	6,0	3,6	1,7	**0,79**
COX2	27,9	12,3	7,2	3,8	1,7	**0,84**
DHFR	11,5	5,6	4,5	3,5	1,9	**0,86**
EGFr	3,6	1,8	1,9	1,5	1,2	0,58
$ER_{agonist}$	15,1	9,0	5,1	3,3	1,7	**0,79**
$ER_{antagonist}$	10,5	4,6	3,3	2,3	1,2	0,66
FGFrl	0,0	0,3	0,7	1,0	0,9	0,48
FXa	1,4	1,5	1,9	1,5	1,5	0,66
GART	0,0	0,0	1,9	2,6	1,9	**0,77**
GPB	0,0	1,5	1,2	1,2	1,1	0,52
GR	7,6	2,3	1,5	1,2	1,1	0,57
HIVPR	5,8	4,9	3,8	2,6	1,5	**0,72**
HIVRT	9,8	4,0	3,0	1,8	1,3	0,64
HMGR	0,0	0,0	0,3	0,6	0,8	0,42
HSP90	0,0	0,0	0,8	0,8	1,0	0,54
InhA	15,4	6,4	3,4	1,8	1,0	0,54
MR	30,6	14,4	7,3	4,0	1,6	**0,82**
NA	0,0	0,4	0,2	0,5	0,7	0,40
P38MAP	1,6	2,3	2,5	2,2	1,3	0,62
PARP	9,2	5,5	4,2	2,9	1,6	**0,74**
PDE5	7,7	3,5	2,7	1,7	1,1	0,57
PDGFrb	9,5	2,7	1,4	1,1	0,8	0,48
PNP	0,0	0,8	1,2	1,8	1,4	0,66
PPARg	1,2	3,2	2,5	1,7	1,3	0,64
PR	0,0	1,5	1,1	1,1	0,7	0,43
RXRa	31,2	16,2	8,0	4,2	1,9	**0,93**
SAHH	9,0	7,2	5,2	3,2	1,7	**0,80**
SRC	2,6	3,5	2,8	2,2	1,5	0,69
thrombin	9,1	5,2	4,5	2,8	1,5	**0,73**
TK	0,0	0,0	0,0	0,2	1,5	0,55
trypsin	6,8	1,8	1,4	1,6	1,4	0,63
VEGFr2	9,5	3,5	2,7	1,6	1,1	0,56

compostos ranqueados (Tabela 1.7). Dentre as quatro abordagens, a DeepVS-ADV exibe o melhor valor médio de AUC e *ef* a 2% e 20%. DeepVS-ADV tem o melhor valor de AUC para 20 dos 40 receptores do banco de dados DUD.

Em geral, a qualidade do *docking* influencia na performance da DeepVS. O que é esperado, uma vez que estados conformacionais incorretos podem gerar ruído para a DeepVS, de tal forma que a rede pode aprender representações inconsistentes ou incoerentes relacionadas ao correto modo de ligação do complexo proteína-composto. Na Seção 1.4.4, discutiremos em mais detalhas como a qualidade dos resultados dos programas de *docking* podem influenciar na performance da DeepVS.

Um exemplo de como o resultado do *docking* pode influenciar de forma positiva no desempenho da DeepVS é o caso da abordagem DeepVS-ADV. O programa de *docking* molecular ADV produziu uma AUC média melhor (AUC média = 0,62) se comparado ao programa Dock (AUC média = 0,48). Consequentemente, a performance da DeepVS associada ao programa ADV (DeepVS-ADV) gerou melhores valores de AUC, $ef_{2\%}$ e $ef_{20\%}$, quando comparado à performance da DeepVS associada ao programa Dock (DeepVS-Dock) (Tabela 1.7). De fato, o desempenho médio do programa Dock (AUC média = 0,48) para todo o bando de dados DUD não é melhor que uma seleção de ligantes realizada de forma aleatória. Por outro lado, há casos em que a abordagem DeepVS (DeepVS-Dock) foi capaz de melhorar o resultado do programa Dock em até 220%. Por exemplo, para os receptores AR (PDB_ID: 2ao6), COX1 (PDB_ID: 1p4q), HSP90 (PDB_ID: 1uy6), InhA (PDB_ID: 1p44), PDE5 (PDB_ID: 1xp0), PDGFrb (modelo) e PR (PDB_ID: 1sr7) o Dock6.6 produziu AUC <0,40 o que equivale a uma seleção aleatória de dados, a DeepVS conseguiu melhorar o resultados para todos esses receptores resultando em valores de AUC >0,70.

Na Figura 1.30, comparamos os valores de AUC reportados pela a abordagem DeepVS-ADV e o programa de *docking* ADV para cada um dos 40 receptores presentes no banco de dados DUD. A DeepVS-ADV apresentou o valor de AUC >0,70 para 33 receptores. Em contra partida, para programa de *docking* ADV esse número foi de apenas 13. O número de receptores com AUC <0,50 foi de 2 para DeepVS-ADV e 9 para ADV. A AUC gerada para a DeepVS-ADV é mais alta que a do programa de *docking* ADV em 31 receptores. Em média, a AUC computada para a DeepVS-ADV (AUC = 0,81) foi 31% melhor que AUC reportada para o programa de *docking* ADV (AUC = 0,62). Adicionalmente, quando selecionamos 20% dados levando em consideração o ranking dos compostos, em média, o valor do fator de enriquecimento da DeepVS-ADV ($ef_{20\%}$ = 3,1) foi 55% maior que o valor de fator de enriquecimento reportado pelo ADV ($ef_{20\%}$ = 2,0).

Tabela 1.7: Valores de AUC ROC, fator de enriquecimento (*ef*) à 2%, 20% e fator de enriquecimento máximo (ef_{max}) para cada proteína depositada no DUD correspondente a performance de *virtual screening* de três diferentes abordagens: Autodockvina1.1.2., Dock6.6 e DeepVS. Valores em negrito indicam ao maior valor de AUC computado em cada caso.

	Dock				**DeepVS-Dock**				**ADV**				**DeepVS-ADV**			
	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	auc	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	auc	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	auc	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	auc
média	20,1	5,3	1,3	0,48	16,9	5,9	3,0	0,74	16,4	6,0	2,0	0,62	16,0	6,6	3,1	**0,81**
ACE	36,2	3,0	0,6	0,41	3,6	2,0	2,7	**0,75**	9,1	3,0	1,4	0,38	2,2	1,0	1,8	0,72
AChE	12,1	0,5	0,8	0,50	1,6	0,5	0,8	0,48	5,4	4,8	3,0	**0,68**	4,0	1,9	1,1	0,51
ADA	24,5	13,0	3,3	0,86	8,5	6,5	4,1	**0,87**	1,1	0,0	1,1	0,47	9,2	2,2	3,3	0,83
ALR2	1,0	0,0	0,2	0,38	1,7	0,0	1,3	0,56	8,2	3,8	2,5	**0,70**	5,7	3,8	1,3	0,66
AmpC	5,1	4,8	1,4	**0,57**	1,3	0,0	0,0	0,44	1,0	0,0	0,2	0,23	1,1	0,0	0,0	0,42
AR	7,3	0,7	0,1	0,25	36,5	18,3	4,1	0,80	36,5	14,2	3,5	0,74	21,9	11,5	4,2	**0,88**
CDK2	36,6	11,9	2,1	0,56	11,4	8,9	3,7	**0,82**	18,3	8,9	2,1	0,66	6,1	4,0	2,9	0,79
COMT	20,0	17,8	2,7	0,64	40,1	13,4	3,2	0,88	40,1	8,9	1,4	0,41	20,0	8,9	4,6	**0,92**
COX1	1,1	0,0	0,2	0,35	35,0	8,2	3,0	0,75	20,0	10,3	3,6	**0,79**	5,0	2,1	2,8	0,77
COX2	1,3	0,7	1,2	0,58	18,4	12,4	3,6	0,78	36,8	21,4	3,8	0,84	36,8	12,7	4,3	**0,91**
DHFR	36,5	3,7	1,3	0,48	18,3	10,4	4,5	0,88	36,5	8,2	3,5	0,86	9,1	6,7	4,8	**0,94**
EGFr	34,5	12,0	1,9	0,49	23,0	8,4	4,6	**0,93**	4,9	2,5	1,5	0,58	8,6	5,5	3,6	0,86
$ER_{agonist}$	18,1	2,3	1,0	0,43	6,5	0,8	3,5	0,75	17,7	16,6	3,3	0,79	8,1	6,0	3,9	**0,88**
$ER_{antagonist}$	7,1	5,9	2,2	0,68	7,1	5,9	4,0	**0,90**	13,8	8,9	2,3	0,66	7,4	3,8	3,8	0,88
FGFrl	36,6	8,9	2,1	0,49	15,7	8,1	4,6	**0,91**	1,1	0,0	1,0	0,48	36,6	7,7	3,3	0,85
FXa	36,9	3,2	1,1	0,49	2,0	0,4	1,7	0,71	3,2	2,1	1,5	0,66	4,3	1,4	3,9	**0,86**
GART	10,5	7,4	4,5	0,90	12,3	4,9	4,8	**0,92**	2,9	0,0	2,6	0,77	2,6	0,0	2,4	0,77
GPB	1,0	0,0	0,1	0,23	2,7	1,9	1,2	0,51	3,1	2,9	1,2	**0,52**	1,0	0,0	0,9	0,42
GR	18,4	1,9	0,3	0,22	11,1	7,6	2,2	0,49	18,4	4,4	1,2	0,57	20,3	10,8	4,4	**0,91**
HIVPR	1,0	0,0	0,4	0,16	4,1	0,9	2,1	0,51	36,6	6,6	2,6	0,72	6,2	5,6	4,1	**0,88**
HIVRT	36,9	4,9	1,0	0,40	36,9	6,2	2,8	0,69	24,6	6,2	1,8	0,64	7,4	4,9	2,3	**0,73**
HMGR	18,2	4,2	0,4	0,16	36,5	2,8	1,0	0,24	1,0	0,0	0,6	0,42	36,5	19,6	4,9	**0,96**
HSP90	2,3	0,0	1,0	0,39	5,8	2,0	3,3	0,74	1,3	0,0	0,8	0,54	10,0	4,1	5,0	**0,94**
InhA	36,7	5,3	1,2	0,38	36,7	13,0	4,3	0,90	36,7	11,2	1,8	0,54	6,7	16,6	4,5	**0,94**
MR	18,3	3,3	1,0	0,36	14,7	6,7	2,3	0,55	36,7	20,0	4,0	**0,82**	24,4	16,7	3,3	**0,82**
NA	36,6	13,2	3,3	0,73	3,5	0,0	2,8	**0,78**	1,1	0,0	0,5	0,40	2,1	0,0	1,0	0,68
P38MAP	33,8	11,1	2,2	0,62	33,8	16,0	4,2	**0,91**	2,8	1,4	2,2	0,62	21,6	16,4	3,9	0,87
PARP	36,6	10,7	1,5	0,51	2,9	0,0	0,8	0,63	36,6	4,6	2,9	**0,74**	1,7	0,0	0,8	0,65
PDE5	36,5	3,0	0,9	0,39	12,2	3,9	2,9	0,75	36,5	6,9	1,7	0,57	36,5	8,9	3,1	**0,86**
PDGFrb	36,8	5,4	1,2	0,38	17,4	14,4	4,5	**0,92**	36,8	5,4	1,1	0,48	36,8	19,2	3,7	0,91
PNP	4,8	4,0	0,6	0,44	10,3	2,0	4,8	**0,94**	3,0	2,0	1,8	0,66	4,2	0,0	4,0	0,86
PPARg	2,6	1,8	0,4	0,49	36,9	1,2	2,0	0,79	4,1	3,1	1,7	0,64	36,9	2,5	4,4	**0,87**
PR	3,1	1,8	0,6	0,33	4,8	1,8	3,3	0,70	1,5	0,0	1,1	0,43	8,5	5,5	2,4	**0,77**
RXRa	36,4	12,1	2,0	0,73	6,4	4,9	4,2	0,91	36,4	24,3	4,2	**0,93**	12,1	9,7	3,0	0,85
SAHH	1,7	1,5	1,1	0,50	19,3	15,1	4,7	0,94	13,5	10,5	3,2	0,80	19,9	18,1	4,7	**0,95**
SRC	38,4	12,3	1,8	0,44	25,6	12,9	3,9	**0,88**	4,0	2,9	2,2	0,69	19,2	9,7	3,4	0,85
thrombin	3,6	2,3	1,4	0,60	36,3	5,4	1,1	0,59	18,1	6,2	2,8	0,73	36,3	6,2	3,2	**0,83**
TK	2,0	0,0	1,4	**0,63**	1,2	0,0	0,5	0,44	1,5	0,0	0,2	0,55	1,4	0,0	0,2	0,54
trypsin	36,1	4,5	1,6	0,51	18,0	5,6	1,9	0,65	9,8	3,4	1,6	0,63	36,1	6,8	2,4	**0,80**
VEGFr2	36,7	10,9	1,4	0,43	9,7	6,1	3,8	0,88	36,7	5,4	1,6	0,56	36,7	4,8	4,1	**0,90**

A comparação dos valores de AUC computados para DeepVS-Dock e o programa de *docking* Dock referente aos 40 receptores presentes no bando de dados DUD é representada na Figura 1.31. Em média, a AUC reportada pela DeepVS-Dock (AUC = 0,74) foi 54% melhor que o valor apresentado para o Dock (AUC = 0,48). Enquanto o programa de *docking* Dock atingiu valores de AUC >0,70 para somente 10% dos receptores (4 proteínas), a DeepVS-Dock registrou uma AUC >0,70 para 68% de receptores (27 proteínas). O número de receptores com AUC <0,50 foi cinco para DeepVS-Dock e 23 para o Dock. A AUC reportada pela a DeepVS-Dock foi mais alta em 36 receptores quando comparado com os valores de AUC provenientes do programa de *docking* Dock. Finalmente, quando selecionamos 20% dos dados de acordo com a lista de compostos ranqueados, em média, o valor do fator de enriquecimento da DeepVS-Dock ($ef_{20\%}$ = 3,0) foi mais que o dobro reportado pelo o Dock ($ef_{20\%}$ = 1,3).

Segundo Durrant & Mccammon [61] nenhuma função de pontuação ou metodologia de *rescoring* única é perfeitamente adequada para todos os tipos de receptores (proteínas). Apesar da DeepVS demonstrar uma alta taxa de sucesso para a maioria dos receptores, há casos em que o uso da DeepVS pode piorar os resultados. Como por exemplo, para o programa Dock6.6 a DeepVS reduziu o resultado de AUC para quatro receptores AChE (PDB_ID: 1eve), AmpC (PDB_ID: 1xgj), thrombin (PDB_ID 1ba8) e TK (PDBPDB_ID: 1kim). De fato, o *vitual screening* efetuado com o Dock6.6 para AmpC e TK tem o melhor resultado de AUC quando comparado às outras três estratégias (Tabela 1.7). Porém, para os outros dois receptores a diferença foi insignificante: AChE com o Dock6.6 possui AUC de 0,50, quando aplicamos a DeepVS a AUC cai para 0,48; thrombin tem AUC de 0,60 com o DOCk6.6, a qual cai para 0,59 quando aplica-se a DeepVS.

Levando-se em consideração o programa de *docking* ADV, a DeepVS reduziu o valor de AUC para sete receptores. Em seis receptores a AUC reportada pelo ADV foi a melhor quando comparada as demais metodologias de *virtual screening* (Tabela 1.7), e em apenas um desses seis receptores (TK) a AUC reportada pelo ADV não foi a melhor dentre as quatro abordagens. Para o TK, a AUC advinda do ADV foi de 0,55 e a DeepVS reduziu esse valor para 0,54.

Os resultados dos experimentos apresentados nessa seção, utilizando a saída de dois programas de *docking* distintos, são uma forte evidência de que a rede DeepVS pode ser usada como uma estratégia efetiva para melhorar *Docking-Based Virtual Screening*-DBVS.

1.4.2 Qual Abordagem de *Virtual Screening* Utilizar Dado um Projeto Especifico?

Os dados apresentados na Tabela 1.7, assim como mencionado na literatura [20, 58, 61], sugerem que a melhor abordagem para *virtual screening* depende direta-

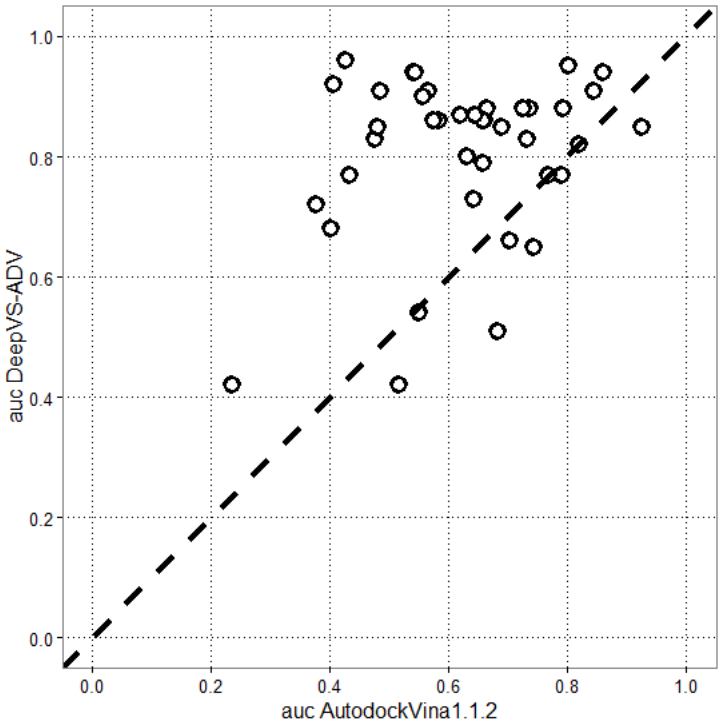


Figura 1.30: **DeepVS-ADV vs AutodockVina1.1.2**. Valores de AUC calculados para avaliar a performance das abordagens de DeepVS-ADV e AutodockVina1.1.2. Os círculos representam o valor de AUC reportado para cada um dos 40 receptores presentes no banco de dados DUD, a linha tracejada indica o limite onde a performance de ambos os métodos se sobrepõem.

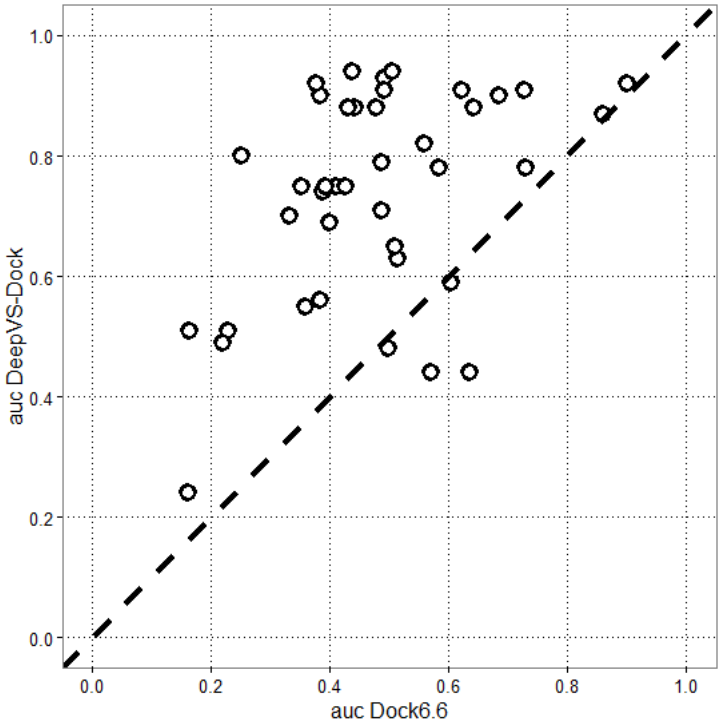


Figura 1.31: **DeepVS-ADV vs Dock6.6**. Valores de AUC calculados para avaliar a performance das abordagens de DeepVS-Dock e Dock6.6. Os círculos representam o valor de AUC reportado para cada um dos 40 receptores presentes no banco de dados DUD, a linha tracejada indica o limite onde a performance de ambos os métodos se sobrepõem.

mente do receptor (proteína) a ser estudado. Embora a DeepVS possui em média o melhor desempenho, ela não foi de forma constante a melhor abordagem de *virtual screening* para todos os receptores. Por exemplo, para o receptor RXRa (PDB_ID: 1mvc) a melhor escolha para um protocolo de *virtual screening* seria o uso do programa ADV, dado que o mesmo reportou AUC = 0,93, resultado superior à DeepVS-Dock e DeepVS-ADV.

É notável que estudos prévios com ligantes conhecidamente ativos e não-ativos para um determinado receptor se faz necessários para a escolha da melhor abordagem de *virtual screening* a ser adotada [20, 58, 61]. Não apenas relacionado a capacidade do programa de *docking* retornar a pose correta para o ligante cristalográfico (estudos de RMSD) [58], mas também estudos com controles positivos e negativos (ligantes ativos) / negativos (não-ativos) [61] e avaliação da abordagem usando métricas bem estabelecidas como AUC ROC e fator de enriquecimento. Uma vez confirmado a eficácia da abordagem escolhida, pode dar-se inicio aos processos de *virtual screening* em bibliotecas com um grande número de moléculas como por exemplo o ZINC [20, 58, 61].

Porém, há casos em que o uso da abordagem DeepVS é fortemente recomendada, como por exemplo para os receptores: COMT, COX2, DHFR, EGFr, $ER_{antagonist}$, FGFrl, GART, GR, HMGR, HSP90, InhA, P38MAP, PDGFrb, PNP, SAHH e VEGFr2. Para esses receptores, a DeepVS não apenas apresentou melhor desempenho do que os demais programas de *docking*, mas também registrou um alto valor de AUC ($\geq$ 0,90).

Embora a abordagem proposta nesse trabalho não seja a ideal para executar *virtual screening* para todos os tipos de receptores, ela demonstra ser o método (dentre os testados) que mais se aproxima desse objetivo (Seção 1.4.5).

1.4.3 Sensibilidade da DeepVS aos Hiperparâmetros

Nesta seção descrevemos os resultados referentes aos experimentos realizados para investigar a sensibilidade da DeepVS a seus principais hiperparâmetros. Parâmetros que não podem ser diretamente aprendidos pela rede no processo regular de treinamento são chamados de "hiperparâmetros", geralmente esses parâmetros expressam propriedades de alto nível do modelo, como por exemplo a rapidez com que a rede deve aprender.

Para todos os experimentos computacionais relacionados aos hiperparâmetros da DeepVS, foram utilizados os complexos proteína-composto resultantes do processo de *docking* com o programa Autodockvina 1.1.2 (ADV). Portanto, todos os resultados reportados nessa seção foram gerados usando a DeepVS-ADV. Optamos por usar o ADV por que o resultado de *docking* desse programa é superior ao resultado do programa Dock6.6. Em nossos experimentos, variamos um dos hiperparâmetros, como

por exemplo a taxa de aprendizado λ, e fixamos os demais. Os hiperparâmetros foram fixados nos seguintes valores: $d^{atm/amino/chrg/dist} = 200$, $cf = 400$, $\lambda = 0,075$, $k_c = 6$ e $k_p = 2$.

Na Tabela 1.8, apresentamos os resultados dos experimentos onde variamos o hiperparâmetro tamanho do *embedding* e reportamos o ef_{max}, $\text{ef}_{2\%}$, $\text{ef}_{20\%}$ e o valor de AUC médios para a DeepVS. Como podemos verificar na Tabela 1.8, os valores relativos ao tamanho do *embedding* >50 melhoram principalmente o resultado para o fator de enriquecimento. Experimentos adicionais demonstraram que valores maiores que 200 para o hiperparâmetro tamanho *embedding* não melhoram os resultados de AUC ou fator de enriquecimento.

Tabela 1.8: Teste de sensibilidade da DeepVS com relação ao hiperparâmetro tamanho do *embedding*.

$d^{atm/amino/chrg/dist}$	ef_{max}	$\text{ef}_{2\%}$	$\text{ef}_{20\%}$	auc
50	15,4	6,5	3,0	0,803
100	15,6	6,8	3,1	0,799
200	16,0	6,6	3,1	0,807

Os resultados dos experimentos para a variação do número de filtros da camada convolucional (cf) são apresentados na Tabela 1.9. A AUC melhora quando aumenta-se o número de filtros da camada convolucional até o valor de 400. Por outro lado, o uso do hiperparâmetro cf = 200 resultou no melhor valor de fator de enriquecimento em 2% ($\text{ef}_{2\%}$) do conjunto de dados. O que é um fator que deve ser considerado quando pretende-se utilizar uma quantidade limitada de ligantes para a fase de teste experimental.

Tabela 1.9: Sensibilidade da DeepVS ao hiperparâmetro número de filtros da camada convolucional (cf).

cf	ef_{max}	$\text{ef}_{2\%}$	$\text{ef}_{20\%}$	auc
100	14,7	6,4	3,0	0,797
200	16,3	7,3	3,0	0,799
400	16,0	6,6	3,1	0,807
800	16,8	7,0	3,1	0,804

Na Tabela 1.10, apresentamos os resultados dos experimentos para o treinamento da DeepVS variando a taxa de aprendizado (λ). Pelos resultados da tabela, fica evidente que taxas de aprendizado entre 0,05 e 0,1 funcionam melhor para o banco de dados DUD. A taxa de aprendizado de 0,1 reportou melhores resultados em termos de AUC e fator de enriquecimento. Experimentos adicionais também demonstraram que taxas de aprendizado maiores do que 0,1 não apresentam melhores resultados.

O impacto do uso de diferentes números de átomos vizinhos tanto para o composto (k_c) quanto para a proteína (k_p) também foi investigado. Na Tabela 1.11 apresentamos o resultados para a variação de ambos os parâmetros k_c e k_p. Por exemplo,

Tabela 1.10: Sensibilidade da DeepVS com relação ao hiperparâmetro taxa de aprendizado (λ).

λ	$\mathbf{ef}_{max}$	$\mathbf{ef}_{2\%}$	$\mathbf{ef}_{20\%}$	auc
0,1	17,3	6,9	3,2	0,809
0,075	16,0	6,6	3,1	0,807
0,05	15,5	6,6	3,0	0,801
0,025	14,7	6,4	3,0	0,795
0,01	15,7	6,2	3,1	0,800

quando k_c = 0 e k_p = 5, significa que somente a informação da proteína foi utilizada. O inverso é verdadeiro, ou seja, quando k_p=0 apenas a informação do ligante é utilizada pela DeepVS. Na primeira metade da tabela, nós mantemos o valor de k_p fixo em cinco e variamos o valor de k_c. Na segunda metade da tabela, nós fixamos k_c para o valor igual a seis e variamos os valores de k_p. Esses valores foram fixados baseados em dados disponíveis na literatura [109].

Tabela 1.11: Sensibilidade da DeepVS para número de átomos vizinhos selecionados a partir do complexo proteína-composto.

k_c	k_p	$\mathbf{ef}_{max}$	$\mathbf{ef}_{2\%}$	$\mathbf{ef}_{20\%}$	auc
0	5	6,99	2,18	1,54	0,574
1	5	15,55	4,22	2,27	0,697
2	5	15,44	5,62	2,59	0,743
3	5	16,74	6,38	2,76	0,752
4	5	17,38	6,25	2,91	0,782
5	5	19,07	6,47	3,18	0,799
6	5	17,89	6,79	3,04	0,799
6	4	17,02	6,38	3,17	0,801
6	3	16,44	6,82	2,99	0,793
6	2	16,03	6,62	3,14	0,807
6	1	16,03	6,99	3,13	0,806
6	0	16,92	6,95	3,06	0,803

Na primeira metade da Tabela 1.11, notamos que, quando aumentamos o número de vizinhos selecionados a partir do composto (k_c), tanto o valor do fator de enriquecimento (*ef*) quanto a AUC também sofre um aumento significativo. Na segunda ametade da tabela, verificamos que usando valores de k_p >2 a performance da DeepVS-ADV cai consideravelmente. Formulamos duas hipóteses que podem explicar a redução da performance da DeepVS ao utilizar mais informação da proteína: (1) a qualidade do *virtual screening* gerado pelos os programas de *docking*; (2) a DeepVS não está utilizando a informação da proteína, ou seja, está realizando *virtual screening* baseado no ligante. Enquanto a hipótese 2 será analisada com maiores detalhes em trabalhos futuros, na seção 1.4.4 mostramos evidências que corroboram a hipótese 1. Na seção 1.4.4 demonstramos que, quando o resultado do *virtual screening* é ruim

($AUC \leq 0,50$) a informação da proteína contribui de forma negativa no aprendizado da DeepVS. Isto pode estar relacionado com o estado conformacional dos compostos selecionado pelo programa ADV. Uma vez dado como entrada para rede um complexo proteína-composto com estado conformacional errôneo isso pode gerar falsas informações com relação a interação entre o composto e a proteína. Por exemplo, átomos de um composto com geometria incorreta podem estar próximos a átomos da proteína que não estão relacionados com o modo de ligação proteína-composto.

Adicionalmente aos testes da sensibilidade da DeepVS ao seus hiperparâmetros, realizamos experimentos para testar a performance da DeepVS com relação a variabilidade da semente do gerador de números aleatórios, com o objetivo de avaliar a robustez do DeepVS em relação à inicialização das matrizes de peso (parâmetros de rede). Foram executadas 10 rodadas para DeepVS-ADV usando uma semente aleatória diferente em cada execução, e mantendo os hiperparâmetros fixos: $d^{atm/amino/chrg/dist} = 200$, $cf = 400$, $\lambda = 0.075$, $k_c = 6$ e $k_p = 2$.

Tabela 1.12: Sensibilidade da DeepVS a diferentes sementes escolhidas de forma aleatória.

rodada	**ef**$_{max}$	**ef**$_{2\%}$	**ef**$_{20\%}$	**auc**
média	15,97	6,82	3,10	0,805
desvio padrão	0,73	0,16	0,04	0,003
1	16,53	6,88	3,19	0,807
2	16,32	6,63	3,06	0,799
3	15,30	6,76	3,10	0,805
4	14,92	6,89	3,12	0,807
5	17,08	6,74	3,10	0,807
6	16,82	6,84	3,05	0,804
7	15,90	6,52	3,07	0,803
8	15,59	7,00	3,06	0,805
9	16,08	6,86	3,12	0,806
10	15,18	7,04	3,11	0,803

Na Tabela 1.12, exibimos os resultados dos experimentos de variabilidade de sementes para 10 diferentes rodadas da DeepVS. Esta tabela demonstra que o desvio padrão é muito próximo ao valor zero tanto para o fator de enriquecimento quanto para AUC. Isso significa que a variação da semente aleatória não interfere de forma direta na performance da DeepVS.

1.4.4 Qualidade do *Virtual Screening* Versus a Performance da DeepVS

Nesta seção discutimos como a qualidade do *virtual screening* pode alterar a performance da DeepVS. Como descrito na seção 1.1.4.1 a abordagem do *virtual*

screenig baseado em *docking* ou DBVS é divida em duas: (1) método de busca; e (2) função de avaliação ou pontuação. Dessa forma, diferentes orientações e conformações possíveis para um composto no sítio de ligação da proteína são exploradas de modo a encontrar um mínimo global de energia ou a solução de conformação ótima para aquele composto, o qual é selecionado usando uma função de pontuação. Infelizmente, as funções de pontuação muitas vezes não retornam o melhor estado conformacional do composto. De forma que, para um programa de *docking* é mais fácil gerar a melhor conformação de um determinado composto durante a busca conformacional do que sua função de pontuação predizer o correto modo de ligação do composto para sua respectiva proteína alvo.

Nesse trabalho, utilizamos a área sob a curva ROC (AUC) como um método para classificar a qualidade do *virtual screening* e em função da AUC avaliar a capacidade do método em distinguir entre verdadeiros positivos (ligantes) e falsos positivos (*decoys*). Portanto, utilizamos aqui o termo "*virtual screening* de boa qualidade" para designar o resultado de *virtual screening* para programa de *docking* que reportou a AUC >0,75 e o termo "*virtual screening* de pobre qualidade" para o resultado de *virtual screening* do programa de *docking* que reportou a AUC <0,50. Para esse trabalho o termo "qualidade do *virtual screening*" significa diretamente a capacidade do programa em classificar de forma correta ligantes ativos em detrimento de *decoys*, ou seja, a capacidade da sua função de pontuação em estimar corretamente a qualidade das conformações produzidas durante a busca conformacional de forma a gerar uma lista ordenada de compostos de acordo com a sua afinidade pelo receptor.

Nos resultados dos experimentos reportados na Tabela 1.11, notamos que quando o contexto dos átomos formadores do complexo proteína-composto são criados usando valores de k_p >2 (o número de átomos vizinhos advindos da proteína) não repercute no melhoramento do resultado de AUC média. De fato, se for utilizada somente a informação proveniente do composto (k_p = 0), o que equivaleria a uma performance de *virtual screening* baseado no ligante, o resultado de AUC média em si só é bastante expressivo (0,803). Nossa hipótese é que esse comportamento está diretamente ligado à qualidade do *virtual screening* gerado pelos programas de *docking* que é dado como entrada para DeepVS, que varia muito entre as 40 proteínas DUD. Com o intuito de testar a nossa hipótese, nós analisamos separadamente a AUC advinda da DeepVS para proteínas do DUD cuja saída do programa de *docking* ADV era considerada boa qualidade (AUC >0,75) (Tabela 1.13) ou cuja a saída do *docking* era considerada de pobre qualidade (AUC <0,50) (Tabela 1.14).

Na Tabela 1.13, apresentamos o resultado de AUC advindo da DeepVS-ADV para proteínas cuja AUC reportada pelo ADV fosse maior que 0,75. O resultado para três diferentes valores de k_p foram registrados, $k_p = 0$, $k_p = 2$ e $k_p = 5$. Para todos os três experimentos o valor do k_c foi constante ($k_c = 6$). Verificamos que para proteínas cujo o resultado de *virtual screening* possui uma boa qualidade, a média de AUC da DeepVS aumenta quando valor de k_p é elevado. Esses resultados sugerem que a DeepVS pode se beneficiar da informação da proteína e assim gerar melhores resultados,

quando ela recebe uma correta informação estrutural do complexo proteína-composto.

Tabela 1.13: Resultados da DeepVS-ADV para proteínas com boa qualidade de *virtual screening* reportada.

	AUC >0,75		
	$k_p = 0$	$k_p = 2$	$k_p = 5$
média	0,83	0,86	0,87
COX1	0,78	0,77	0,80
COX2	0,89	0,91	0,91
DHFR	0,96	0,94	0,96
$ER_{agonist}$	0,89	0,88	0,89
GART	0,78	0,77	0,77
MR	0,80	0,82	0,80
RXRa	0,64	0,85	0,90
SAHH	0,93	0,95	0,95

Tabela 1.14: Resultados da DeepVS-ADV para proteínas com pobre qualidade de *virtual screening* reportada.

	AUC <0,50		
	$k_p = 0$	$k_p = 2$	$k_p = 5$
média	0,80	0,78	0,77
ACE	0,71	0,72	0,66
ADA	0,80	0,83	0,83
AmpC	0,59	0,42	0,46
COMT	0,92	0,92	0,89
FGFrl	0,83	0,85	0,79
HMGR	0,97	0,96	0,96
NA	0,67	0,68	0,66
PDGFrb	0,91	0,91	0,91
PR	0,81	0,77	0,79

Na Tabela 1.14, descrevemos os resultados de AUC advindos da DeepVS para proteínas cujo o valor de AUC reportado pelo *virtual screening* efetuado usando o programa de *docking* ADV foi considerado de pobre qualidade (AUC <0,50). Para estas proteínas a AUC média da DeepVS diminui quando o valor de k_p é elevado. Estes resultados sugerem que, caso a informação estrutural advinda do complexo proteína-composto que a rede recebe como entrada tenha uma qualidade ruim, a rede neural funciona melhor sem a informação da proteína.

Este comportamento pode estar relacionado com a conformação dos compostos selecionados pelos os programas de *docking*. Por exemplo, quando os resultados de *virtual screening* provenientes do programa de *docking* reportam um valor de AUC acima de 0,70, significa que de alguma forma esse programa é bom em selecionar ligantes em detrimentos de *decoys* e provavelmente o programa deve estar escolhendo

o modo de ligação correto ou o mais próximo do cristalográfico para esse conjunto de complexos proteína-composto (Tabela 1.13). Como a informação da proteína (resíduos associados) dada como entrada para rede neural profunda está diretamente relacionada com a distância e consequentemente com a posição do átomo do composto no espaço, então estados conformacionais dos compostos que não caracterizam o seu correto modo de ligação podem influenciar de forma negativa a DeepVS durante o seu treinamento (Tabela 1.14).

Nas Tabelas 1.15 e 1.16, apresentamos os resultados para o experimento em que trocamos as proteínas alvo e simulamos rodadas de *docking* com o conjunto de ligantes e *decoys* (também conhecidos como *cross-screening*). Nós realizamos esse experimento com dois pares de proteínas selecionadas de forma aleatória a partir do banco de dados DUD, no qual cada proteína formadora do par pertence a uma classe biológica distinta. Por exemplo, a Tabela 1.15 exibe o resultados de etapas de *cross-screening* para a proteína HSP90 representante da classe biológica das "quinases" e HIVPR representante da classe biológica "outras enzimas". Na segunda Tabela 1.16 são reportados os resultados de etapas de *cross-screening* para a proteína alvo EGFr pertencente à classe das "quinases" e a proteína alvo $\text{ER}_{agonist}$ pertencente a classe "nucleares hormonais".

Tabela 1.15: Experimento de *cross-screening* para os alvos HSP90 e HIVPR.

	HIVPR		**HSP90**	
	ADV	**DeepVS**	**ADV**	**DeepVS**
HIVPR	0,72	0,88	0,52	0,75
HSP90	0,31	0,82	0,54	0,94

Tabela 1.16: Experimento de *cross-screening* para os alvos EGFr e $\text{ER}_{agonist}$.

	EGFr		**$\text{ER}_{agonist}$**	
	ADV	**DeepVS**	**ADV**	**DeepVS**
EGFr	0,58	0,86	0,33	0,80
$\text{ER}_{agonist}$	0,71	0,86	0,79	0,88

O objetivo principal dos experimentos de *cross-screening* é demonstrar que a DeepVS utiliza informações estruturais pertencentes ao complexo proteína-composto. Como demonstrado na Tabela 1.15 os valores de AUC para ambas as metodologias de *virtual screening* ADV e DeepVS diminuem em média 30,8%, quando para uma proteína alvo é usado um conjunto diferente de compostos pertencentes a uma outra proteína.

Na Tabela 1.16, podemos verificar que aparentemente existe *cross-enriquecimento* positivo entre os alvos EGFr e $\text{ER}_{agonist}$. De acordo com a primeira coluna da Tabela 1.16, o programa ADV produz melhores resultados (AUC = 0,71) ao efetuar *docking* dos ligantes e *decoys* da EGFr na $\text{ER}_{agonist}$, do que quando o *docking* é efetuado para

a própria EGFr (AUC = 0,58). A DeepVS segue o mesmo padrão do programa ADV em todos os casos.

1.4.5 Comparação com o Estado-da-Arte em *Docking-Based Virtual Screening*-DBVS

Nessa seção comparamos os resultados da DeepVS com os resultados de trabalhos previamente publicados na literatura sobre melhoramento de DBVS usando como banco de dados o DUD. Primeiro apresentamos uma comparação de forma detalhada entre o desempenho da DeepVS e dois sistemas de melhoramento de DBVS baseado em redes neurais, o *Docking Data Features Analysis* (DDFA) [2] e *Neural-Network-Based Scoring Function* (NNScore) [14,61,67]. Adicionalmente, comparamos de forma detalhada a performance da DeepVS com o programa de *docking Schrödinger's* Glide uma abordagem bem estabelecida de *docking* de uso comercial. [71, 142, 143]. Por fim, comparamos a AUC média obtida pela a DeepVS com a AUC média reportada por outros sistemas de melhoramento de DBVS e que usam o mesmo banco de dados.

O DDFA usa um conjunto de *features* previamente definidas de forma manual que são derivadas a partir da saída de programas de *docking*. Exemplos de *features* empregadas pelo DDFA são o cálculo do quociente entre o melhor *score* do composto e o número de átomos daquele composto, cálculo do coeficiente de Tanimoto dos cinco compostos mais semelhantes para uma determinada proteína e cálculo das cinco melhores poses para um determinado composto. As *features* são dadas como entrada para uma rede neural rasa (com apenas uma camada escondida) que discrimina entre ligantes ativos de *decoys* dado como entrada um complexo proteína-composto. O DDFA recebe como entrada as seis melhores poses (aquelas que possuem menor energia) advindas da saída de programas de *docking*, enquanto a DeepVS recebe como entrada os dados advindos somente da melhor pose (pose de menor energia) reportada pelo programa de *docking* aplicado.

Na Figura 1.32 comparamos a AUC reportada pela DeepVS-ADV versus a AUC da DDFA-ADV, que é a versão do DDFA que usa como entrada o resultado de *docking* proveniente do Autodock Vina. Nessa figura cada círculo representa um receptor de um total de 40 receptores pertencentes ao banco de dados DUD. A DeepVS-ADV produziu uma AUC maior que a AUC proveniente da DDFA-ADV em 27 receptores, o que representa 67,5% do banco de dados.

O DDFA-ALL é uma versão da DDFA mais robusta que usa como entrada simultaneamente a saída de três diferentes programas de *docking*: Autodock4.2 (AD4) [74], AutodockVina1.1.2 (ADV) [70] e RosettaLigand3.4 (RL) [75]. Consequentemente, o DDFA-ALL usa três vezes mais *features* de entrada para a rede neural que o DDFA-ADV. Na figura 1.33, comparamos a AUC da DeepVS versus a AUC da DDFA-ALL. Apesar da DeepVS usar como dado de entrada apenas uma pose dos compostos

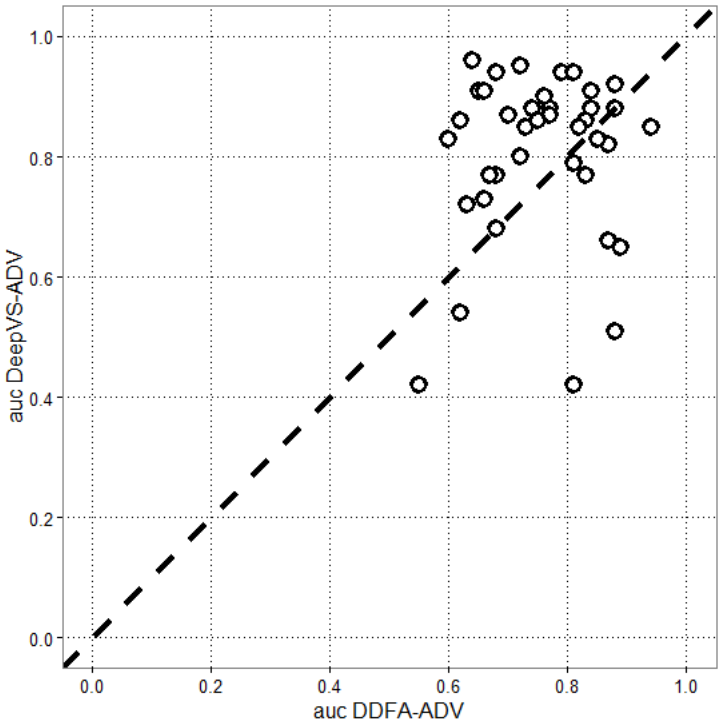


Figura 1.32: **DeepVS-ADV vs DDFA-ADV**. Os resultados de AUC obtidos pelas abordagens DeepVS-ADV e DDFA-ADV. Onde, os círculos representam cada um dos 40 receptores presentes no banco de dados DUD, a linha tracejada indica o limite onde a performance de ambos os métodos se sobrepõem.

produzida pelo programa de *docking*, A DeepVS reporta uma AUC mais alta que a DDFA-ALL para 25 receptores, representando 62,5% do banco de dados. Os resultados apresentados nessa seção indicam que a DeepVS pode ser considerada uma estratégia de reclassificação robusta.

As redes NNScore1 e NNScore2 utilizam a saída do *virtual screening* do programa AutodockVina e possuem respectivamente 5 e 15 *features* extraídas de forma manual, entre elas: (1) NNScore1 - contatos próximos (número de pares de átomos do receptor e ligante com distância de 2 Å), semi-contatos (números de pares de átomos do receptor e ligante até uma distância de 4 Å), interações eletrostáticas (cálculo da energia eletrostática da proteína e dos ligantes até uma distância de 4 Å), tipo de átomo de ligante (lista com tipos de átomos permitidos) e número de ligações rotacionais do ligante [14]; (2) NNScore2 - usa 15 *features* extraídas de forma manual, divididas em *features* relacionadas às características da função de pontuação do AutodockVina: “gauss 1”, “gauss 2” , “repulsão”, termos hidrofóbicos e termos de ligação de hidrogênio e *featuras* extraídas a partir da saída do algoritmo BINANA [144]: Número de átomos em um complexo proteína-composto até uma distância de 2,5 Å; Seleção de pares de átomos permitidos, como: (A, A), (A, C), (A, CL), (A, F), (A, FE), (A, HD), (A, MG), (A, MN), (A, N), (A, NA), (A, OA), (A, SA) e etc; caracterização eletrostática dos pares de átomos permitidos; soma da energia eletrostática entre átomos do composto e da proteína em uma distância de até 4 Å; soma de átomos que possuem contatos próximos entre uma distância de até 2,5 Å; número de contatos hidrofóbicos;

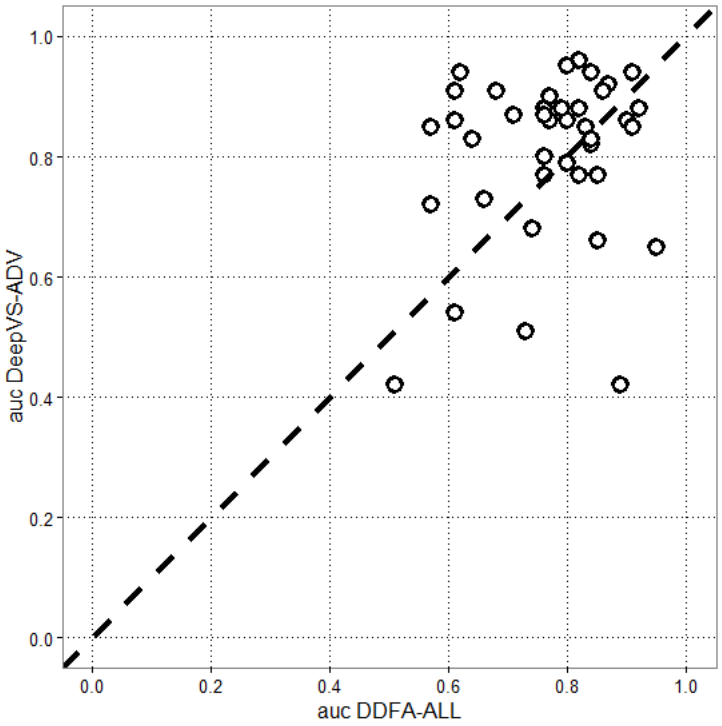


Figura 1.33: **DeepVS-ADV vs DDFA-ALL**. Os resultados de AUC obtidos pelas abordagens DeepVS-ADV e DDFA-ALL. Onde, os círculos representam cada um dos 40 receptores presentes no banco de dados DUD, a linha tracejada indica o limite onde a performance de ambos os métodos se sobrepõem.

número de ligações de hidrogênio; número de pontes salinas; números de interações π; número de ligações rotacionais do ligante; flexibilidade do sítio de ligação, cada átomo da proteína que se encontra dentro de 4,0 Å de distância para qualquer átomo de ligante é caracterizado como pertencente ou não a uma cadeia lateral; exclusão de átomos não permitidos.

Nas figuras 1.34 e 1.35 comparamos os resultados de AUC da DeepVS para cada proteína no banco de dados DUD com os resultados reportados para NNScore1 e NNScore2, respectivamente. Os resultados para NNScore1 e NNScore2 foram retirados de uma publicação recente, onde Durrant e colaboradores [67] fizeram uma comparação entre os dois sistemas e o programa de *docking* Glide. Em todos os casos os autores utilizaram em seus experimentos as proteínas e respectivos ligantes ativos do conjunto de dados DUD e um conjunto diferente e mais simples de *decoys* extraídos do *National Cancer Institute* - NCI [50]. Portanto a comparação não é totalmente justa, visto que o conjunto total de moléculas a serem ranqueadas é diferente. De qualquer forma, é interessante verificar que a DeepVS, mesmo ranqueando um conjunto mais complexo de moléculas, possui AUC melhor em 57,5% do conjunto de dados (23 proteínas) quando comparado a NNScore1 e NNScore2. Se compararmos a DeepVS apenas com a NNScore2, em dois casos (2 proteínas) ambos os sistemas reportam o mesmo resultado de AUC.

O Glide é um programa de *docking* flexível de uso comercial que foi implemen-

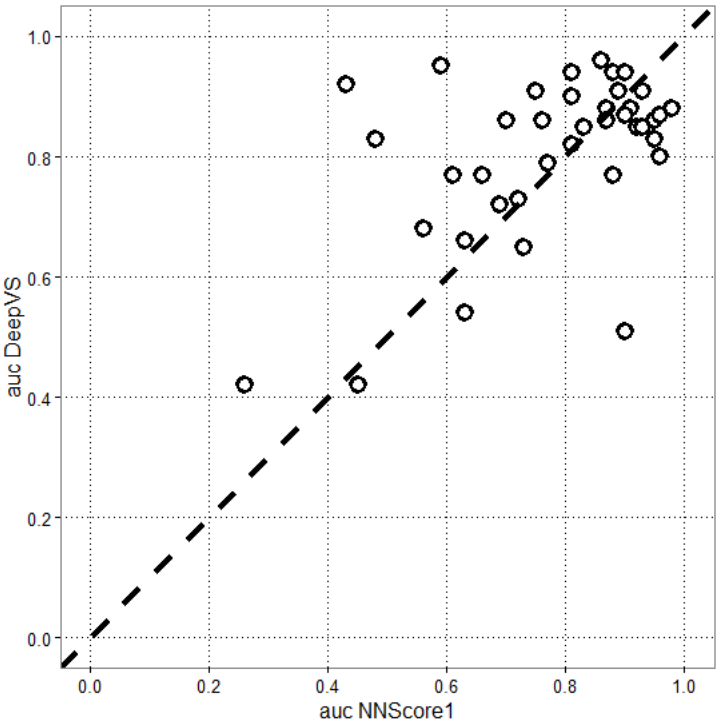


Figura 1.34: **DeepVS-ADV vs NNScore1**. Valores de AUC para a DeepVS-ADV e NNScore1, os círculos representam cada um dos 40 receptores presentes no banco de dados DUD, a linha tracejada indica o limite onde a performance de ambos os métodos se sobrepõem.

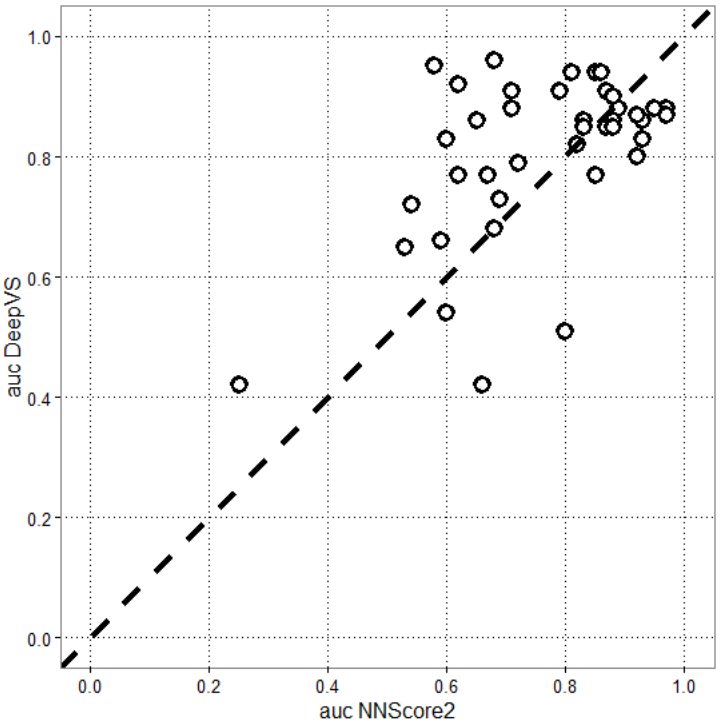


Figura 1.35: **DeepVS-ADV vs NNScore2**. Valores de AUC para a DeepVS-ADV e NNScore2. Onde os círculos representam cada um dos 40 receptores presentes no banco de dados DUD, a linha tracejada indica o limite onde a performance de ambos os métodos se sobrepõem.

tado para realizar uma busca exaustiva aproximada de posição, orientação e espaço conformacional do ligante, mantendo uma velocidade computacional adequada para rastrear grandes bibliotecas de compostos utilizando uma série de filtros hierárquicos [71, 142, 143]. É considerada uma abordagem de alta precisão e se encontra entre o estado-da-arte para DBVS [136, 145–147].

O programa de *docking* Glide é dividido de acordo com a acurácia e tempo computacional em três diferentes modos: (1) HTVS (*virtual screening* de alto rendimento) *virtual screening* para bibliotecas com até milhões de compostos; (2) SP (padrão de precisão) para *docking* para bibliotecas com até dezenas de milhares de compostos que possui um maior custo computacional; e (3) XP (precisão extra) utilizado como um filtro adicional ao processo de *virtual screening* e possui o mais alto custo computacional. Normalmente, usa-se o modo HTVS para o processo de *virtual screening* e os demais modos SP e XP como filtros. Os resultados apresentados nessa seção levam em consideração apenas o modo HTVS o qual é usualmente utilizado para *virtual screening* de compostos para um banco de dados do tamanho do DUD.

Os dados registrados para o programa Glide apresentados nesse trabalho foram retirados do trabalho elaborado por Durrant e colaboradores [67] e por essa razão usa um conjunto de *decoys* diferentes do encontrado no banco de dados DUD. Na figura 1.36 comparamos o resultado AUC de DeepVS para cada um dos 40 receptores do DUD com programa de *docking* Glide. Apesar da DeepVS ter ranqueado um conjunto mais complexo de moléculas, possui AUC melhor em 70% do conjunto de dados (28 proteínas) quando comparado com o Glide. Para os receptores AR (PDB_ID: 1xq2) e HIVRT (PDB PDB_ID: 1rt1) foi reportado o mesmo valor de AUC. A DeepVS demonstra resultado melhor que o Glide. Os nossos resultados sugerem que DeepVS pode ser utilizada como uma metodologia acessível e de baixo custo para melhoramento de *Docking-Based Virtual Screening*-DBVS.

Na Tabela 1.17 comparamos a AUC média reportada pela a DeepVS usando como entrada a saída do programa de *docking* que possui livre acesso e código aberto *Autodockvina1.1.2* com outros sistemas publicados em literatura para DBVS. Adicionalmente, comparamos os resultados da DeepVS que usa como entrada a saída do programa, também de acesso livre, Dock6.6 com as funções de pontuação Amber Score e LMOD Score.

A DeepVS-ADV produziu o melhor resultado de AUC média dentre todos os sistemas reportados, superando até mesmo os resultados de programas de *docking* de uso comercial como ICM e o Glide-SP. Além disso, a DeepVS demonstrou uma AUC média melhor do que sistemas que utilizaram intervenção humana extensa para melhoramento do resultado. Por exemplo, no caso do programa de *docking* Dock6 a DeepVS melhorou o resultado da AUC média de 0,48 para 0,74, resultado superior a métodos como a própria função de pontuação presente no programa de *docking* Dock6 Amber Score que necessita uma extensa análise manual para preparação dos compostos. Os resultados da função Amber Score e LMOD continuam inferiores aos apresentados pela DeepVS. Até onde sabemos, a AUC média para a DeepVS é o me-

Tabela 1.17: Desempenho de diferentes sistemas para DBVS usando o banco de dados DUD.

Sistemas	AUC
DeepVS-ADV	**0,81**
ICM [148] [b]	0,79
NNScore1-ADV [67] [a]	0,78
Glide SP [136] [b]	0,77
DDFA-ALL [2]	0,77
DDFA-RL [2]	0,76
NNScore2-ADV [67] [a]	0,76
DDFA-ADV [2]	0,75
DeepVS-Dock	**0,74**
DDFA-AD4 [2]	0,74
Glide HTVS [67] [a]	0,73
Surflex [136] [b]	0,72
Glide HTVS [136]	0,72
ICM [148]	0,71
RAW-ALL [2]	0,70
Autodock Vina [67] [a]	0,70
Surflex [136]	0,66
RosettaLigand [2]	0,65
Autodock Vina [2]	0,64
ICM [136]	0,63
Autodock Vina	**0,62**
FlexX [136]	0,61
Autodock4.2 [2]	0,60
Dock6(Amber Score/LMOD Score) [140][b]	0,60
PhDOCK [136]	0,59
Dock4.0 [136]	0,55
Dock6.6	**0,48**

[a] Usou um conjunto de *decoys* diferentes.

[b] Resultado melhorado a partir de conhecimento especializado.

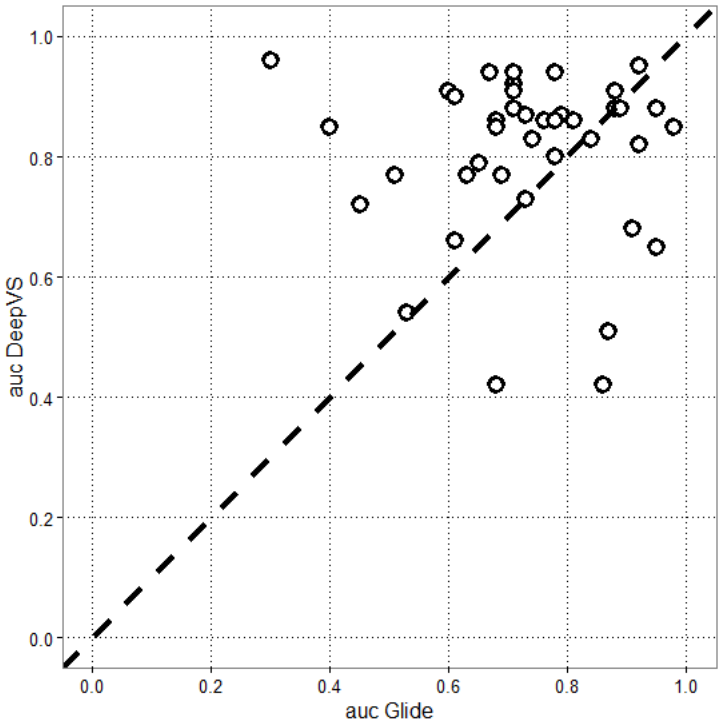


Figura 1.36: **DeepVS-ADV vs Glide-HTVS**. Valores de AUC para a DeepVS-ADV e Glide-HTVS, os círculos representam cada um dos 40 receptores presentes no banco de dados DUD, a linha tracejada indica o limite onde a performance de ambos os métodos se sobrepõem.

lhor valor reportado até o presente momento para *Docking-based Virtual Screening* usando como conjunto de dados os 40 receptores do banco de dados DUD.

1.4.6 Validação da DeepVS usando um subconjunto de proteínas do DUD-E

A robustez da DeepVS foi avaliada por meio de experimentos adicionais usando o DUD-E [119] como um conjunto de dados externo. Para os nossos experimentos selecionamos aleatoriamente 44 receptores do DUD-E sendo (a) existem receptores das oito classes biológicas existentes no DUD-E; e (b) nenhum dos 44 receptores está entre as 40 receptores do DUD. Nos experimentos apresentados nessa seção, o programa Autodockvina (ADV) foi utilizado na fase de *docking*.

Dois experimentos principais foram executados: (1) No primeiro, a DeepVS-ADV foi treinada utilizando as 40 proteínas do DUD, e em seguida foi utilizada para executar um experimento de *virtual screening* para cada um dos 44 receptores do DUD-E. A DeepVS-ADV treinada com o banco de dados DUD é referenciada nesse seção como DeepVS-ADV (DUD); (2) No segundo experimento, utilizamos os 44 receptores do DUD-E para testar a DeepVS-ADV usando a metodologia de validação cruzada *leave-one-out*, no qual uma proteína é escolhida para teste e as demais são utilizadas para treino. A DeepVS-ADV treinada utilizando validação cruzada é referenciada nessa se-

ção como DeepVS-ADV (VC). Nesse experimento não retiramos do conjunto de treino as proteínas pertencentes a mesma família da proteína teste. Optamos por essa opção em virtude do banco de dados DUD-E possuir uma assimetria elevada no número de proteínas em cada família, por exemplo as famílias Citocromo P450, Canal iônico e GPCR possuem 2, 2 e 3 proteínas respectivamente, enquanto as famílias Outras Enzimas, Quínases e Proteases possem respectivamente 35, 26 e 15 proteínas. Adicionalmente, outros trabalhos que reportam resultados utilizando o DUD-E não removeram do conjunto de treino proteínas da mesma família de proteínas teste [97, 149]. Em todos os experimentos foram mantidos os mesmos hiperparâmetros utilizados nos experimentos com o conjunto de dados DUD: d^{atm} = 200; d^{amino} = 200; d^{chrg} = 200; d^{dist} = 200; cf = 400; h = 50; λ = 0,075; k_c = 6; k_p = 2.

Na Tabela 1.18 apresentamos os resultados para ambos os experimentos envolvendo o DUD-E. A DeepVS-ADV (DUD) melhorou o resultado de AUC do ADV em 63,63% dos receptores atingindo uma AUC média de 0,76 contra 0,73 do ADV. No entanto, os resultados de $ef_{2\%}$ e ef_{max} pioraram.

Tabela 1.18: Valores de AUC, fator de enriquecimento ($ef_{2\%}$, $ef_{20\%}$ e ef_{max}) para AutodockVina, DeepVS-ADV treinada usando o DUD, DeepVS-ADV treina usando validação cruzada referentes à 44 proteínas do DUD-E. Valores em negrito indicam o maior valor de AUC computado.

	ADV				**DeepVS-ADV (DUD)**				**DeepVS-ADV (VC)**			
	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	auc	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	auc	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	auc
média	24,1	6,7	2,5	0,73	13,3	4,3	2,6	0,76	51,6	25,3	4,4	**0,93**
AA2AR	1,2	0,4	0,9	0,56	2,8	1,3	2,5	0,74	20,4	11,2	4,0	**0,88**
ADRB1	65,1	3,4	2,3	0,73	2,8	2,6	2,3	0,74	65,1	41,5	5,0	**0,99**
ADBR2	59,9	12,8	2,8	0,78	2,8	2,2	2,4	0,75	65,9	45,3	5,0	**1,00**
AKT2	11,5	6,6	3,2	0,79	29,7	5,5	2,5	0,78	28,4	18,8	4,4	**0,93**
AOFB	9,3	5,3	2,1	0,72	57,6	0,8	1,5	0,54	57,6	7,4	3,4	**0,81**
BACE1	5,3	2,9	2,0	0,63	12,0	9,9	3,6	0,85	40,6	28,1	4,6	**0,95**
CASP3	23,5	2,1	1,6	0,61	7,3	0,5	1,7	0,65	32,1	15,9	4,5	**0,93**
CP2C9	5,3	2,9	2,0	0,63	3,3	2,9	2,7	0,75	60,5	26,7	4,5	**0,95**
CP3A4	23,5	2,1	1,6	0,61	35,2	4,4	3,0	0,81	70,4	29,7	4,7	**0,96**
CXCR4	1,5	0,0	0,8	0,59	8,6	3,7	2,0	0,62	9,2	5,0	3,8	**0,85**
DEF	56,8	4,4	2,3	0,69	2,6	0,0	2,2	0,70	56,8	29,9	4,7	**0,97**
DRD3	13,3	3,2	2,5	0,74	2,3	1,7	2,1	0,72	72,0	20,0	4,4	**0,92**
ESR2	30,3	9,9	3,1	0,79	35,0	12,7	4.3	0.91	56,0	27,8	4,5	**0,93**
FA7	23,2	10,5	4,4	0,91	27,9	9,2	3,6	0,86	55,8	15,8	4,4	**0,93**
FABP4	35,7	15,9	3,0	0,78	14,9	3,2	3,7	0,82	59,5	17,0	4,7	**0,93**
FAK1	54,5	10,5	2,8	0,81	31,8	14,0	4,7	0,93	54,5	39.5	4.9	**0.99**
FKB1A	26,6	4,5	2,1	0,75	3,7	1,4	3,4	0,82	11,8	4,5	2,9	**0,77**
FNTA	5,1	4,1	2,5	0,79	1,5	0,1	0,5	0,64	5,7	5,5	4,3	**0,96**
GLCM	1,2	0,0	1,1	0,52	5,0	4,4	1,1	**0,72**	7,9	3,7	2,3	0,67
GRIA2	20,7	5,4	2,7	0,74	1,6	0,3	1,2	0,64	75,9	32,9	4,6	**0,94**
GRIK1	6,2	3,5	2,0	0,60	26,3	11,9	2,7	0,76	65,8	35,6	4,9	**0,98**
HXK4	3,2	1,6	0,9	0,57	6,2	3,8	2,5	0,72	52,0	15,2	3,8	**0,88**
ITAL	1,6	0,7	1,0	0,62	5,4	3,3	3,4	0,82	44,6	15,5	4,0	**0,89**
JAK2	61,7	9,3	2,9	0,78	61,7	7,5	3,4	0,81	61,7	27,6	4,7	**0,95**
KIF11	60,0	19,4	3,7	**0,87**	15,0	1,7	2,5	0,74	31,8	16,0	3,8	0,86
LKHA4	13,0	10,0	4,2	0,89	3,8	2,9	2,3	0,72	56,3	34,3	4,9	**0,99**
MAPK2	22,5	13,4	3,9	0,89	5,0	2,2	1,5	0,68	61,9	38,1	4,9	**0,98**
MK01	7,5	5,0	4,1	0,86	2,9	0,6	2,3	0,79	58,6	28,3	4,7	**0,95**
MK10	21,5	5,3	2,4	0,76	12,3	4,8	3,4	0,83	64,5	28,9	4,8	**0,97**
MMP13	8,2	3,6	2,2	0,65	4,2	3,3	1,4	0,54	65,9	33,3	4,8	**0,97**
MP2K1	1,5	0,4	1,0	0,61	11,4	1,2	3,2	0,79	68,3	32,7	4,8	**0,97**
NOS1	2,9	2,0	1,4	0,59	6,6	4,5	2,1	0,65	81,5	13,5	3,5	**0,83**
PA2GA	2,4	1,0	1,1	0,62	4,1	3,0	1,4	0,63	26,5	17,7	3,8	**0,88**
PLK1	2,7	2,3	2,1	0,63	2,1	0,0	1,9	0,70	64,5	28,1	4,6	**0,94**
PPARA	8,2	6,6	3,8	0,86	11,3	9,5	4,2	0,89	52,9	42,0	5,0	**0,99**
PPARD	2,8	1,3	2,4	0,76	9,8	9,2	4,1	0,88	51,9	44,8	5,0	**0,99**
PTN1	56,7	15,8	3,6	0,83	4,1	2,7	3,2	0,83	56,7	28,6	4,9	**0,97**
PYRD	33,4	13,5	3,3	0,86	2,3	0,0	1,9	0,70	59,1	21,6	4,2	**0,92**
RENI	67,9	2,9	2,1	0,65	6,0	2,5	3,0	0,92	67,9	30,8	5,0	**0,98**
ROCK1	21,3	4,5	1,6	0,72	12,8	3,5	1,7	0,70	64,0	35,5	5,0	**0,99**
THB	73,2	18,9	3,4	0,82	5,9	2,9	2,7	0,79	61,0	26,2	4,7	**0,96**
TRYB1	39,5	5,7	3,0	0,78	11,1	10,1	3,3	0,81	52,6	20,9	4,6	**0,95**
WEE1	61,3	35,3	4,4	0,95	61,3	16,7	5,0	0,96	61,3	47,5	5,0	**1,00**
XIAP	3,4	2,5	1,8	0,67	3,5	1,5	2,3	0,75	26,2	9,5	4,3	**0,91**

Por outro lado, os experimentos de validação cruzada da DeepVS-ADV (VC) resultaram em valores de AUC e fator de enriquecimento bastante elevados. A AUC média da DeepVS-ADV (VC) (0,93) representa um aumento de 27,4% no valor da AUC do ADV (0,73). Enquanto que o $ef_{2\%}$ da DeepVS-ADV (VC) (25,3) consiste num aumento de 277% com relação ao valor do $ef_{2\%}$ da ADV (6,7%).

A diferença de performance entre a DeepVS-ADV (DUD) e o ADV para o DUD-E não é tão expressiva quanto para o DUD. Acreditamos que esse comportamento está associado ao fato do conjunto de treinamento (DUD) possuir algumas características diferentes das encontradas no conjunto de teste (DUD-E) como por exemplo: (1) os ligantes gerados para o DUD foram provenientes do banco de dados ZINC [115] enquanto que os ligantes gerados para o DUD-E foram retirados do banco de dados ChEMBL [49]; (2) no DUD ocorre a distribuição aleatória de ligantes com quimiotipos predominantes enquanto que no DUD-E os quimiotipos predominantes foram separados em subconjuntos de forma a evitar "bias análogas"; (3) no banco de dados DUD-E houve uma realocação de proteínas em famílias e adição de três novas famílias (GPCR, Canais lônicos e Citocromo P450); (4) no banco de dados DUD-E parte dos *decoys* possuem inatividade comprovada em ensaios laboratoriais diferentemente do DUD, no qual todos os seus *decoys* foram gerados utilizando o banco de dados ZINC. Dessa forma, os bancos de dados DUD e DUD-E possuem diferentes estratégias de geração de *decoys*.

Os resultados para a DeepVS (VC) confirmam a nossa hipótese sobre a alteração do comportamento da DeepVS está relacionada a diferenças entre os bancos de dados DUD e DUD-E. Quando a DeepVS é treinada utilizando validação cruzada com o DUD-E ela demonstra um desempenho expressivamente superior à DeepVS treinada utilizando o banco de dados DUD.

Os resultados expressivos apresentados pela DeepVS (VC) para o banco de dados DUD-E corroboram com os resultados de outras metodologias de melhoramento de *virtual screening* baseado em estrutura (DBVS) para o DUD-E descritas na literatura. Na Tabela 1.19 comparamos os nossos resultados com resultados de dois métodos de DBVS baseados em *deep learning* publicados recentemente: cmpds ECFP + LR [97] e AtomNet [149]. Apesar dos autores do cmpds ECFP + LR e do AtomNet utilizarem em seus experimentos as 102 proteínas DUD-E, é importante enfatizar que os autores não utilizam um conjunto de validação, o treinamento da rede é feito dividindo 70% para treino e 30% do conjunto de dados para teste. Os autores das respectivas publicações não informam quais proteínas foram utilizadas para treino e quais proteínas foram utilizada para teste. Além disso, os resultados de AUC para cada uma das 102 proteínas do DUD-E não são reportados assim como os valores de fator de enriquecimento. Dessa forma, a comparação não é totalmente justa, visto que: (1) não é totalmente claro se há ou não a presença de *overfitting* no conjunto de teste dado que os autores não informam o uso de um conjunto de validação; (2) em cada iteração da validação cruzada, nosso conjunto de treino contém apenas 43 proteínas, enquanto que o conjunto de treino usado nos trabalhos supracitados contém 70 proteínas. De

qualquer forma, é interessante verificar que, mesmo utilizando um conjunto de treino menor e um método mais rigoroso de avaliação (validação cruzada *leave-one-out*), a DeepVS possui um resultado superior quando comparada com os métodos cmpds ECFP + LR e AtomNet.

Tabela 1.19: Resultados do desempenho de metodologias para melhoramento de DBVS para o banco de dados DUD-E.

Banco de dados	**Método**	**AUC média**
DUD-E (44 proteínas)	DeepVS [a]	**0,931**
DUD-E (102 proteínas)	cmpds ECFP + LR [b]	0,904
DUD-E (102 proteínas)	AtomNet [b]	0,855

[a] Validação cruzada do tipo *leave-one-out*.

[b] 70% do banco de dados para treino e 30% para teste.

Acreditamos que com a adição das 58 proteínas restantes do banco de dados DUD-E o nosso resultado se mantenha igual, ou melhor, ao resultado para 44 proteínas (UAC média de 0,931). Isso por que, é amplamente conhecido que em metodologias de *deep learning* o resultado tende a melhorar a medida que mais dados são acrescentados ao treinamento.

1.5 Perspectivas

Deep leraning muitas vezes é pensada como uma caixa preta, na qual não se sabe ao certo o que a rede está aprendendo durante a etapa de treinamento. Porém, recentemente alguns trabalhos estão mudando essa perspectiva sobre *deep learning*. Por exemplo, o trabalho desenvolvido por Mikolov *et al.* [110] de interpretação do modelo para *embeddings* de palavras (*word embeddings*). Diante dessa possibilidade, pretendemos realizar estudos voltados para identificação de padrões biológicos aprendidos pela DeepVS que são considerados relevantes para distinguir entre ligantes e *decoys*. Pretendemos também realizar experimentos adicionais para testar a hipótese de que a rede pode estar aprendendo a distinguir entre ligantes e *decoys* utilizando apenas a informação proveniente dos compostos, o que caracterizaria um processo de *virtual screening* baseado no ligante.

Um dos grandes diferenciais da abordagem de *deep learning* é melhorar o seu desempenho mediante a adição de mais dados. Por isso, futuramente iremos treinar a DeepVS com bancos de dados maiores.

Os conceitos introduzidos nesse trabalho sobre *embeddings* de átomos e aminoácidos (resíduos) possuem potencial de aplicabilidade em outros problemas da biologia estrutural, como por exemplo para modelagem de estruturas e desenho de fármacos. Dessa forma, em trabalhos futuros planejamos aplicar essa ideia para predição de estrutura da proteína a partir de sua sequência de aminoácidos.

2 PARTE II

2.1 Estudo de Caso: Cruzaína

2.1.1 Introdução

2.1.1.1 Doenças Tropicais Negligenciadas

Doenças Tropicais Negligenciadas (DTN) é um termo usado para descrever um grupo heterogêneo de doenças que não compartilham patogenia, mas compartilham um estado social de “extrema pobreza”. Isso porque as pessoas afetadas por DTN em sua grande maioria são pobres, vivem em áreas rurais, favelas urbanas, regiões remotas ou zonas de conflito de países em desenvolvimento, onde o acesso à saúde e educação é precário [150, 151].

DTN estão ligadas diretamente à falta de tratamento, de preços de fármacos acessíveis e de medicamentos de uso fácil. Existe desinteresse da indústria farmacêutica em financiar tratamento para esse tipo de doenças uma vez que não há incentivo público ou potencial de mercado [152, 153]. Estas indústrias, por sua vez, investem quase que exclusivamente em medicamentos que sejam facilmente comercializáveis ou compostos lucrativos voltados principalmente para dor, câncer e complicações cardíacas [153]. Em um levantamento nos anos de 1975 a 1999 Trouiller *et al.* [154] identificaram 1.393 novos medicamentos comercializados, onde desses apenas 16% eram destinados a doenças negligenciadas e mais de dois terços destes eram versões modificadas de medicamentos existentes, embora doenças negligenciadas correspondem a mais de 10% das doenças globais.

Segundo a Organização Mundial de Saúde [151], das 17 doenças que são consideradas DTN, três são causadas por protozoários: Doenças de Chagas (*Trypanosoma cruzi*), tripanossomíase africana humana conhecida como Doença do Sono (*Trypanosoma Brucei*) e Leishmaniose (*Leishmania spp.*).

A definição ou classificação de protozoários é complexa e um pouco controversa. Entretanto se generalizarmos, podemos definir protozoários como organismos unicelulares eucariontes [155–157].

Segundo a revisão feita por Imam [156], existem cerca de 200 mil espécies descritas de protozoários e em torno de 10 mil parasitam invertebrados e vertebrados. Em

particular, esses parasitas são responsáveis por grande parte das doenças comuns e devastadoras que afetam humanos e animais domésticos [158] .

Por acometer grande parte da população do mundo em estado de “vulnerabilidade social” a maioria das doenças ligadas a protozoários parasitas não são atrativas para indústria farmacêutica, resultando em tratamentos não preventivos e ineficientes por meio de poucos fármacos anti-protozoários, que muitas vezes estão associados a um alto grau de morbidade, toxicidade e resistência ao medicamento [152, 153, 158].

2.1.1.2 Doença de Chagas

A doença de Chagas ou tripanossomíase americana é causada pelo protozoário flagelado *Trypanosoma cruzi*. Considerada uma doença negligenciada, segundo um estudo realizado pela a organização mundial de saúde até o ano de 2010 essa doença afetou cerca de 8-10 milhões de pessoas no mundo [159]. É uma grave endemia na América Latina onde 25 milhões de pessoas vivem em área de risco. Ultimamente vem se tornando um grave problema de saúde nos Estados Unidos, na Europa, em alguns países da Asia e na Austrália [150, 160, 161] (Figura 2.1).

Os tipos de transmissão da doença de Chagas são divididos em dois: (1) transmissão comum ou predominante e (2) transmissão incomum ou acidental [160, 162]. A transmissão comum ou predominante normalmente ocorre pelas fezes ou urina de triatomíneos hematófagos infectados, também conhecidos como barbeiros e por transfusão de sangue contaminado [160, 162, 163]. A transmissão incomum ou acidental ocorre por transplante de órgãos, ingestão de comida contaminada, pelo leite materno e transmissão congênita quando o parasita ultrapassa a placenta [160, 163].

O agente etiológico da doença de Chagas, o *Trypanosoma cruzi*, pode se manifestar em três formas em seu ciclo de vida, as quais podem ser identificadas morfologicamente pela a posição do cinetoplasto: epimastigota, o cinetoplasto e a bolsa flagelar estão em posição anterior ao núcleo, forma de multiplicação no interior do intestino do vetor ou em cultura; tripomastigota, o cinetoplasto está situado na região posterior do flagelo, em posição terminal ou subterminal, e o flagelo emerge da chamada bolsa flagelar, de localização próxima ao cinetoplasto, constitui a fase extracelular caracterizada por ser a fase infectante do parasita; amastigota constitui-se por organismos arredondados que apresentam flagelos inconspícuos, presente na fase intracelular infectante [164, 165].

Figura 2.1: **Mapa de distribuição geográfica global da Doença de Chagas.** Círculos em vermelho representam transmissões ocorridas sem a presença do vetor triatomíneo, como por exemplo por transfusão sanguínea; círculos em azul representam transmissões ocorridas com a presença do vetor triatomíneo porém de forma acidental, como por exemplo a ingestão de comida contaminada pelos restos mortais do triatomíneo; e círculos em amarelo representam transmissões diretamente ocorridas pelo vetor triatomíneo. Fonte: Ribeiro *et. al.* (2012) adaptado por Pereira J.C..

A figura 2.2 resume o ciclo de vida do *Trypanosoma cruzi* em sua forma mais comum de transmissão, através da fezes ou urina de triatomíneos: (1) O triatomíneo ou barbeiro ao se alimentar do sangue de um vertebrado qualquer, libera suas fezes contendo a forma tripomastigota do *Trypanosoma cruzi* próximo à área da picada. Esta forma do parasita penetra no organismo do hospedeiro através da ferida causada pela picada ou por contato com mucosas; (2) Dentro do hospedeiro, as formas tripomastigotas invadem as células próximas ao local de inoculação, no qual se diferenciam em sua fase intracelular amastigota que possuem forma oval e ausência de flagelo; (3) As formas amastigotas multiplicam-se rapidamente, por divisão binária, causando o rompimento celular e a infestação do protozoário na corrente sanguínea; (4) Neste estágio, o protozoário assume sua forma tripomastigota, se espalha e infecta mais células causando lesões em tecidos, como as musculares. As formas tripomastigotas encontradas na corrente sanguínea transformam-se em amastigotas intracelulares em locais de novas infecções. Manifestações clínicas podem ocorrer durante esse estágio. As formas tripomastigotas da corrente sanguínea não se replicam como ocorre em outras espécies do gênero. A replicação é retomada quando os parasitas invadem outras células ou são ingeridos por outro vetor; (5) O triatomíneo pode se contaminar consumindo sangue humano ou de outro vertebrado infectado, pela forma tripomastigota do *Trypanosoma cruzi*, através da sua picada; (6) As formas tripomastigotas ingeridas pelo barbeiro se transformam em epimastigotas no intestino delgado do vetor; (7) As formas epimastigotas se multiplicam no intestino delgado; (8) As formas epimastigotas se diferenciam no intestino grosso na forma de tripomastigotas infectantes. Essa forma é liberada pelo triatomíneo por meio de fezes ou urina, renovando o ciclo de vida do protozoário [162–167].

Outra forma comum de transmissão é por transfusão sanguínea e ocorre principalmente em locais onde o vetor triatomíneo foi controlado [160]. Um dos maiores problemas relacionados à transmissão via transfusão sanguínea é a alta incidência de reações sorológicas inconclusivas entre doadores de sangue [160, 168]. Estudos desenvolvidos por Picka *et al.* [168] sugerem que o melhor modo de confirmar sorologia positiva em indivíduos que obtiveram dois ou mais resultados inconclusivos é por meio do TESA-cruzi.

A doença de Chagas pode apresentar dois estágios clínicos: (1) agudo, ocorre imediatamente após a infecção, caracterizado por uma forte evidência de imunidade no paciente, mas o mesmo continua infectado. Nesse estágio a doença é assintomática em adultos, porém é sintomática em crianças em sua primeira década de vida e pode evoluir para óbito por complicações cardíacas ou processos inflamatórios [163]. (2) crônico, descrito por Carlos Chagas em 1916 [169] como “forma crônica intermitente”, a fase crônica da doença de Chagas é o período mais longo da doença, que pode ser inicialmente assintomático sem a presença de anormalidades eletrocardiográficas ou radiológicas no coração, esôfago ou cólon. Com o tempo pode evoluir para sintomas cardíacos ou digestivos [170].

O tratamento para doença de Chagas não é eficaz principalmente para fase crô-

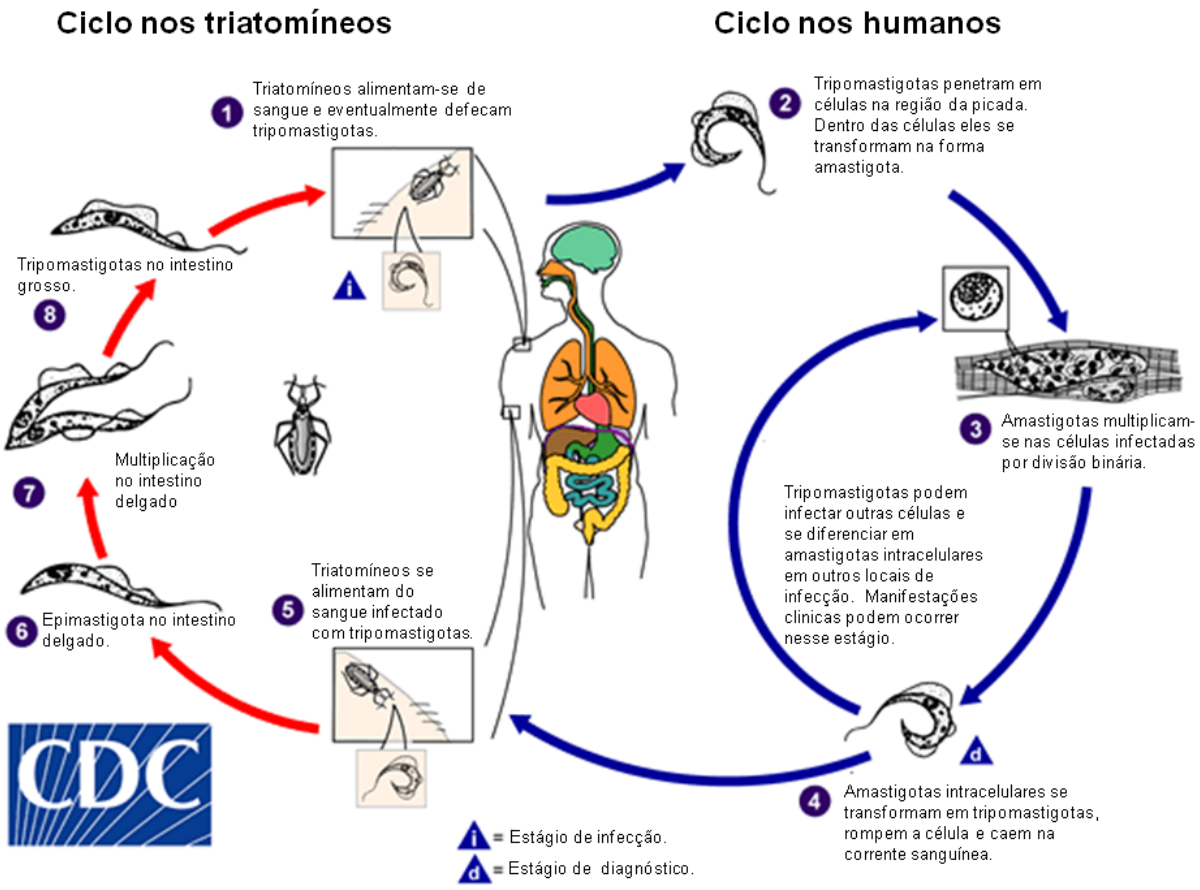
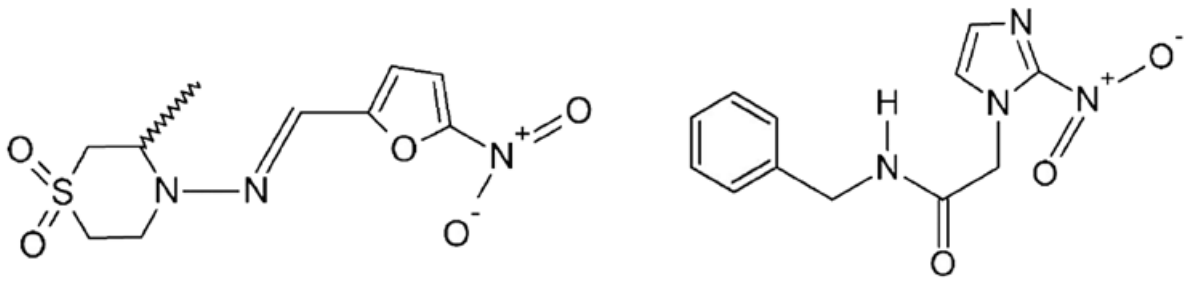


Figura 2.2: **Ciclo de vida do *Trypanosoma cruzi***. Fonte: CDN (*Centers for Disease Control and Prevention*) 2016, disponível em: *https://www.cdc.gov/dpdx/trypanosomiasisamerican/modules/amertryp_lifecycle.gif*, adaptado por Pereira J.C..

nica e muitas vezes os medicamentos disponibilizados possuem efeitos secundários que variam desde reações de hipersensibilidade à depressão da medula óssea e polineuropatia periférica [150, 160, 171].

Os fármacos disponíveis para tratamento quimioterápicos são compostos por nitrofuranos como nifurtimox (Lampit®, Bayer) e benzonidazol (Radanil®/ Rochagan®, Roche), normalmente são utilizados combinados ou separados para curar a fase aguda da doença (Figura 2.4) [172]. Em um estudo controlado, foi demonstrado que benzonidazol é mais eficiente que nifurtimox para o tratamento da doença em fase crônica [173]. O nifurtimox possui venda proibida pela Anvisa no Brasil por razão de toxicidade [174].

Os principais efeitos nocivos do nifurtimox são anorexia; perda de peso; alterações psicológicas como excitabilidade ou sonolência; problemas digestivos tais como como náuseas, vômitos, cólicas intestinais e diarreia. Os efeitos colaterais mais graves apresentados pelo benzonidazol são agranulocitose, iniciada por neutropenia, dor de garganta, febre e septicemia; e púrpura trombocitopênica, caracterizada pela redução de plaquetas, bolhas hemorrágicas e hemorragia mucosa. Alguns dos outros efeitos



nifurtimox benzonidazol

Figura 2.3: **Representação das estruturas químicas dos fármacos nifurtimox e benzonidazol.**

causados por benzonidazol são dermatites cutâneas, dores articular e muscular, depressão da medula óssea e polineurite dos nervos periféricos [163]. Apesar desses medicamentos apresentarem efeitos colaterais severos e nocivos possuem eficácia limitada, em torno de 20%, para o tratamento da fase crônica da doença [175].

2.1.1.3 Proteases

Proteases são também conhecidas como peptidases ou enzimas proteolíticas e formam um dos maiores e mais importantes grupos de enzimas. Elas são responsáveis pela execução da proteólise, ou seja, por catalizar a hidrólise de ligações peptídicas em proteínas ou peptídeos [176, 177].

As proteases estão envolvidas em diversos processos fisiológicos importantes tais como: fertilização; coagulação do sangue; crescimento e diferenciação celular; sinalização celular; apoptose; e resposta imune [176]. Podem atuar como enzimas que medeiam hidrólise inespecífica em proteínas como também em processos extremamente específicos, clivando o substrato de forma seletiva e eficiente descadeando mecanismos específicos que influenciam processos biológicos. Inibidores de proteases podem ser considerados como importantes para intervenção terapêutica de vários tipos de doenças em seus diversos estágios, onde as proteases são responsáveis por várias funções dentre elas a invasão celular do parasita no hospedeiro [176–178].

A classificação das proteases ocorre de acordo com o grupo catalítico no sítio ativo. Dessa forma, as proteases podem ser classificadas como: serina, treonina, cisteína, aspartato, glutamato ou zinco em metaloproteases [178]. Podem ser classificadas também pela reação que elas promovem, como: exopeptidases e endopeptidases. Exopeptidases clivam um ou mais resíduos de aminoácidos tanto da extremidade N-terminal quando da extremidade C-terminal do peptídeo. Endopeptidases, são res-

ponsáveis por clivarem uma ligação peptídica interna [178].

A superfície da protease possui sub-sítios (cavidades) ou regiões do sítio ativo capazes de acomodar uma única cadeia lateral do resíduo do substrato. Segundo a nomenclatura desenvolvida por Schechter e Berger [179] os sub-sítios são enumerados S1-Sn em direção ao N-terminal do substrato e S1'- Sn' seguindo a direção C-terminal, começando pelos sítios de cada lado onde ocorre a clivagem da ligação. Os resíduos do substrato acomodados nesse sub-sítios são enumerados P1 - Pn e P1'- Pn respectivamente ao sentido C-terminal e N-terminal da ligação petídica do substrato. Essa estrutura é conhecida como ocupação de sub-sítios intrínsecos e determina quais resíduos do substrato podem interagir com sítios de ligação de substratos específicos da protease, o que determina a especificidade do substrato da protease [180] (Figura).

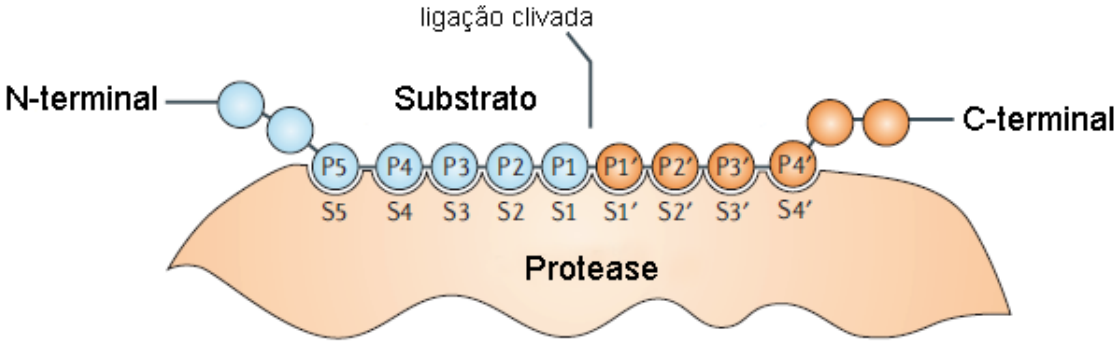


Figura 2.4: **Modelo de interação proteína-substrato para protease.** Sub-sítios na protease representados por S1-S5 e S1'-S4'. Os resíduos do substratos são representados por P1-P5 e P1'-P4'. A clivagem do substrato ocorre entre os resíduos P1 e P1'. Fonte Turk & Boris (2006), adaptado por Pereira J.C.

2.1.1.4 Cruzaína

A enzima Cruzaína também conhecida como cruzipaína (GP 57/51) membro da família de papaína C1 é considerada a principal e mais abundante cisteíno-protease do *Trypanosoma cruzi*, também reconhecida como antígeno imunodominante que ocorre durante a infecção em humanos [181–183].

A cisteíno-protease Cruzaína é expressa como uma mistura complexa de isoformas durante as principais fases de desenvolvimento do parasita sendo responsável pela nutrição e desenvolvimento do parasita, evasão do sistema imune, diferenciação celular do parasita e invasão celular no hospedeiro [184].

A estrutura da Cruzaína é composta por uma cadeia polipeptídica contendo 215 resíduos de aminoácidos formada por dois domínios, no qual um deles é fundamentalmente composto por hélices-α (domínio L) e outro é caracterizado por extensas interações de folhas-β antiparalelas (domínio R) [185] (Figura 2.5). O sítio ativo ao substrato é encontrado entre os dois domínios e é composto pela tríade catalítica Cys25, His162 e Asn182 (númeração da cruzaína) [147, 186].

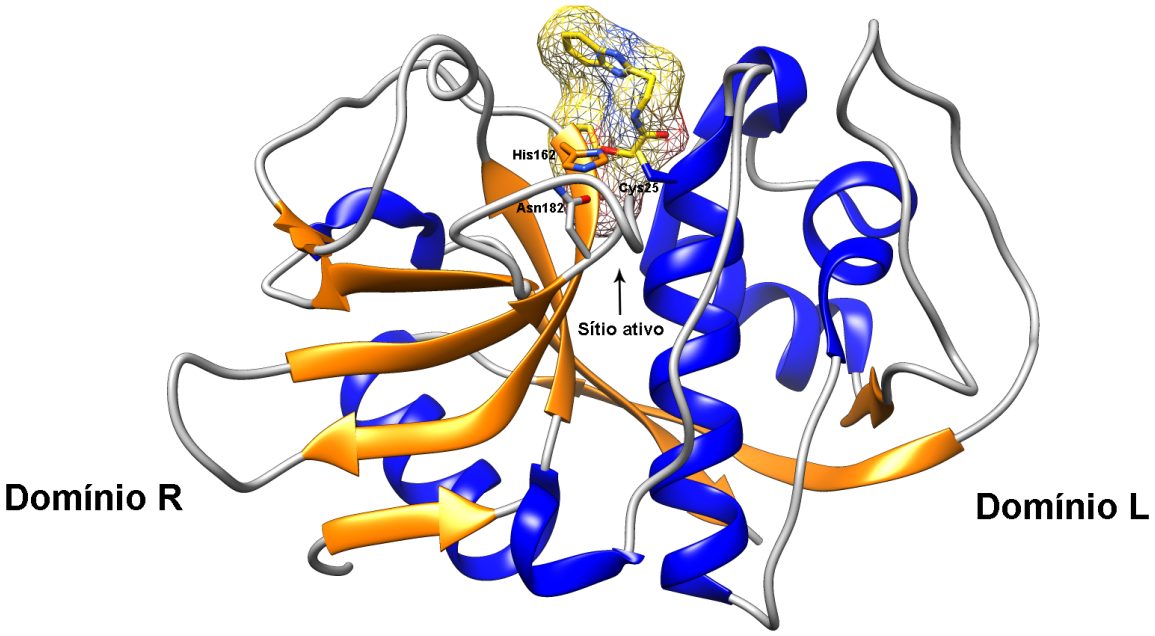


Figura 2.5: **Representação da estrutura da Cruzaína.** Domínio R corresponde a folhas-β antiparalelas (laranja) e domínio L corresponde a α-hélices (azul). O sítio ativo localizado entre os dois domínios. PDB_ID: 3kku, ligante B95.

O sítio ativo da Cruzaína é formado por quatro sub-sítios de ligação o S1, S2, S3 e S'. O sub-sítio S2 o principal responsável pela especificidade da proteína, no qual em pH entorno de 5,5 possui maior preferencia por Phe em detrimento de Arg na posição P2 [176]. Os sub-sítios S1, S3 e S' são menos definidos e possuem mais acessibilidade ao solvente, o que explica as interações dos resíduos P1 e P3 parecerem menos discriminatórias [187].

O sub-sítio S2 possui um caráter hidrofóbico e é delimitado pelos os resíduos Met68, Ala138, Leu160, Gly163 e localizado na base do sub-sítio o Glu208 [185–187]. O resíduo Glu208 pode se movimentar dentro do sub-sítio S2 adotando diferentes conformações com relação ao substrato. Em pH neutro a porção do ácido carboxílico do Glu208 oscila no sub-sítio S2 e em pH ácido é movida para longe do sub-sítio [176, 185, 187]. As interações no sub-sítio S2 em pH ácido possuem uma preferência por resíduos alifáticos ou com grupo lateral aromático. Porém, em pH neutro, por conta da flexibilidade do resíduo Glu208, também pode interagir com aminas da posição P2 do substrato [187].

Os principais inibidores para a enzima Cruzaína são baseados em vinilsulfonas, tetrafluorometilcetonas e diazometilcetonas que interferem no ciclo intracelular *in vitro* do *T. cruzi* [176, 185, 187]. Todos possuem o mesmo mecanismo, uma "ogiva" eletrofílica que pode inativar covalentemente e irreversivelmente a Cruzaína como resultado do ataque nucleofílico pela cisteína presente no sítio ativo (Figura 2.6) [188].

O inibidor *N*-Mpip-Phe-Hph-VSPh também conhecido como K11777 é uma vinilsulfona que demonstrou atividade em ensaios *in vitro* e *in vivo* para o agente etio-

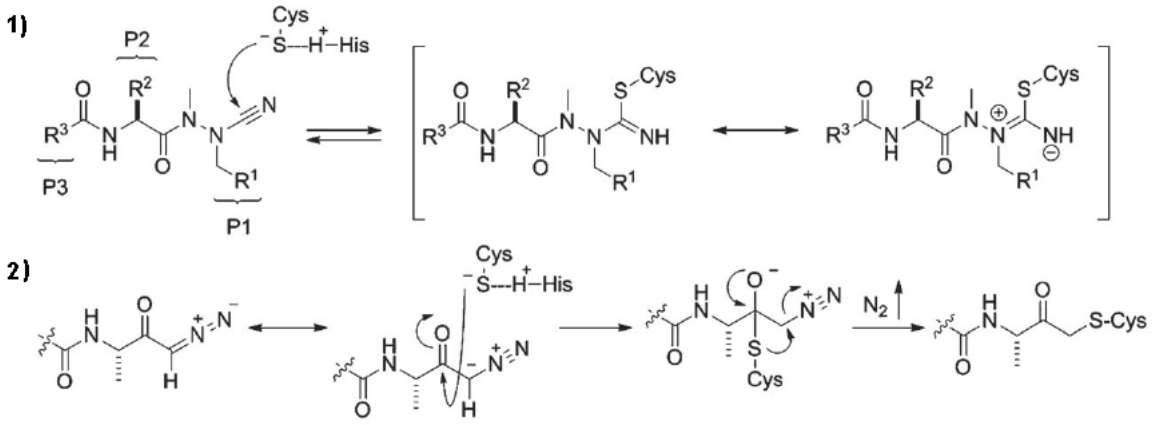


Figura 2.6: **Representação do mecanismo utilizado para inibir cisteína proteases por (1) azanitrilos e (2) diazometilcetonas.** Fonte Yang *et al.* (2012).

lógico da doença de Chagas o *Trypanosoma cruzi*, apresentando elevada afinadade para a Cruzaína, o que seria uma alternativa para o tratamento da fase crônica da doença [176, 188]. O K11777 também demonstrou atividade para o *Trypanosoma brucei* apresentando afinidade com a Rodesiana. Apesar do avanços com o inibidor K11777 um dos grandes desafios é produzir inibidores que possuam uma alta seletividade com relação a proteases dos parasitas de modo a distinguir de proteases encontradas em humanos [176, 188, 189].

2.2 Objetivos

2.2.1 Objetivo Geral

Verificar o comportamento da DeepVS em um estudo de caso padrão de DBVS usando como alvo a enzima Cruzaína.

2.2.2 Objetivos Específicos

- Realizar estudos estruturais do sítio ativo da Cruzaína por meio de pesquisa em literatura e comparação de estrutura utilizando programas de visualização;
- Selecionar a melhor metodologia envolvendo a DeepVS para estudos de DBVS da Cruzaína utilizando controle positivo;
- Verificar o comportamento da DeepVS em um estudo de caso;
- Realizar etapas de *virtual screening* para um conjunto de compostos filtrados do banco de dados ZINC utilizando *docking* molecular e *deep learning* para o receptor da Cruzaína.
- Examinar os compostos ranqueadas para Cruzaína.

2.3 Material e Métodos

2.3.1 Seleção da estrutura cristalográfica

No banco de dados de estruturas de proteínas PDB [42] estão disponíveis 25 estruturas para a Cruzaína (última checagem Janeiro de 2016). As estruturas foram filtradas utilizando as seguintes características: ligantes que não se ligam covalentemente ao receptor e não formam agregadores. Após a filtragem cinco estruturas foram selecionadas (Figura 2.7): ID_PDB: 1me3 [190], ID_PDB: 1me4 [5], ID_PDB: 3kku [191], ID_PDB: 4klb [192] e ID_PDB: 4xui [193]. Para escolha da estrutura da enzima Cruzaína para etapas de *virtual screening* foram realizadas comparações entre estruturas e de valores de resolução. Por exemplo, optamos por estruturas que possuem um sítio conservado e possuem uma alta resolução. Dessa forma, foi escolhida a estrutura ID_PDB: 3kku [191] para as demais etapas de DBVS (Figura 2.8).

2.3.2 Estudos estruturais

Estudos estruturais da enzima Cruzaína foram realizados utilizando o programa de visualização de estruturas Chimera [121], PyMOL [194], o algoritmo BINANA [144] que avalia o sítio de ligação e as ligações formadas pelo complexo proteína-composto, o programa PoseView [195] e o LigPlot [196] a partir dos dados cristalográficos disponíveis no banco de dados PDB [42]. Com isso, foi possível identificar resíduos do sítio de ligação da enzima que possuem interação com compostos (Figura 2.8). Como por exemplo: Cys 25, Gly 66, Asp 161, His 162, Glu 20 e Leu 160 dentro de um raio de corte de 5 Å a partir do ligante.

2.3.3 Geração da lista de ligantes conhecidos e *decoys* para a Cruzaína

Uma lista com um total de 38 ligantes conhecidos para a Cruzaína foi construída baseado na literatura e na lista de compostos disponibilizada por Santos [197]. Para desenvolver a lista de ligantes conhecidos foram considerados ligantes disponíveis em banco de dados públicos como: BindingDB [198], ChEMBL [49] e ZINC [115]. Além disso, ligantes que se ligam covalentemente ou que sejam agregadores foram descartados.

Os ligantes selecionados foram retirados dos seguintes trabalhos: 13 ligantes pertencentes à publicação de Du *et al.* (2000) [199]; 22 ligantes pertencentes à publicação Ferreira *et al.* (2010) [191] e Ferreira *et al.* (2014) [200]; e 3 pertencentes à publicação Rogers *et al.* (2012) [201]. As estruturas dos ligantes em formato **.mol2**

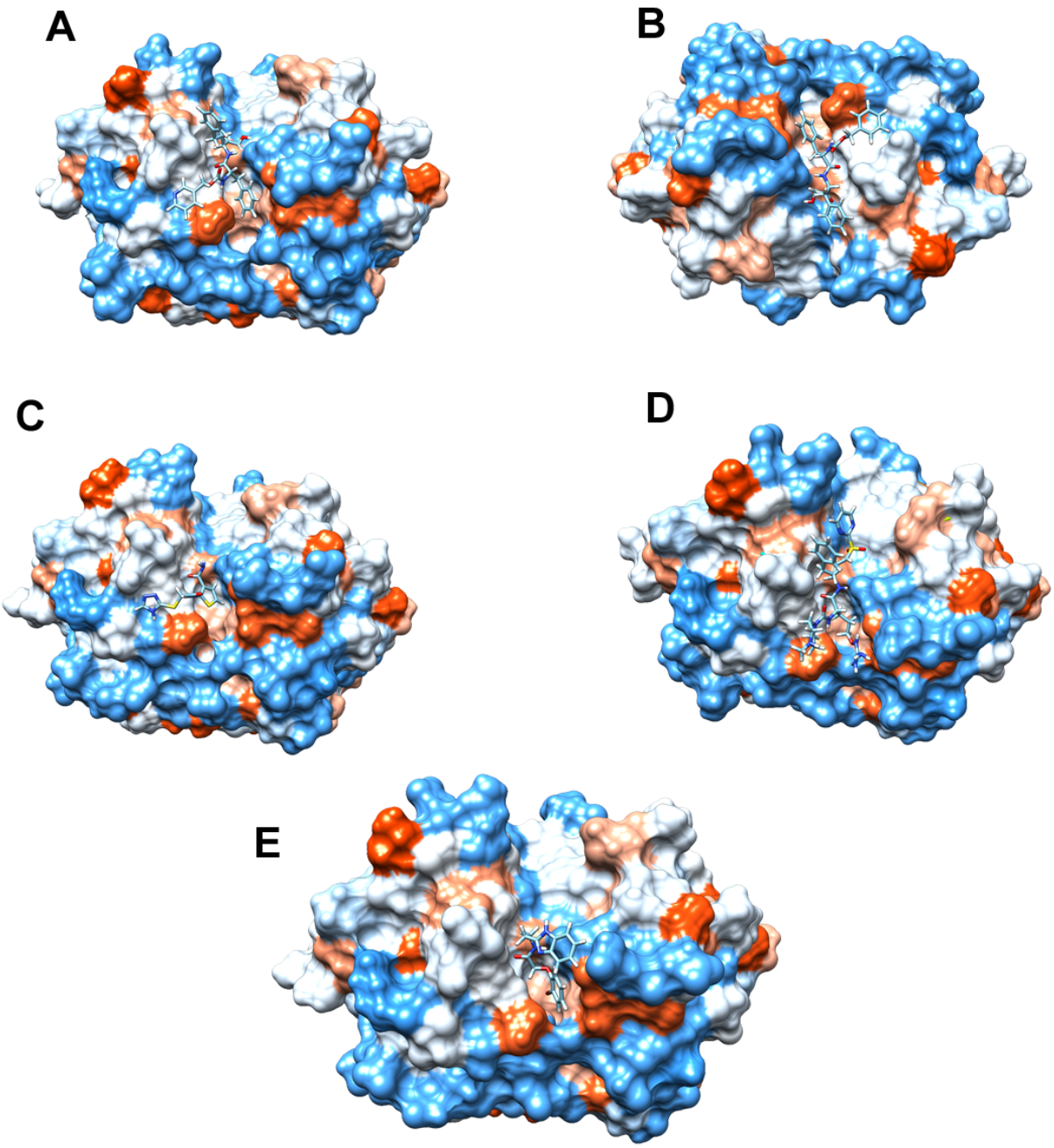


Figura 2.7: **Estruturas pertencentes à enzima Cruzaína selecionadas do PDB.** A) representação da estrutura ID_PDB: 1me3. B) representação da estrutura ID_PDB: 1me4. C) representação da estrutura ID_PDB: 4klb. D representação da estrutura ID_PDB: 4xui. E) representação da estrutura selecionada para etapas de *virtual screening* ID_PDB: 3kku.

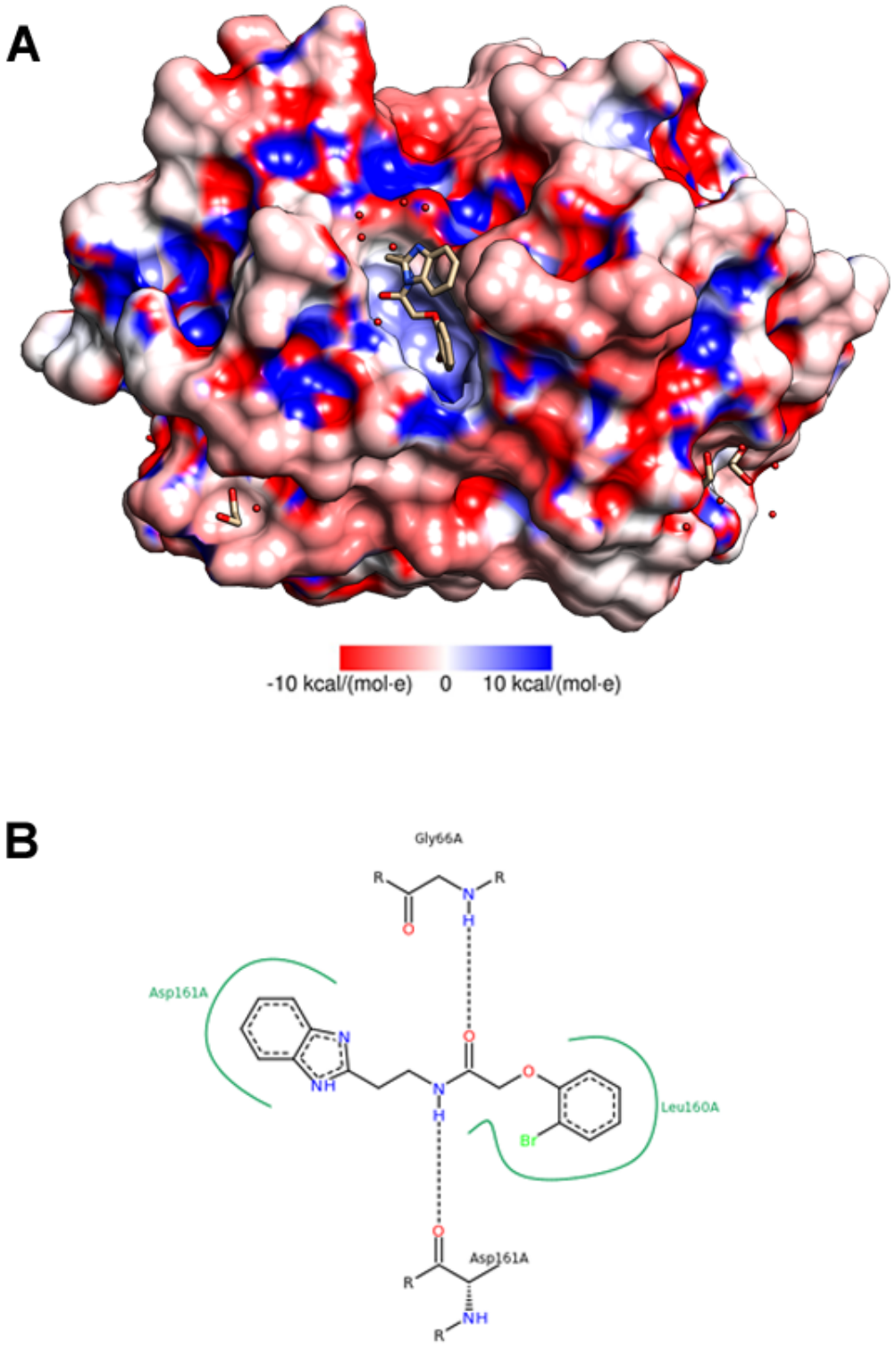


Figura 2.8: **Superfície da enzima Cruzaína e seu ligante B95.** A) Potencial eletrostático da superfície da Cruzaína ID_PDB: 3kku. Em vermelho áreas carregadas negativamente, em azul áreas carregadas positivamente e em branco áreas neutras. B) Mapa de interação do inibidor B95.

foram retiradas diretamente do banco de dados ZINC [115] (Figuras 2.9 e 2.10).

Os *decoys* foram gerados utilizando o banco de dados ZINC [115] em uma proporção média de 32 *decoys* para cada ligante. Os compostos gerados pelo ZINC conhecidos como *decoys* possuem propriedades semelhantes (peso molecular, logP, doadores e aceptores de hidrogênio) aos do ligante porém são quimicamente diferentes. Características como ordem de ligação, estereoquímica e estado de protonação gerados pelo o ZINC [115] foram mantidos. Para os procedimentos de *docking* e *virtual screening* foram atribuídas cargas atômicas parciais do tipo *Gasteiger* [124] por meio do *software* Open Babel [123] para o programa *Dock6.6* [118] e AutoDockTools4 [74] para o programa *AutoDockVina* [70].

2.3.4 Preparação da estrutura molecular do receptor

A estrutura cristalográfica da Cruzaína foi obtida utilizando o banco de dados PDB [42] pertencente ao código ID_PDB: 3kku [191], com resolução de 1,28 Å. A preparação do estrutura do receptor consistiu na adição de hidrogênios nos resíduos de acordo com o estado de protonação ácida (pH = 5,5) usando o programa Propka [202].

2.3.5 Escolha do programa de *docking* utilizando controles positivos

Os programas de *docking* Dock6.6 [118] e Autodockvina1.2 [70] foram avaliados utilizando controles positivos (ligantes conhecidos) e seus respectivos controles negativos (*decoys*) para a enzima Cruzaína (seção 2.3.3). A escolha do programa de *docking* utilizado para etapa de *virtual screening* se deu mediante a comparação dos resultados utilizando métricas de avaliação bem estabelecidas como o fator de enriquecimento (*Enrichment Factor*-EF) e a área sob a curva ROC (*Receiver Operating Characteristic*) [133, 134] (seção 1.3.4).

2.3.5.1 Configuração do programa Dock6.6 para a Cruzaína

A estrutura do receptor foi preparada usando a ferramenta Dock Prep fornecida pelo o programa *Chimera* [121]. O Dock Prep foi utilizado para remoção das moléculas de água, reparação de cadeias laterais truncadas e remoção do ligante. Os átomos de hidrogênio foram adicionados utilizando programa Propka [202].

As esferas foram geradas usando o programa SPHGEN [126]. O conjunto *cluster* de esferas foi selecionado utilizando o programa *sphere_selector* e as coordenadas do átomo de enxofre pertencente ao resíduo Cys 25. O conjunto de esferas escolhido está localizado em um raio de 10 Å de distância tomando o átomo de enxofre como centro (Figura 2.11).

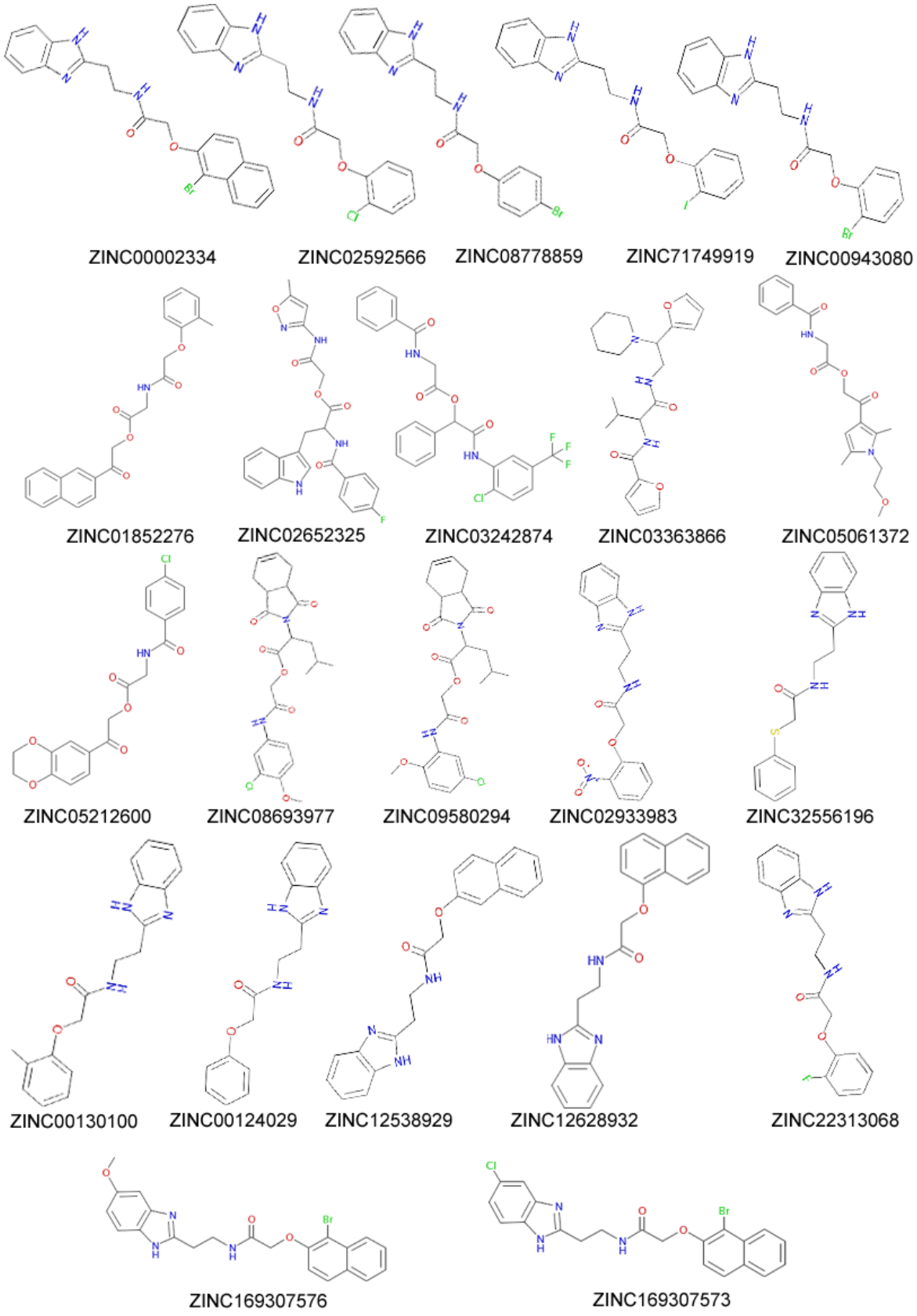


Figura 2.9: **Estrutura em 2D dos ligantes conhecidos para a Cruzaína extraúdos das publicações de Ferreira *et al.* (2010) e Ferreira *et al.* (2014).**

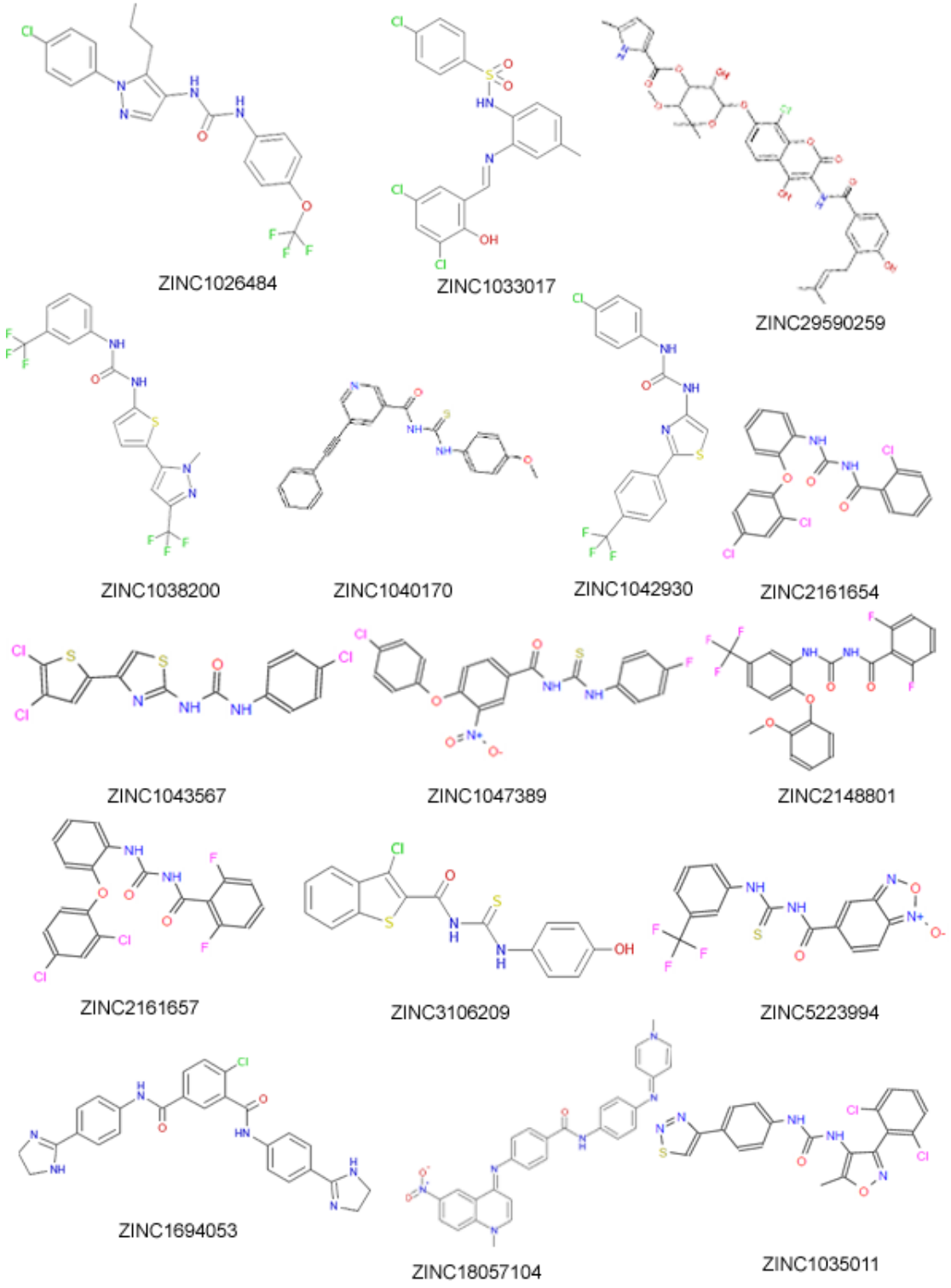


Figura 2.10: **Estrutura em 2D dos ligantes conhecidos para a Cruzaína extraídos das publicações de Du *et al.* (2000) e Rogers *et al.* (2012)**

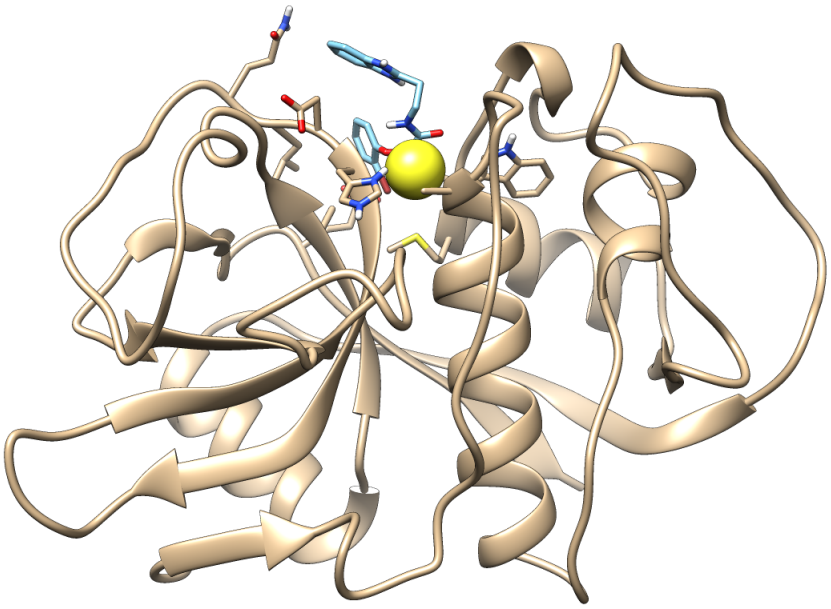


Figura 2.11: **Representação do átomo de enxofre na Cruzaína** Representação da estrutura da Cruzaína ID_PDB: 3kku, em amarelo o átomo de enxofre do resíduo Cys 25 utilizado para selecionar as esferas e em azul o ligante B95.

Para delimitar o tamanho da *grid* foi construída uma caixa cúbica (*box*) em torno do sítio de ligação com o programa *showbox*, delimitado pelo conjunto de esferas selecionadas, com uma margem extra de 5 Å em todas as 3 direções escolhidas (Figura 2.12).

A *grid* foi computada usando o programa Grid e foram utilizadas as seguintes configurações: espaçamento da *grid* de 0,3 Å , uma distância de corte de 9,999 Å, expoentes de Lennard-Jones 6-12 (atrativos-repulsivos) e uma constante dielétrica dependente da distância $4r$.

Rodadas de *docking* semiflexível foram efetuadas usando o programa Dock6.6 para os dois estados de protonação da díade catalítica do receptor (par iônico (His 162)-NH^+/(Cys 25)-S^- e a cisteína eu seu estado neutro (His 162)-NH^+/(Cys25)-SH) somado à lista de ligantes conhecidos e *decoys* gerados para esse trabalho (seção 2.3.3). Os compostos foram ranqueados utilizando o valor da função de pontuação Grid Score.

2.3.5.2 Configuração do programa Autodockvina1.2 para a Cruzaína

O receptor foi preparado utilizando o *script prepare_receptor4.py* disponível no programa MGLTools [74]. Os átomos de hidrogênio foram adicionados utilizando o programa Propka [202]. Os diferentes estados de protonação do receptor foram salvos em formato **.pdbqt**.

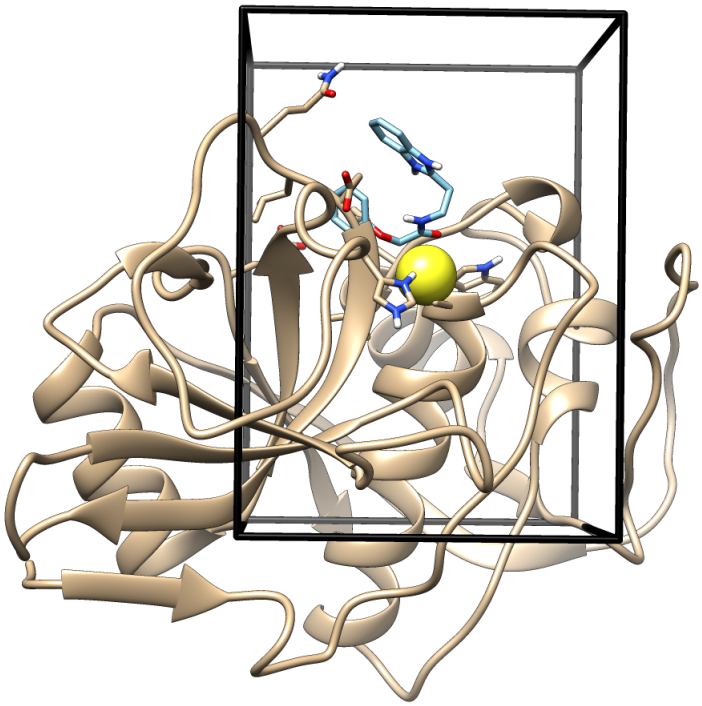


Figura 2.12: **Representação da caixa cúbica (*box*) em preto, que delimita o espaço no qual a *grid* será criada.**

A *grid* foi definida com dimensões x, y, z contendo 22 Å em cada direção. O átomo de enxofre do resíduo Cys 25 foi utilizado como átomo central para a construção *grid box* (Figura 2.13).

As rodadas de *docking* foram efetuadas para os dois estados de protonação da díade catalítica do receptor (par iônico (His 162)-NH^+/(Cys25)-SH e a cisteína eu seu estado neutro (His 162)-NH^+/(Cys25)-SH) e seus receptivos ligantes e *decoys* gerados para esse trabalho (seção 2.3.3), utilizando o programa de *docking* Autodockvina1.1.2 [70]. As configurações utilizadas para o procedimento de *docking* são: diferença de energia máxima entre a melhor ligação (*energy_range*) igual a 10; o número máximo de modos ligação gerados (*num_modes*) igual a 1, no qual compreende a melhor pose gerada pelo programa; a quantidade de CPUs igual a 6; tempo gasto aproximadamente para efetuar a pesquisa global (*exhaustiveness*) igual 16; e o valor do parâmetro semente (*seed*) correspondente a -16807. Os compostos foram ranqueados utilizando o valor da função de pontuação pertencente ao programa Autodockvina1.1.2.

2.3.6 Conjunto de compostos para a etapa de *virtual screening*

Para seleção do conjunto de moléculas para *virtual screening* utilizamos a versão 15 do banco de dados ZINC [53]. Selecionamos apenas as moléculas do tipo *lead-like*. Esses compostos são caracterizados por serem grandes o suficientes para

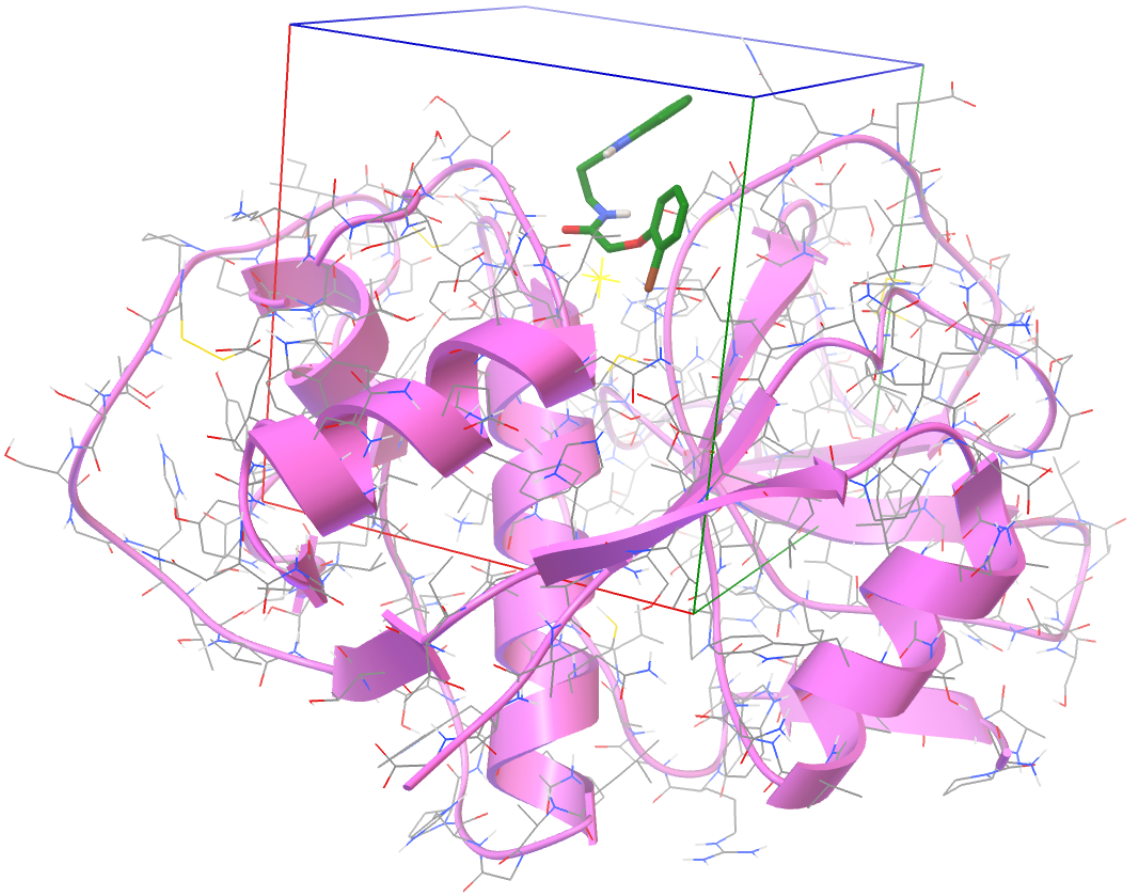


Figura 2.13: **Representação da *grid* utilizada na etapa de *docking* com o programa Autodockvina1.2**. O x em amarelo representa o ponto a partir do qual a *grid* foi criada (átomo de enxofre); o ligante B95 é representado em verde; a *grid box* é representada em suas dimensões x, y e z nas cores vermelho, verde e azul respectivamente.

serem detectados em ensaios de alto rendimento, mas menores e mais solúveis que a maioria dos fármacos disponíveis. Além da seleção de moléculas do tipo *lead-like*, aplicamos os seguintes filtros: pH entorno de 5,5; peso molecular $\leq$ 350 *daltons*; LogP $\leq$ 3,5; compostos que estão disponíveis no estoque. Um total de 90.769 compostos foi selecionado para etapa de *virtual screening*.

2.3.7 *Virtual Screening*

O primeiro passo para a bordagem de *virtual screening* baseado em estrutura (*Docking-Based Virtual Screening*-DBVS) utilizada neste trabalho para o estudo de caso da enzima Cruzaína consiste na escolha da metodologia. Para isso, foram utilizados dois programas de *docking* Autodockvina1.2 e Dock6.6 (Figura 2.14).

A DeepVS foi aplicada à saída do processo de *virtual screening* para ambos os programas de *docking* Autodockvina1.2 e Dock6.6. Dessa forma, foi gerada uma

lista ranqueada de compostos para duas diferentes metodologias: Autodockvina1.2 + DeepVS e Dock6.6 + DeepVS.

O teste para escolha da metodologia consistiu no uso de um controle positivo. Ou seja, uma lista de ligantes conhecidos e *decoys* foi gerada para avaliar o quanto cada metodologia conseguiu distingir entre verdadeiros positivos em função de falsos positivos (seção 2.3.3). A metodologia que registrou o melhor valor da AUC e fator de enriquecimento (EF) foi a metodologia utilizada para os demais processos de DBVS. Nesse caso, a metodologia escolhida foi a Autodockvina1.2 + DeepVS (Figura 2.14).

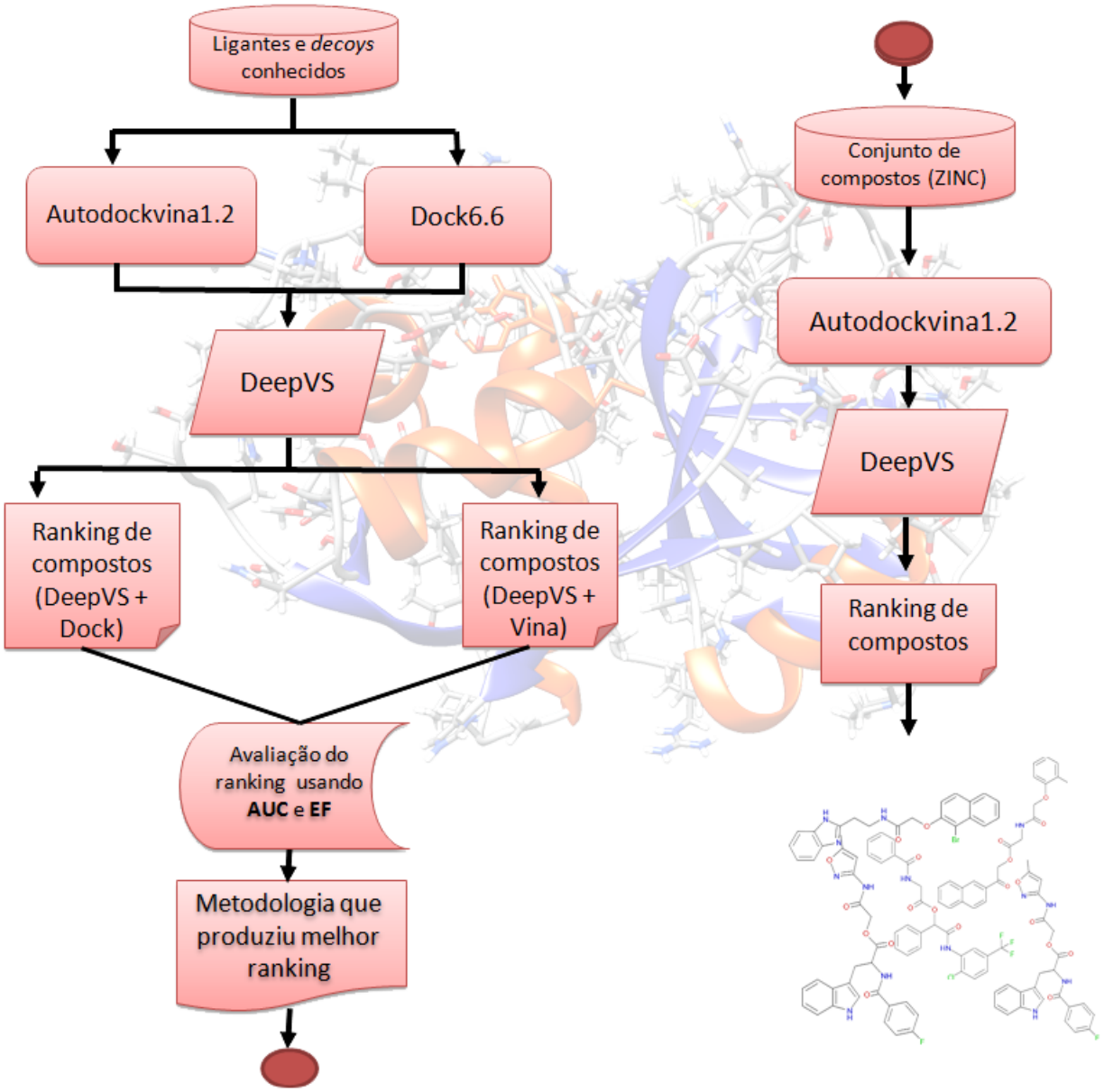


Figura 2.14: **Representação esquemática dos processos de DBVS usados para a enzima Cruzaína**.

As demais etapas de DBVS consistiram na seleção de conjunto compostos utilizando como base de dados o banco de compostos ZINC (seção 2.3.6). Para a etapa de *docking* molecular de todos os compostos pertencentes ao conjunto no receptor

da enzima Cruzaína foi utilizado o programa Autodockvina1.2. O resultado desse processo foi dado como entrada para a DeepVS, que por fim gerou uma lista de compostos ranqueados.

Para o presente estudo de caso foi utilizada a versão da DeepVS treinada com todas as 40 proteínas do DUD. Os hiperparâmetros utilizados foram: d^{atm} = 200; d^{amino} = 200; d^{chrg} = 200; d^{dist} = 200; cf = 400; h = 50; λ = 0,075; k_c = 6; k_p = 2. Planejamos, no futuro, realizar um estudo de sensibilidade dos parâmetros especificamente para enzima Cruzaína.

2.4 Resultados e Discussão

Segundo o pesquisador Yin [203], um estudo de caso deve ser completo e significativo. Completo no sentido que os limites do fenômeno estudado e seu cotexto sejam bem definidos e significativo no sentido de buscar uma contribuição para sociedade e/ou despertar o interesse do público em geral para o objeto de estudo.

Dessa forma, a enzima Cruzaína demonstra ser uma excelente candidata para o estudo de caso envolvendo a estratégia de melhoramento de DBVS proposta pelo o presente trabalho, a DeepVS, pois a enzima (apesar de ser uma protease) não possui nenhuma relação estrutural próxima com os receptores formadores do banco de dados DUD [64], o que reforça o sentido completo do estudo de caso.

Com relação à significância do estudo de caso, a enzima Cruzaína é considerada um alvo molecular importante para combater o *Trypanosoma cruzi*, parasita causador da doença de Chagas. Por ser uma das principais e mais abundantes cisteína-proteases expressas durante o processo de infecção pelo *Trypanosoma cruzi*, essa enzima é considerada essencial para o estágio de infecção em humanos [182]. A doença de Chagas ou tripanossomíase americana é uma infecção parasitária que atinge em sua grande parte a população em um estado de "vulnerabilidade social" e não possui uma medicação segura e eficaz para o seu tratamento. Dessa forma, o uso de novas tecnologias como *deep learning* para busca de fármacos eficazes e seguros para tratamento da doença de Chagas constitui uma ampla contribuição não só para a acadêmia como para o público em geral.

2.4.1 Identificação da estrutura

No banco de dados de estruturas de proteínas (*Protein data Bank* - PDB) [42] estão depositadas 25 estruturas para a enzima Cruzaína. Após etapas de filtragem e estudos estruturais foi escolhida a ID_PDB: 3kku como estrutura alvo para as demais etapas de *virtual screening* baseado em estrutura (*Docking-Based Virtual Screening*-DBVS).

A estrutura portadora da ID_PDB: 3kku [191], corresponde à estrutura da enzima Cruzaína complexada com o inibidor B95, foi escolhida entre outros motivos por possuir alta qualidade, com resolução de 1,28 Å e o seu sítio ativo possuir um alto grau de conservação se comparado com outras estruturas disponíveis no PDB para o parasita *Trypanosoma cruzi*. Um outro fator importante que levou à escolha dessa estrutura, foram os tipos de iterações observadas entre o complexo proteína-composto que prioriza ligações de hidrogênio entre o ligante e os resíduos Gly 66 e Asp 161 (numeração cruzaína), e um modo de interação não covalente com o inibidor (Figura 2.15).

O estado de protonação dos resíduos cisteína e histidina formadores da díade catalítica da enzima Cruzaína não é totalmente elucidado. Alguns trabalhos sugerem que a díade consiste de um par de íons tiolato-imidazólio (His 162)-NH^+/(Cys 25)-S^-

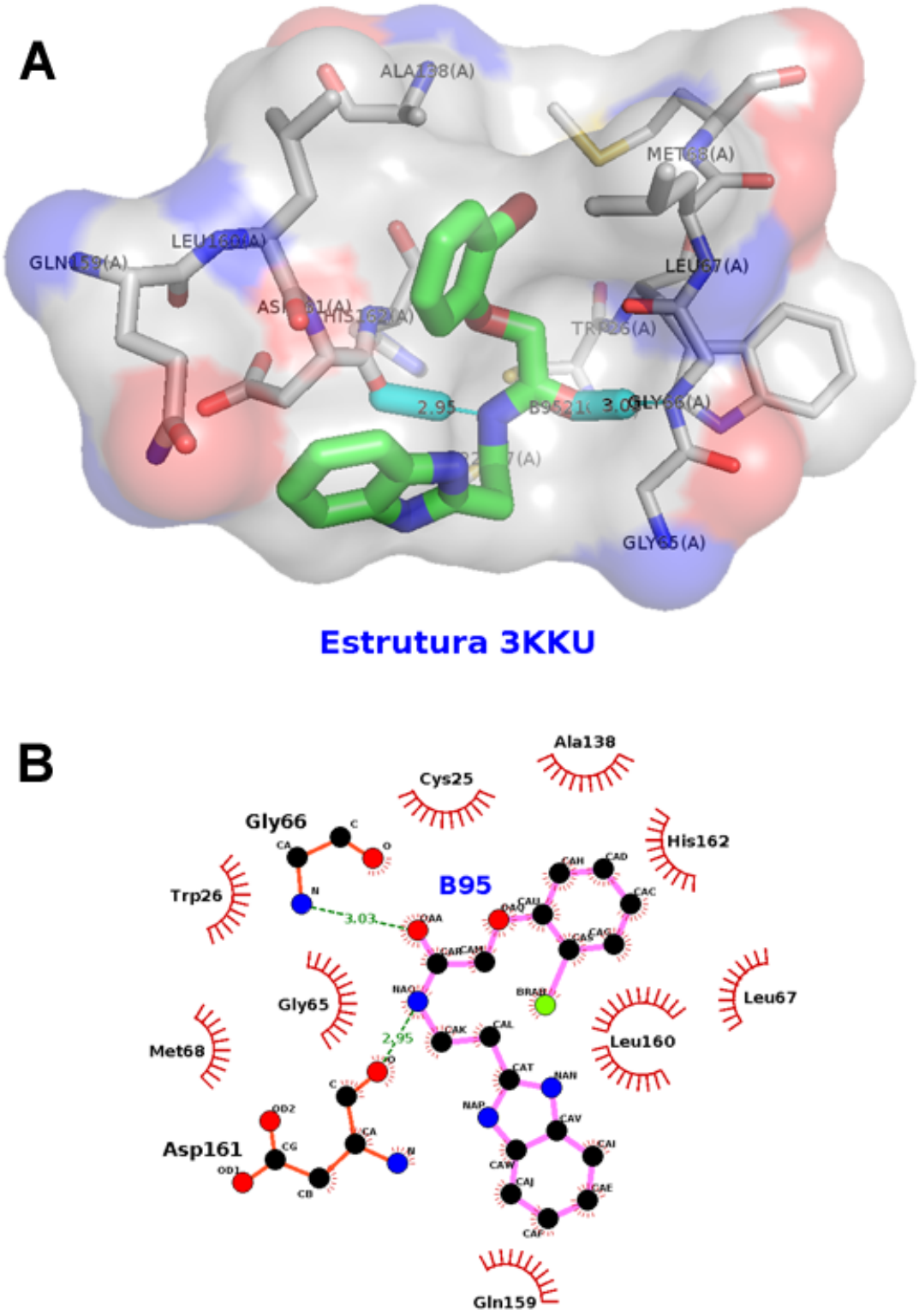


Figura 2.15: **Interações complexo proteína-composto estrutura ID_PDB: 3kku**. A) Representação do ligante B95 (verde) no sítio de ligação da enzima Cruzaína. B) Mapa de interação do ligante B95, em verde as duas ligações de hidrogênio realizadas entre o ligante e os resíduos Gly 66 e Asp 161.

(numeração cruzaína), representado nesse trabalho como (His 162)p/ (Cys 25)d [204, 205]. Porém, outros trabalhos sugerem que a díade catalítica é formada com a cisteína em seu estado neutro no qual à (His 162)-NH^+/(Cys25)-SH [206, 207], representada nesse trabalho como (His 162)p/(Cys 25)n (Figura 2.16). Como não há um consenso a respeito do mecanismo catalítico da enzima Cruzaína, para esse estudo de caso, consideramos os resíduos da enzima em seus dois estados de protonação (His 162)-NH^+/(Cys 25)-S^- e (His 162)-NH^+/(Cys 25)-SH.

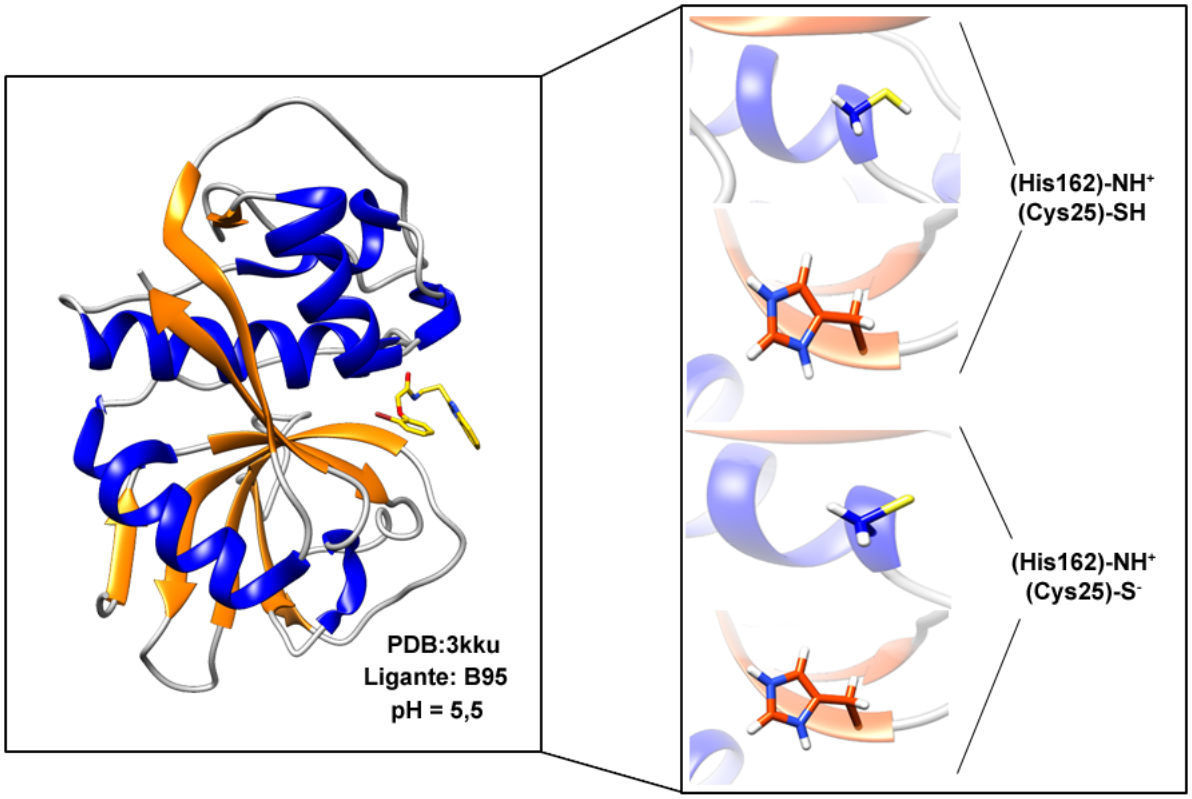


Figura 2.16: **Representação gráfica do sítio de ligação da enzima Cruzaína**. A díade catalítica está representada em ambos os estados de protonação.

2.4.2 Seleção da metodologia utilizando controle positivo

As metodologias consistiram na utilização de dois programas de *docking* molecular distintos AutodockVina1.1.2 [70] e Dock6.6 [118], onde o resultado de cada programa de *docking* individualmente foi dado como entrada para a DeepVS treinada utilizando o banco de dados DUD [64]. Dessa forma, duas metodologias foram avaliadas: (1) DeepVS-Dock (Dock6.6 + DeepVS) e (2) DeepVS-ADV (AutodockVina1.1.2 + DeepVS).

Em ordem de selecionar a melhor metodologia levando em consideração um alvo específico, a Cruzaína, uma lista de ligantes conhecidamente ativos foi construída baseado na literatura e seus respectivos *decoys* foram gerados utilizando o banco de dados ZINC [115]. O uso de ligantes conhecidamente ativos para um determinado

alvo como forma de avaliação de uma metodologia de *docking* é representado nesse trabalho como controle positivo. Para avaliar a performance de cada metodologia proposta foram utilizados métricas bem estabelecidas como fator de enriquecimento (ef) e a área sob a curva ROC (AUC).

O programa de *docking* molecular Dock6.6 (AUC = 0,55) demonstrou melhor performance para o receptor contendo o par iônico (His 162)-NH^+/(Cys 25)-S^- se comparado ao programa AutodockVina1.1.2 (AUC = 0,34) (Tabela 2.1 e Figura 2.17). No entanto, o programa de *docking* molecular AutodockVina1.1.2 (AUC = 0,63) demonstrou melhor performance se comparado ao programa Dock6.6 (AUC = 0,56) quando a Cys 25 pertencente a díade catalítica encontrava-se no seu estado neutro (His 162)-NH^+/(Cys 25)-SH (Tabela 2.2 e Figura 2.18).

Tabela 2.1: Valores de AUC ROC, fator de enriquecimento (*ef*) a 2%, 20% e fator de enriquecimento máximo (ef_{max}) para a Cruzaína em dois diferentes estados de protonação correspondente a performance de *virtual screening* do Dock6.6 e da DeepVS-Dock.

	Dock				**DeepVS-Dock**			
	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	**AUC***	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	**AUC***
(His 162)p/(Cys 25)d	3,2	2,6	0,8	0,55	2,5	1,3	2,0	**0,70**
(His 162)p/(Cys 25)n	1,4	0,0	1,1	0,56	1,5	1,3	1,4	**0,62**

* Valores em negrito indicam o maior valor de AUC computado em cada caso.

Tabela 2.2: Valores de AUC ROC, fator de enriquecimento (*ef*) a 2%, 20% e fator de enriquecimento máximo (ef_{max}) para a Cruzaína em dois diferentes estados de protonação correspondente a performance de *virtual screening* do AutodockVina1.1.2 e da DeepVS-ADV.

	ADV				**DeepVS-ADV**			
	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	**AUC***	ef_{max}	$ef_{2\%}$	$ef_{20\%}$	**AUC***
(His 162)p/(Cys 25)d	2,0	2,0	0,8	0,34	3,6	2,0	2,2	**0,66**
(His 162)p/(Cys 25)n	2,2	0,0	1,2	0,63	4,9	0,0	3,0	**0,74**

* Valores em negrito indicam o maior valor de AUC computado em cada caso.

Aparentemente, a carga dos resíduos no sítio catalítico da enzima Cruzaína não possui uma grande influência na performance do programa Dock6.6 (Tabela 2.1). Por um outro lado, o estado de protonação da díade catalítica na Cruzaína é crucial para a performance do AutodockVina1.1.2 (Tabela 2.2). No programa de *docking* molecular AutodockVina1.1.2, pode-se observar uma melhora entorno de 85% no valor de AUC apenas alterando o estado de protonação da díade catalítica. O comportamento dos programas de *docking* para diferentes estados de protonação da díade catalítica da Cruzaína pode estar relacionado ao tipo de estratégia usada durante à busca conformacional e avaliação dos modos de ligação pela função de pontuação. Os resultados

apresentados nessa seção corroboram com os resultados discutidos na seção 1.4.2 e reforçam a importância de estudos prévios com ligantes conhecidamente ativos para um determinado receptor para a escolha da melhor abordagem de *virtual screening* a ser adotada. A deepVS melhorou o resultado de AUC em todos os casos propostos, em ambos estados de protonação da díade catalítica e para ambos os programas de *docking* molecular Dock6.6 e AutodockVina1.1.2 (Figuras 2.17 e 2.18). Além disso, quando foi selecionado 20% do conjunto de dados a deepVS melhorou o resultado do fator de enriquecimento em todos os experimentos propostos (Figuras 2.19 e 2.20).

No caso específico do programa AutodockVina1.1.2 para o receptor portador do par iônico ((His 162)-NH^+/(Cys 25)-S^-), a deepVS melhorou o resultado de AUC em torno de 95% (Tabela 2.2 e Figura 2.18) e o o fator de enriquecimento em 20% do conjunto de dados em 175% (Tabela 2.2 e Figura 2.20).

A estratégia que reportou melhor resultado tanto de AUC quanto de fator de enriquecimento foi a DeepVS-ADV para o receptor quando a Cys 25 pertencente a díade catalítica encontrava-se em seu estado neutro (Figuras 2.18 e 2.20). Estudos de dinâmica molecular realizados por Santos [208] para os dois estados de protonação da díade catalítica da Cruzaína ((His 162)-NH^+/(Cys 25)-S^- e (His 162)-NH^+/(Cys 25)-SH)) e respectivamente o ligante nativo B95 protonado e desprotonado para a estrutura ID_PDB: 3kku, demonstrou que Cys 25 em seu estado neutro e o ligante B95 desprotonado apresentam a conformação mais próxima da cristalográfica supondo que o mecanismo de troca de próton pode envolver (His 162)-NH^+/(Cys 25)-SH) e o ligante em seu estado neutro, o que corrobora com os nossos resultados.

Os resultados dos experimentos apresentados nessa seção sugerem que a rede DeepVS pode ser usada como uma estratégia efetiva para melhorar *virtual screening* baseado em estrutura e pode ser aplicada como uma estratégia alternativa na busca de possíveis ligantes ativos para a enzima Cruzaína.

2.4.3 *Virtual Screening*

A estratégia escolhida para as demais etapas de *virtual screening* baseado em estrutura consiste em rodadas de *docking* molecular utilizando o AutodockVina1.1.2 e subsequentemente geração da lista de compostos ranqueados utilizando a DeepVS, que corresponde à metodologia DeepVS-ADV para enzima Cruzaína apresentando a Cys 25 pertencente a díade catalítica em estado neutro.

A metodologia DeepVS-ADV foi escolhida por apresentar os melhores valores de AUC e fator de enriquecimento durante os experimentos utilizando controle positivo (seção 2.4.2). Outro fator que contribuiu para escolha desta metodologia foi o ef máximo ter sido reportado em torno dos 4% do conjunto de dados e constituir o maior ef máximo registrado para todos os experimentos utilizando controle positivo efetuados (Figuras 2.19 e 2.20).

Um conjunto de 90.769 compostos do tipo *lead-like* foi selecionando a partir do

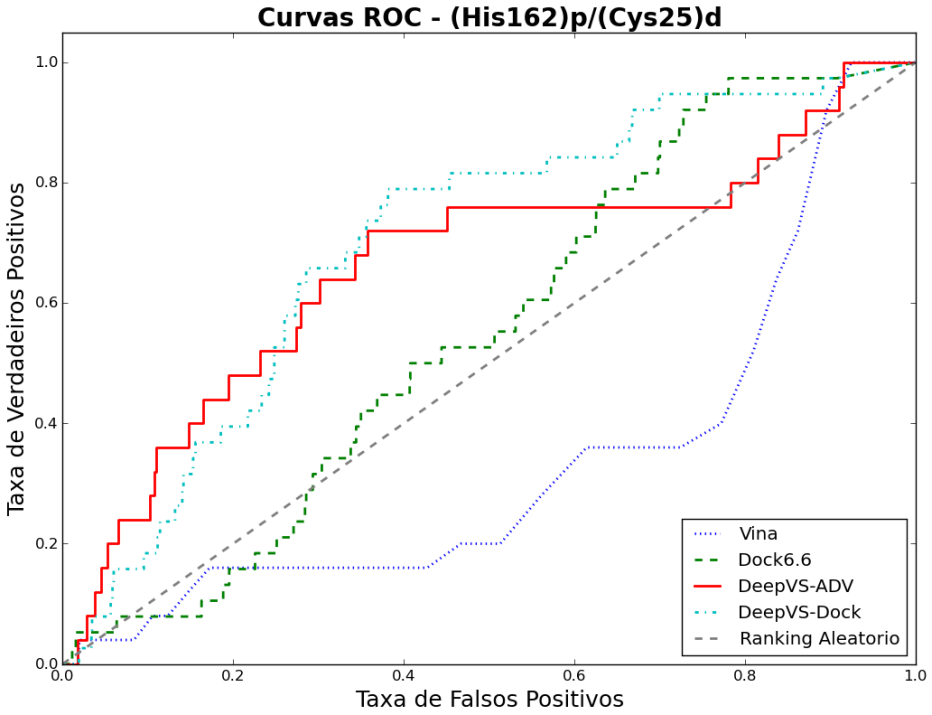


Figura 2.17: **Curvas ROC do desempenho das metodologias**, AutodockVina1.1.2 (azul), Dock6.6 (verde), DeepVS-ADV (vermelho) e DeepVS-Dock (ciano) para o receptor com o par iônico (His 162)-NH^{+}/(Cys 25)-S^{-}.

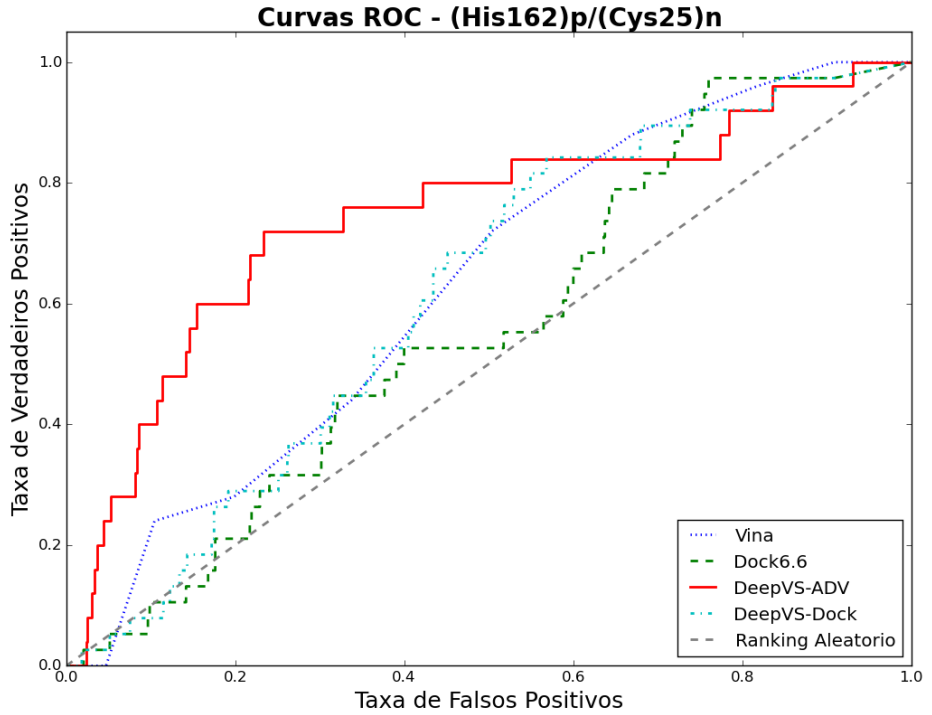


Figura 2.18: **Curvas ROC do desempenho das metodologias**, AutodockVina1.1.2 (azul), Dock6.6 (verde), DeepVS-ADV (vermelho) e DeepVS-Dock (ciano) para o receptor com (His 162)-NH^{+}/(Cys 25)-SH.

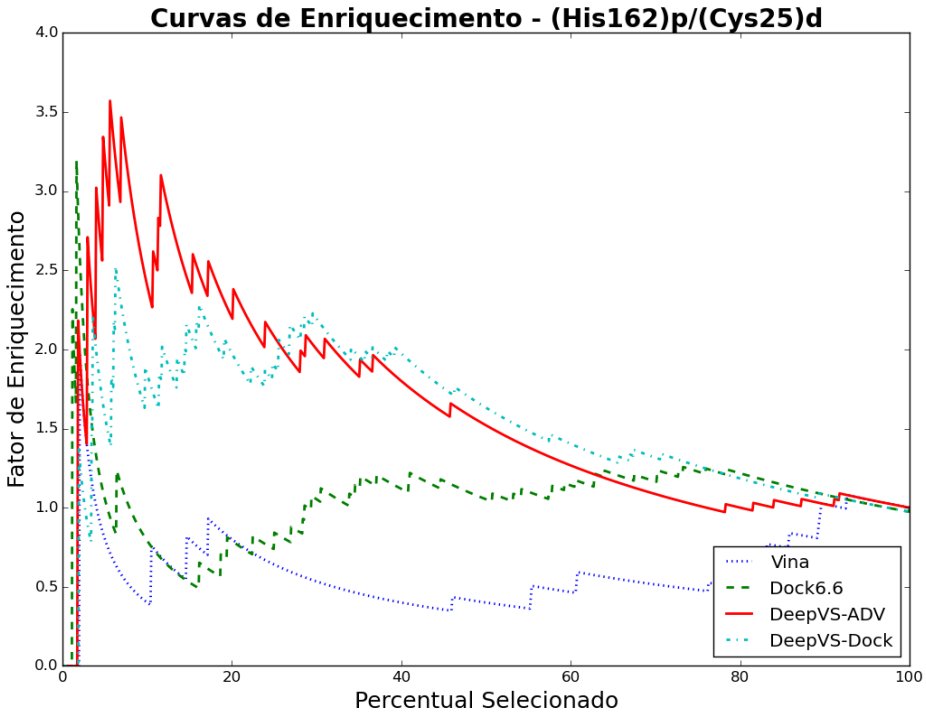


Figura 2.19: **Gráfico de enriquecimento do desempenho das metodologias**, Autodock-Vina1.1.2 (azul), Dock6.6 (verde), DeepVS-ADV (vermelho) e DeepVS-Dock (ciano) para o receptor com o par iônico (His 162)-NH^{+}/(Cys 25)-S^{-}.

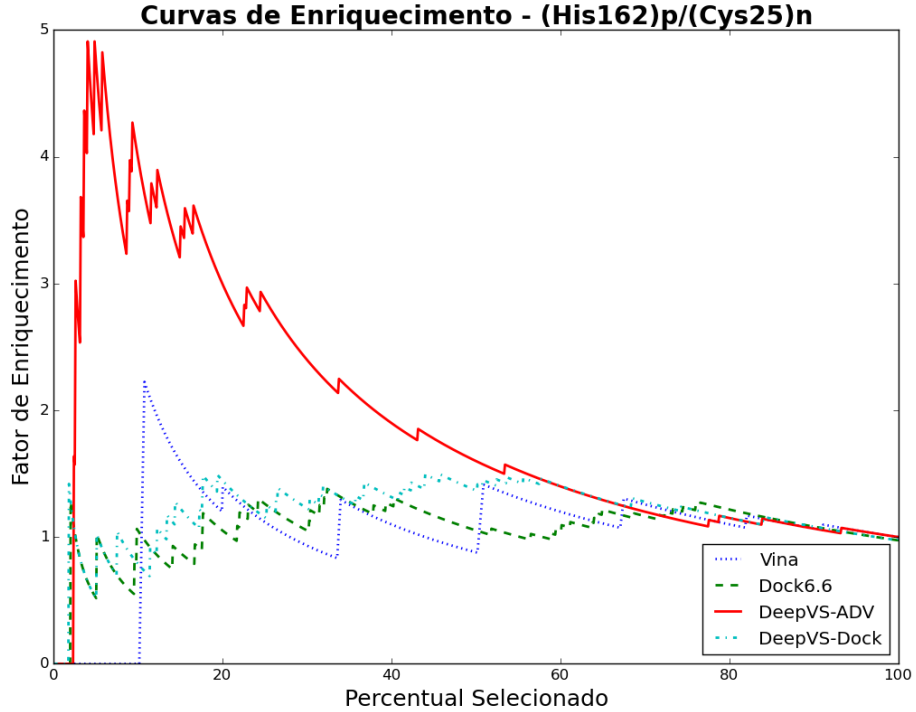


Figura 2.20: **de enriquecimento do desempenho das metodologias**, AutodockVina1.1.2 (azul), Dock6.6 (verde), DeepVS-ADV (vermelho) e DeepVS-Dock (ciano) para o receptor com (His 162)-NH^{+}/(Cys 25)-SH.

banco de dados ZINC versão 15 [53]. A metodologia DeepVS-ADV recebeu como entrada o conjunto de compostos filtrados do ZINC e gerou como saída uma lista de compostos ordenados com relação a uma possível afinidade com a enzima Cruzaína. Os 200 primeiros compostos ranqueados foram selecionados para testes adicionais que envolvem predição de toxidade e avaliação visual de interação entre os compostos e o receptor.

O servidor ChemBioSer [209] foi utilizado para verificar uma possível toxicidade dos compostos selecionados. Para os 200 compostos verificados, 11 reportaram a presença de um ou mais grupos orgânicos tóxicos (Tabela 2.3). Todos os 11 compostos foram removidos da lista de compostos selecionados. Os 50 primeiros compostos da lista resultante foram para a etapa de análise visual. Para essa etapa foi utilizado os programas de visualização gráfica Chimera [121] e PoseView [195]. Compostos que estão localizados dentro sub-sítio S2 da Cruzaína e apresentam uma ou mais ligações de hidrogênio foram selecionados em detrimentos dos demais compostos (Figura 2.21). Compostos que apresentaram uma conformação anormal foram descartados (Figura 2.23). O composto ZINC000067842600 (Figura 2.22) foi descartado do conjunto final de ligantes por ser um composto positivamente carregado relacionado ao sítio da Cruzaína no qual representa um ambiente hidrofóbico. Seis compostos foram selecionados, dentre estes, três compostos estão localizados entre as 10 primeiras posições no ranking gerado pela DeepVS-ADV (Tabela D.1) .

Os resultados apresentados nessa seção indicam que a metodologia apresentada demonstra-se eficaz para escolher compostos possivelmente ativos para a enzima Cruzaína. Esse resultado consiste em mais uma evidência de que a DeepVS é um método robusto para melhoramento de DBVS. Porém, para determinar qual composto poderia ser um candidato a fármaco, são necessários testes adicionais com outros métodos tais como dinâmica molecular.

Tabela 2.3: Compostos com toxicidade reportada presente no sub-conjunto de 200 compostos.

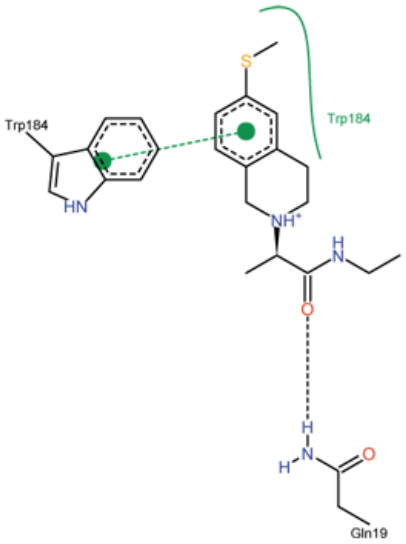
Compostos (ID_ZINC)	**Grupo tóxico[a]**
ZINC000007677132	catecol (1,2-dihidroxibenzeno)
ZINC000011953190	catecol (1,2-dihidroxibenzeno)
ZINC000013113949	diazeno (Diimide ou Diimine)
ZINC000013115388	diazeno (Diimide ou Diimine)
ZINC000020028773	catecol e Benzodioxano
ZINC000007677132	aminothiazol
ZINC000040481024	catecol (1,2-dihidroxibenzeno)
ZINC000084612899	catecol (1,2-dihidroxibenzeno)
ZINC000091814921	catecol (1,2-dihidroxibenzeno)
ZINC000095369152	catecol e Benzodioxano
ZINC000106819613	catecol (1,2-dihidroxibenzeno)

[a] Compostos orgânicos tóxicos encontrados pelo ChemBioServ.

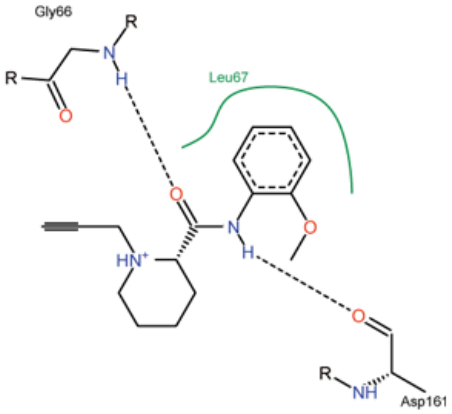
Tabela 2.4: Compostos selecionados no sub-conjunto.

Composto[a]	Posição no ranking	Localização no sítio	Nº ligações de hidrogênio
ZINC000252481569	3º	S2	1
ZINC000048253541	5º	S2	2
ZINC000097632071	7º	S1 e S2	1
ZINC000168598709	25º	S1 e S2	4
ZINC000221674213	29º	S1 e S2	1
ZINC000069951508	43º	S1 , S2 e S3	1

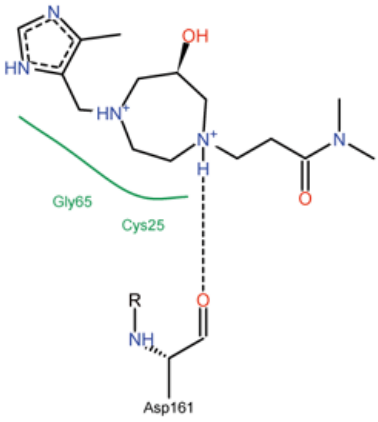
[a] Identificação do ZINC dos compostos selecionados.



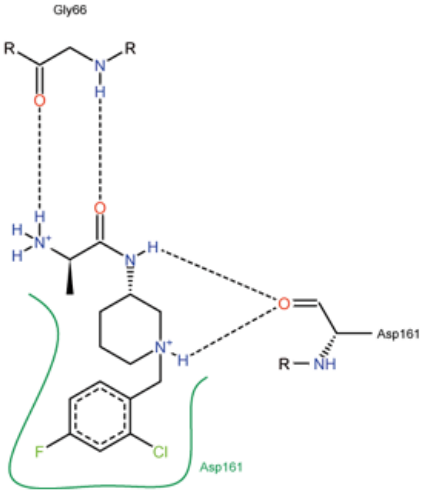
ZINC000252481569



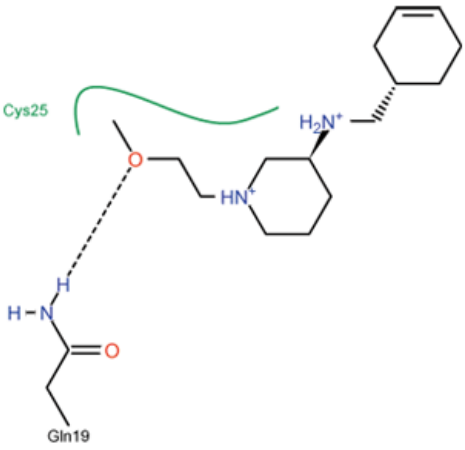
ZINC000048253541



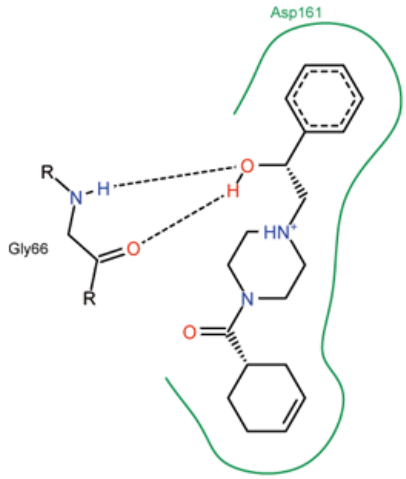
ZINC000097632071



ZINC000168598709

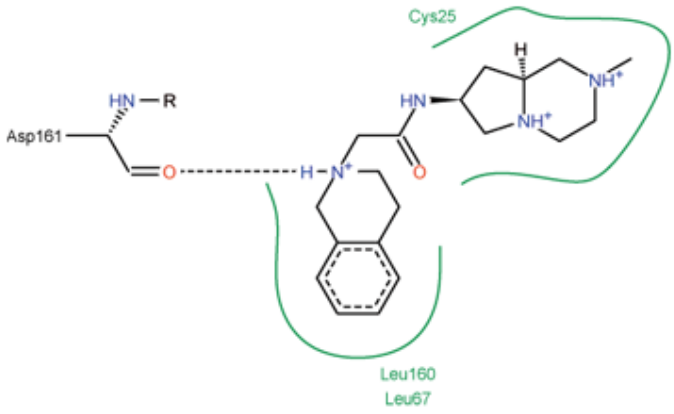


ZINC000221674213



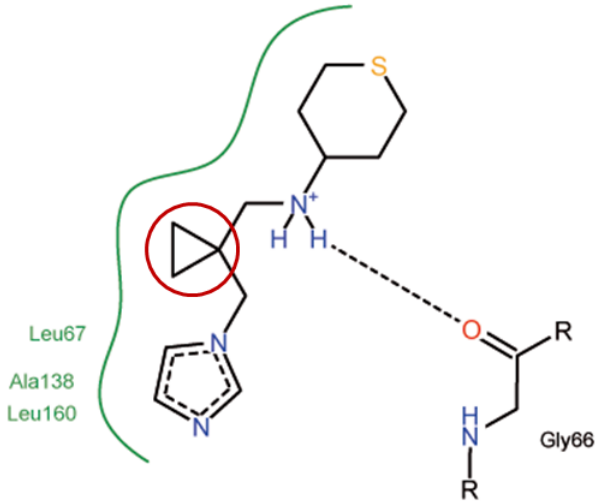
ZINC000069951508

Figura 2.21: **Interação dos compostos com o resíduos dos sub-sítios de ligação da enzima Cruzaína.** Ligações de hidrogênio são representados em forma de linha preta traceja.



ZINC000067842600

Figura 2.22: **Ligante descartado por estar positivamente carregado em ambiente hidrofóbico.**



ZINC000065463037

Figura 2.23: **Ligante com conformação anormal (conformação anormal destacada como um círculo em vermelho).**

2.5 Perspectivas

Para trabalhos futuros planejamos estudos envolvendo dinâmica molecular para prever a dinâmica e o comportamento energético dos compostos selecionados nesse trabalho e caso seja necessário gerar um segundo ranking.

Como os compostos selecionados nesse trabalho são de fácil e amplo acesso, pois foram avaliados apenas compostos de uso comercial e em estoque no banco de dados ZINC [53], experimentos de avaliação em laboratório poderiam facilmente ser realizados.

3 CONCLUSÕES

- Neste trabalho apresentamos a DeepVS, um método baseado em *deep learning* para melhoramento de *virtual screening* baseado em estrutura. Utilizando como entrada os dados de *docking* gerados pelo programa Autodockvina1.1.2, a DeepVS foi capaz de produzir o melhor valor da área sob a curva ROC (AUC) reportado na literatura para o banco de dados DUD.

- O presente trabalho apresenta ideias inovadoras de como modelar um complexo proteína-composto a partir do dado bruto (estrutura tridimensional) para ser utilizado em redes neurais profundas. A nossa estratégia de representar o complexo por um conjunto de contextos de átomos que são processados usando uma camada convolucional é original. Este trabalho também introduziu a ideia de *embeddings* de átomos e aminoácidos, que também podem ser utilizados em abordagens de *deep learning* para ajudar a solucionar outros problemas da biologia estrutural.

- De um modo geral, a DeepVS melhorou o resultado tanto de AUC quanto de fator de enriquecimento de dois programas de *docking*: Autodockvina1.1.2 e Dock6.6, demonstrando que a DeepVS é uma alternativa eficaz para melhoramento de *virtual screening* baseado em estrutura.

- A DeepVS obteve o melhor desempenho médio de *virtual screening* para o banco de dados DUD quando comparado aos resultados reportados por abordagens baseadas em redes neurais rasas como DDFA e NNScore.

- Mesmo utilizando um conjunto de treino menor (44 vs. 70 proteínas), a DeepVS obteve o melhor desempenho médio de *virtual screening* para o banco de dados DUD-E quando comparado aos resultados reportados por duas recentes abordagens baseadas em *deep learning*: cmpds ECFP + LR e AtomNet. A DeepVS possui um grande potencial para melhorar sua performance se mais dados forem adicionados ao conjunto de treino. Sistemas de *deep learning* são usualmente treinados com grandes volumes de dados.

- Os resultados dos nossos experimentos indicam que o uso da informação estrutural da proteína pela DeepVS está diretamente relacionado à qualidade das poses (conformações) que a DeepVS recebe como entrada.

- Apesar da DeepVS apresentar em média o melhor desempenho se comparada aos outros métodos reportados nesse trabalho, ela não foi constantemente a melhor abordagem de *virtual screening* para todos os receptores, o que reforça a hipótese de que a melhor abordagem de *virtual screening* depende diretamente do receptor a ser estudado.

- O programa de *docking* molecular Dock6.6 possui melhor performance para o receptor da Cruzaína contendo o par iônico (His 162)-NH^{+}/(Cys 25)-S^{-} enquanto a melhor performance do programa AutodockVina1.1.2 ocorre quando a Cys 25 pertencente a díade catalítica encontrava-se no seu estado neutro (His 162)-NH^{+}/(Cys 25)-SH. Apesar do estado de protonação dos resíduos da díade catalítica da enzima Cruzaína não ser totalmente definido, o melhor resultado de *virtual screening* reportado nesse trabalho foi para a Cys 25 em seu estado neutro o que corrobora com outros trabalhos presentes na literatura.

- Para o controle positivo desenvolvido para a enzima Cruzaína a DeepVS melhorou o resultado de *virtual screening* de ambos os programas de *docking* molecular AutodockVina1.1.2 e Dock6.6. Esse resultado consiste em mais uma evidência de que a DeepVS pode ser usada como uma estratégia efetiva para melhorar *virtual screening* baseado em estrutura.

- A abordagem proposta nesse trabalho demonstrou-se eficaz para a busca de candidatos a ligantes ativos para a enzima Cruzaína. Usando-se a DeeepVS para ranquear um conjunto de mais de 90 mil compostos verificamos que, dentre os 10 compostos melhor ranqueados, 3 possuem ligações de hidrogênio e estão localizados no sub-sítio S2 da Cruzaína.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Hecht D, Fogel GB. Computational intelligence methods for docking scores. Current Computer-Aided Drug Design. 2009;5(1):56–68.

[2] Arciniega M, Lange OF. Improvement of Virtual Screening Results by Docking Data Feature Analysis. Journal of Chemical Information and Modeling. 2014;54(5):1401–1411.

[3] Lill M. Virtual screening in drug design. In Silico Models for Drug Discovery. 2013;p. 1–12.

[4] Cheng T, Li Q, Zhou Z, Wang Y, Bryant S. Structure-Based Virtual Screening for Drug Discovery: a Problem-Centric Review. The AAPS journal. 2012;14(1):133–141.

[5] Huang SY, Grinter SZ, Zou X. Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions. Physical Chemistry Chemical Physics. 2010;12(40):12899–12908.

[6] Villoutreix BO, Eudes R, Miteva MA. Structure-based virtual ligand screening: recent success stories. Combinatorial chemistry & high throughput screening. 2009;12(10):1000–1016.

[7] Evers A, Klebe G. Successful virtual screening for a submicromolar antagonist of the neurokinin-1 receptor based on a ligand-supported homology model. Journal of medicinal chemistry. 2004;47(22):5381–5392.

[8] Cozza G, Bonvini P, Zorzi E, Poletto G, Pagano MA, Sarno S, et al. Identification of ellagic acid as potent inhibitor of protein kinase CK2: a successful example of a virtual screening application. Journal of medicinal chemistry. 2006;49(8):2363–2366.

[9] Lin TW, Melgar MM, Kurth D, Swamidass SJ, Purdon J, Tseng T, et al. Structure-based inhibitor design of AccD5, an essential acyl-CoA carboxylase carboxyltransferase domain of Mycobacterium tuberculosis. Proceedings of the National Academy of Sciences of the United States of America. 2006;103(9):3072–3077.

[10] Bonacci TM, Mathews JL, Yuan C, Lehmann DM, Malik S, Wu D, et al. Differential targeting of Gßγ-subunit signaling with small molecules. Science. 2006;312(5772):443–446.

[11] Vidal D, Blobel J, Pérez Y, Thormann M, Pons M. Structure-based discovery of new small molecule inhibitors of low molecular weight protein tyrosine phosphatase. European journal of medicinal chemistry. 2007;42(8):1102–1108.

[12] Kitchen DB, Decornez H, Furr JR, Bajorath J. Docking and Scoring in Virtual Screening for Drug Discovery: Methods and Applications. Nature Reviews Drug Discovery. 2004;3(11):935–949.

[13] Shoichet BK. Virtual screening of chemical libraries. Nature. 2004;432(7019):862–865.

[14] Durrant JD, McCammon JA. NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein-Ligand Complexes. Journal of Chemical Information and Modeling. 2010;50(10):1865–1871. PMID: 20845954.

[15] Kinnings SL, Liu N, Tonge PJ, Jackson RM, Xie L, Bourne PE. A Machine Learning-Based Method to Improve Docking Scoring Functions and Its Application to Drug Repurposing. Journal of Chemical Information and Modeling. 2011;51(2):408–419. Available from: http://dx.doi.org/10.1021/ci100369f.

[16] Ballester PJ, Mitchell JB. A Machine Learning Approach to Predicting Protein–Ligand Binding Affinity with Applications to Molecular Docking. Bioinformatics. 2010;26(9):1169–1175.

[17] Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2013;35(8):1798–1828.

[18] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–444.

[19] Bengio Y. Learning deep architectures for AI. Foundations and trends® in Machine Learning. 2009;2(1):1–127.

[20] Hawkins PC, Warren GL, Skillman AG, Nicholls A. How to do an evaluation: pitfalls and traps. Journal of computer-aided molecular design. 2008;22(3-4):179–190.

[21] Martin YC, Kofron JL, Traphagen LM. Do structurally similar molecules have similar biological activity? Journal of medicinal chemistry. 2002;45(19):4350–4358.

[22] Bender A, Glen RC. Molecular similarity: a key technique in molecular informatics. Organic & biomolecular chemistry. 2004;2(22):3204–3218.

[23] Johnson MA, Maggiora GM. Concepts and applications of molecular similarity. Wiley; 1990.

[24] Bajorath J. Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. Journal of chemical information and computer sciences. 2001;41(2):233–245.

[25] Livingstone DJ. The characterization of chemical structures using molecular properties. A survey. Journal of chemical information and computer sciences. 2000;40(2):195–209.

[26] Zoete V, Daina A, Bovigny C, Michielin O. SwissSimilarity: A Web Tool for Low to Ultra High Throughput Ligand-Based Virtual Screening. Journal of Chemical Information and Modeling. 2016;56(8):1399–1404.

[27] Lipinski CA, Lombardo F, Dominy BW, Feeney PJ. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced drug delivery reviews. 1997;23(1-3):3–25.

[28] Willett P. Similarity-based virtual screening using 2D fingerprints. Drug discovery today. 2006;11(23):1046–1053.

[29] Cao DS, Liang YZ, Deng Z, Hu QN, He M, Xu QS, et al. Genome-Scale Screening of Drug-Target Associations Relevant to K i Using a Chemogenomics Approach. PloS one. 2013;8(4):e57680.

[30] Ehrlich P. Über den jetzigen Stand der Chemotherapie. Berichte der deutschen chemischen Gesellschaft. 1909;42(1):17–47.

[31] Kumar V, Krishna S, Siddiqi MI. Virtual screening strategies: Recent advances in the identification and design of anti-cancer agents. Methods. 2015;71:64–70.

[32] Purushottamachar P, Khandelwal A, Chopra P, Maheshwari N, Gediya LK, Vasaitis TS, et al. First pharmacophore-based identification of androgen receptor down-regulating agents: discovery of potent anti-prostate cancer agents. Bioorganic & medicinal chemistry. 2007;15(10):3413–3421.

[33] SilicosIT [homepage on the Internet]. Alignit; [cited 2016 Dec 07]. Available from: `http://silicos-it.be.s3-website-eu-west-1.amazonaws.com/#`.

[34] Mohammed A, Glen RC, et al. Applications of rule-induction in the derivation of quantitative structure-activity relationships. Journal of Computer-Aided Molecular Design. 1992;6(4):349–383.

[35] Byvatov E, Fechner U, Sadowski J, Schneider G. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. Journal of Chemical Information and Computer Sciences. 2003;43(6):1882–1889.

[36] Jandu KS, Barrett V, Brockwell M, Cambridge D, Farrant DR, Foster C, et al. Discovery of 4-[3-(trans-3-Dimethylaminocyclobutyl)-1 H-indol-5-ylmethyl]-(4 S)-oxazolidin-2-one (4991W93), a 5HT1B/1D Receptor Partial Agonist and a Potent Inhibitor of Electrically Induced Plasma Extravasation. Journal of medicinal chemistry. 2001;44(5):681–693.

[37] Bender A, Mussa HY, Glen RC, Reiling S. Molecular similarity searching using atom environments, information-based feature selection, and a naive Bayesian classifier. Journal of chemical information and computer sciences. 2004;44(1):170–178.

[38] Merck Molecular Activity Challenge [homepage on the Internet]. Merk; 2012 [updated 2012 Oct 16; cited 2016 Nov 15]. Available from: https://www.kaggle.com/c/MerckActivity.

[39] Schneidman-Duhovny D, Nussinov R, Wolfson HJ. Predicting molecular interactions in silico: II. Protein-protein and protein-drug docking. Current medicinal chemistry. 2004;11(1):91–107.

[40] Schapira M, Raaka BM, Das S, Fan L, Totrov M, Zhou Z, et al. Discovery of diverse thyroid hormone receptor antagonists by high-throughput docking. Proceedings of the National Academy of Sciences. 2003;100(12):7354–7359.

[41] Evers A, Klebe G. Ligand-Supported Homology Modeling of G-Protein-Coupled Receptor Sites: Models Sufficient for Successful Virtual Screening. Angewandte Chemie International Edition. 2004;43(2):248–251.

[42] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Research. 2000;28(1):235–242. Available from: http://nar.oxfordjournals.org/content/28/1/235.abstract.

[43] Lyne PD. Structure-based virtual screening: an overview. Drug discovery today. 2002;7(20):1047–1055.

[44] Waszkowycz B. Towards improving compound selection in structure-based virtual screening. Drug discovery today. 2008;13(5):219–226.

[45] Ghosh S, Nie A, An J, Huang Z. Structure-based virtual screening of chemical libraries for drug discovery. Current opinion in chemical biology. 2006;10(3):194–202.

[46] Wolber G, Langer T. LigandScout: 3-D pharmacophores derived from protein-bound ligands and their use as virtual screening filters. Journal of chemical information and modeling. 2005;45(1):160–169.

[47] Salam NK, Nuti R, Sherman W. Novel method for generating structure-based pharmacophores using energetic analysis. Journal of chemical information and modeling. 2009;49(10):2356–2368.

[48] Kim S, Thiessen PA, Bolton EE, Chen J, Fu G, Gindulyte A, et al. PubChem substance and compound databases. Nucleic acids research. 2015;p. gkv951.

[49] Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, et al. The ChEMBL bioactivity database: an update. Nucleic acids research. 2014;42(D1):D1083–D1090.

[50] National Cancer Institute's Developmental Therapeutics Program [homepage on the Internet]. National Cancer Institute; 2012 [updated 2015 Nov 16; cited 2017 Jan 09]. Available from: https://dtp.cancer.gov/organization/dscb/obtaining/default.htm.

[51] ChemSpider SyntheticPages [homepage on the Internet]. ChemSpider SyntheticPages; 2001 [updated 2017; cited 2017 Feb 06]. Available from: http://cssp.chemspider.com/123.

[52] Accelerating time to market and driving innovation with collaboration, knowledge based under-standing and prediction [homepage on the Internet]. BIOVIA; [updated 2017; cited 2017 Feb 06]. Available from: http://accelrys.com/products/collaborative-science/databases/bioactivity-databases/mddr.html.

[53] Sterling T, Irwin JJ. ZINC 15-Ligand discovery for everyone. 2015;.

[54] Pieper U, Eswar N, Davis FP, Braberg H, Madhusudhan MS, Rossi A, et al. MODBASE: a database of annotated comparative protein structure models and associated resources. Nucleic acids research. 2006;34(suppl 1):D291–D295.

[55] An J, Totrov M, Abagyan R. Pocketome via comprehensive identification and classification of ligand binding envelopes. Molecular & Cellular Proteomics. 2005;4(6):752–761.

[56] Rapp CS, Schonbrun C, Jacobson MP, Kalyanaraman C, Huang N. Automated site preparation in physics-based rescoring of receptor ligand complexes. Proteins: Structure, Function, and Bioinformatics. 2009;77(1):52–61.

[57] Ten Brink T, Exner TE. pKa based protonation states and microspecies for protein–ligand docking. Journal of computer-aided molecular design. 2010;24(11):935–942.

[58] Scior T, Bender A, Tresadern G, Medina-Franco JL, Martínez-Mayorga K, Langer T, et al. Recognizing pitfalls in virtual screening: a critical review. Journal of chemical information and modeling. 2012;52(4):867–881.

[59] Hooft R, Vriend G, Sander C, Abola EE, et al. Errors in protein structures. Nature. 1996;381(6580):272–272.

[60] Laskowski RA, MacArthur MW, Moss DS, Thornton JM. PROCHECK: a program to check the stereochemical quality of protein structures. Journal of applied crystallography. 1993;26(2):283–291.

[61] Durrant JD, McCammon JA. NNScore 2.0: A Neural-Network Receptor-Ligand Scoring Function. Journal of Chemical Information and Modeling. 2011;51(11):2897–2903.

[62] Brooijmans N, Kuntz ID. Molecular recognition and docking algorithms. Annual review of biophysics and biomolecular structure. 2003;32(1):335–373.

[63] Trosset JY, Scheraga HA. Reaching the global minimum in docking simulations: a Monte Carlo energy minimization approach using Bezier splines. Proceedings of the National Academy of Sciences. 1998;95(14):8011–8015.

[64] Huang N, Shoichet BK, Irwin JJ. Benchmarking Sets for Molecular Docking. Journal of medicinal chemistry. 2006;49(23):6789–6801.

[65] Marrone TJ, Briggs JM and, McCammon JA. Structure-based drug design: computational advances. Annual review of pharmacology and toxicology. 1997;37(1):71–90.

[66] Warren GL, Andrews CW, Capelli AM, Clarke B, LaLonde J, Lambert MH, et al. A critical assessment of docking programs and scoring functions. Journal of medicinal chemistry. 2006;49(20):5912–5931.

[67] Durrant JD, Friedman AJ, Rogers KE, McCammon JA. Comparing Neural-Network Scoring Functions and the State of the Art: Applications to Common Library Screening. Journal of Chemical Information and Modeling. 2013;53(7):1726–1735.

[68] Marsland S. Machine learning: an algorithmic perspective. CRC press; 2015.

[69] Mitchell TM, et al.. Machine learning. WCB. McGraw-Hill Boston, MA:; 1997.

[70] Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of computational chemistry. 2010;31(2):455–461.

[71] Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. Journal of medicinal chemistry. 2004;47(7):1739–1749.

[72] Sadowski J, Kubinyi H. A scoring scheme for discriminating between drugs and nondrugs. Journal of medicinal chemistry. 1998;41(18):3325–3329.

[73] Ashtawy HM, Mahapatra NR. BgN-Score and BsN-Score: Bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein-ligand complexes. BMC bioinformatics. 2015;16(4):1.

[74] Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. Journal of computational chemistry. 2009;30(16):2785–2791.

[75] Meiler J, Baker D. ROSETTALIGAND: Protein small molecule docking with full side chain flexibility. Proteins: Structure, Function, and Bioinformatics. 2006;65(3):538–548.

[76] Nielsen MA. Neural networks and deep learning. URL: http://neuralnetworksanddeeplearning com/(visited: 1027 2016). 2015;.

[77] Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks. 2015;61:85–117.

[78] Bengio Y, Goodfellow IJ, Courville A. Deep learning. An MIT Press book in preparation Draft chapters available at http://www iro umontreal ca/ bengioy/dlbook. 2015;.

[79] Alpaydin E. Introduction to machine learning. MIT press; 2014.

[80] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016. `http://www.deeplearningbook.org`.

[81] Rumelhart DE, Hinton GE, Williams RJ. Learning Representations by Back-Propagating Errors. Cognitive Modeling. 1988;5(3):1.

[82] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review. 1958;65(6):386.

[83] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE. 1998;86(11):2278–2324.

[84] Jones N, et al. The learning machines. Nature. 2014;505(7482):146–148.

[85] Le Cun BB, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems. Citeseer; 1990. .

[86] LeCun Y, et al. Generalization and network design strategies. Connectionism in perspective. 1989;p. 143–155.

[87] Zhou Y, Chellappa R. Computation of optical flow using a neural network. In: IEEE International Conference on Neural Networks. vol. 1998; 1988. p. 71–78.

[88] Boureau YL, Ponce J, LeCun Y. A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th international conference on machine learning (ICML-10); 2010. p. 111–118.

[89] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10); 2010. p. 807–814.

[90] Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ. Phoneme recognition using time-delay neural networks. IEEE transactions on acoustics, speech, and signal processing. 1989;37(3):328–339.

[91] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–1105.

[92] Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014;15(1):1929–1958.

[93] CireşAn D, Meier U, Masci J, Schmidhuber J. Multi-column deep neural network for traffic sign classification. Neural Networks. 2012;32:333–338.

[94] Taigman Y, Yang M, Ranzato M, Wolf L. Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2014. p. 1701–1708.

[95] Ning F, Delhomme D, LeCun Y, Piano F, Bottou L, Barbano PE. Toward automatic phenotyping of developing embryos from videos. IEEE Transactions on Image Processing. 2005;14(9):1360–1371.

[96] Pereira JC, Caffarena ER, dos Santos CN. Boosting Docking-based Virtual Screening with Deep Learning. Journal of Chemical Information and Modeling. 2016;.

[97] Gonczarek A, Tomczak JM, Zaręba S, Kaczmar J, Dąbrowski P, Walczak MJ. Learning Deep Architectures for Interaction Prediction in Structure-based Virtual Screening. arXiv preprint arXiv:161007187. 2016;.

[98] dos Santos CN, Gatti M. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In: COLING; 2014. p. 69–78.

[99] dos Santos CN, Zadrozny B. Learning Character-Level Representations for Part-of-Speech Tagging. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14); 2014. p. 1818–1826.

[100] Hadsell R, Sermanet P, Ben J, Erkan A, Scoffier M, Kavukcuoglu K, et al. Learning long-range vision for autonomous off-road driving. Journal of Field Robotics. 2009;26(2):120–144.

[101] Farabet C, Couprie C, Najman L, LeCun Y. Scene parsing with multiscale feature learning, purity trees, and optimal covers. arXiv preprint arXiv:12022160. 2012;.

[102] Dahl GE, Jaitly N, Salakhutdinov R. Multi-Task Neural Networks for QSAR Predictions. arXiv preprint arXiv:14061231. 2014;.

[103] Unterthiner T, Mayr A, Klambauer G, Steijaert M, Wegner JK, Ceulemans H, et al. Deep Learning as an Opportunity in Virtual Screening. In: Proceedings of the Deep Learning Workshop at NIPS; 2014. .

[104] Unterthiner T, Mayr A, Klambauer G, Hochreiter S. Toxicity Prediction Using Deep Learning. arXiv preprint arXiv:150301445. 2015;.

[105] Ramsundar B, Kearnes S, Riley P, Webster D, Konerding D, Pande V. Massively Multitask Networks for Drug Discovery. arXiv preprint arXiv:150202072. 2015;.

[106] Lusci A, Pollastri G, Baldi P. Deep Architectures and Deep Learning in Chemoinformatics: the Prediction of Aqueous Solubility for Drug-like Molecules. Journal of Chemical Information and Modeling. 2013;53(7):1563–1575.

[107] Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, et al. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In: Advances in Neural Information Processing Systems; 2015. p. 2215–2223.

[108] Weber J, Achenbach J, Moser D, Proschak E. VAMMPIRE: a Matched Molecular Pairs Database for Structure-Based Drug Design and Optimization. Journal of medicinal chemistry. 2013;56(12):5203–5207.

[109] Artemenko N. Distance dependent scoring function for describing protein-ligand intermolecular interactions. Journal of chemical information and modeling. 2008;48(3):569–574.

[110] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems (NIPS). 2013;.

[111] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural Language Processing (Almost) from Acratch. The Journal of Machine Learning Research. 2011;12:2493–2537.

[112] Socher R, Huval B, Manning CD, Ng AY. Semantic Compositionality Through Recursive Matrix-Vector Spaces. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics; 2012. p. 1201–1211.

[113] Cereto-Massagué A, Guasch L, Valls C, Mulero M, Pujadas G, Garcia-Vallvé S. DecoyFinder: an easy-to-use python GUI application for building target-specific decoy sets. Bioinformatics. 2012;28(12):1661–1662.

[114] Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, et al. Theano: a CPU and GPU Math Expression Compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy). vol. 4. Austin, TX; 2010. p. 3.

[115] Irwin JJ, Sterling T, Mysinger MM, Bolstad ES, Coleman RG. ZINC: a free tool to discover chemistry for biology. Journal of chemical information and modeling. 2012;52(7):1757–1768.

[116] Armstrong MS, Morris GM, Finn PW, Sharma R, Moretti L, Cooper RI, et al. ElectroShape: Fast Molecular Similarity Calculations Incorporating Shape, Chirality and Electrostatics. Journal of Computer-aided Molecular Design. 2010;24(9):789–801.

[117] Answers on duplicate IDs [homepage on the Internet]. DUDE; 2014 [cited 2016 dec 22]. Available from: http://wiki.bkslab.org/index.php/DUDE.

[118] Lang PT, Brozell SR, Mukherjee S, Pettersen EF, Meng EC, Thomas V, et al. DOCK 6: Combining Techniques to Model RNA–Small Molecule Complexes. Rna. 2009;15(6):1219–1230.

[119] Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. Journal of medicinal chemistry. 2012;55(14):6582–6594.

[120] Meng EC, Shoichet BK, Kuntz ID. Automated docking with grid-based energy evaluation. Journal of computational chemistry. 1992;13(4):505–524.

[121] Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, et al. UCSF Chimera - A Visualization System for Exploratory esearch and Analysis. Journal of Computational Chemistry. 2004;25(13):1605–1612.

[122] Dock Prep [homepage on the Internet]. UCSF Computer Graphics Laboratory; 2007 [updated 2016; cited 2016 Dec 27]. Available from: http://www.cgl.ucsf.edu/chimera/1.2199/docs/ContributedSoftware/dockprep/dockprep.html.

[123] O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open Babel: An open chemical toolbox. Journal of cheminformatics. 2011;3(1):1.

[124] Gasteiger J, Saller H. Calculation of the charge distribution in conjugated systems by a quantification of the resonance concept. Angewandte Chemie International Edition in English. 1985;24(8):687–689.

[125] Richards FM. Areas, volumes, packing, and protein structure. Annual review of biophysics and bioengineering. 1977;6(1):151–176.

[126] Kuntz ID, Blaney JM, Oatley SJ, Langridge R, Ferrin TE. A geometric approach to macromolecule-ligand interactions. Journal of molecular biology. 1982;161(2):269–288.

[127] Generating Spheres [homepage on the Internet]. P. Therese Lang; [updated 2015 Feb 09; cited 2016 Dec 27]. Available from: http://dock.compbio.ucsf.edu/DOCK_6/tutorials/sphere_generation/generating_spheres.htm.

[128] Ewing TJ, Makino S, Skillman AG, Kuntz ID. DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. Journal of computer-aided molecular design. 2001;15(5):411–428.

[129] Wang R, Lai L, Wang S. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. Journal of computer-aided molecular design. 2002;16(1):11–26.

[130] Baxter J. Local optima avoidance in depot location. Journal of the Operational Research Society. 1981;32(9):815–819.

[131] Andrea R, Blesa M, Blum C, Michael S. Hybrid metaheuristics–an emerging approach to optimization. 2008;.

[132] Arlot S, Celisse A. A Survey of Cross-Validation Procedures for Model Selection. Statistics Surveys. 2010;4:40–79.

[133] Jahn A, Rosenbaum L, Hinselmann G, Zell A. 4D Flexible Atom-Pairs: An Efficient Probabilistic Conformational Space Comparison for Ligand-Based Virtual Screening. J Cheminformatics. 2011;3:23.

[134] Nicholls A. What do We Know and when do We Know It? Journal of Computer-Aided Molecular Design. 2008;22(3-4):239–255.

[135] Wallach I, Lilien R. Virtual decoy sets for molecular docking benchmarks. Journal of chemical information and modeling. 2011;51(2):196–202.

[136] Cross JB, Thompson DC, Rai BK, Baber JC, Fan KY, Hu Y, et al. Comparison of Several Molecular Docking Programs: Pose Prediction and Virtual Screening Accuracy. Journal of Chemical Information and Modeling. 2009;49(6):1455–1474.

[137] Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA. Development and testing of a general amber force field. Journal of computational chemistry. 2004;25(9):1157–1174.

[138] Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, et al. The Amber biomolecular simulation programs. Journal of computational chemistry. 2005;26(16):1668–1688.

[139] Graves AP, Shivakumar DM, Boyce SE, Jacobson MP, Case DA, Shoichet BK. Rescoring docking hit lists for model cavity sites: predictions and experimental testing. Journal of molecular biology. 2008;377(3):914–934.

[140] Brozell SR, Mukherjee S, Balius TE, Roe DR, Case DA, Rizzo RC. Evaluation of DOCK 6 as a pose generation and database enrichment tool. Journal of computer-aided molecular design. 2012;26(6):749–773.

[141] Kolossváry I, Guida WC. Low mode search. An efficient, automated computational method for conformational analysis: Application to cyclic and acyclic alkanes and cyclic peptides. Journal of the American Chemical Society. 1996;118(21):5011–5019.

[142] Halgren TA, Murphy RB, Friesner RA, Beard HS, Frye LL, Pollard WT, et al. Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. Journal of medicinal chemistry. 2004;47(7):1750–1759.

[143] Friesner RA, Murphy RB, Repasky MP, Frye LL, Greenwood JR, Halgren TA, et al. Extra precision glide: docking and scoring incorporating a model of hydrophobic enclosure for protein-ligand complexes. Journal of medicinal chemistry. 2006;49(21):6177–6196.

[144] Durrant JD, McCammon JA. BINANA: a novel algorithm for ligand-binding characterization. Journal of Molecular Graphics and Modelling. 2011;29(6):888–893.

[145] Cummings MD, DesJarlais RL, Gibbs AC, Mohan V, Jaeger EP. Comparison of automated docking programs as virtual screening tools. Journal of medicinal chemistry. 2005;48(4):962–976.

[146] Perola E, Walters WP, Charifson PS. A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. Proteins: Structure, Function, and Bioinformatics. 2004;56(2):235–249.

[147] McGaughey GB, Sheridan RP, Bayly CI, Culberson JC, Kreatsoulas C, Lindsley S, et al. Comparison of topological, shape, and docking methods in virtual screening. Journal of chemical information and modeling. 2007;47(4):1504–1519.

[148] Neves MA, Totrov M, Abagyan R. Docking and Scoring with ICM: The Benchmarking Results and Strategies for Improvement. Journal of Computer-Aided Molecular Design. 2012;26(6):675–686.

[149] Wallach I, Dzamba M, Heifets A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv preprint arXiv:151002855. 2015;.

[150] Bhopal A, Callender T, Knox AF, Regmi S. Strength in numbers? Grouping, fund allocation and coordination amongst the neglected tropical diseases. Journal of global health. 2013;3(2).

[151] WORLD HEALTH ORGANIZATION [homepage on the Internet]. WHO; 2016 [updated 2017; cited 2017 Feb 15]. Available from: `http://www.who.int/neglected_diseases/diseases/en/`.

[152] Yamey G. The world's most neglected diseases-Ignored by the pharmaceutical industry and by public-private partnerships. BRITISH MED JOURNAL PUBL GROUP BRITISH MED ASSOC HOUSE, TAVISTOCK SQUARE, LONDON WC1H 9JR, ENGLAND; 2002.

[153] Vanderelst D, Speybroeck N. Quantifying the lack of scientific interest in neglected tropical diseases. PLoS Negl Trop Dis. 2010;4(1):e576.

[154] Trouiller P, Olliaro P, Torreele E, Orbinski J, Laing R, Ford N. Drug development for neglected diseases: a deficient market and a public-health policy failure. The Lancet. 2002;359(9324):2188–2194.

[155] Cavalier-Smith T. Kingdom protozoa and its 18 phyla. Microbiological reviews. 1993;57(4):953–994.

[156] Imam T. The complexities in the classification of protozoa: a challenge to parasitologists. Bayero Journal of Pure and Applied Sciences. 2009;2(2):159–164.

[157] Cavalier-Smith T. Kingdoms Protozoa and Chromista and the eozoan root of the eukaryotic tree. Biology letters. 2010;p. rsbl20090948.

[158] Ouellette M. Biochemical and molecular mechanisms of drug resistance in parasites. Tropical Medicine & International Health. 2001;6(11):874–882.

[159] Working to overcome the global impact of neglected tropical diseases. First WHO report on neglected tropical diseases [homepage on the Internet]. WHO; 2010 [updated 2010; cited 2017 Jul 15]. Available from: http://whqlibdoc.who.int/publications/2010/9789241564090_eng.pdf.

[160] Pereira PCM, Navarro EC. Challenges and perspectives of Chagas disease: a review. Journal of venomous animals and toxins including tropical diseases. 2013;19(1):34.

[161] Ribeiro AL, Nunes MP, Teixeira MM, Rocha MO. Diagnosis and management of Chagas disease and cardiomyopathy. Nature Reviews Cardiology. 2012;9(10):576–589.

[162] Organization WH, et al. Investing to overcome the global impact of neglected tropical diseases: third WHO report on neglected diseases 2015. World Health Organization; 2015.

[163] Coura JR, De Castro SL. A critical review on Chagas disease chemotherapy. Memórias do Instituto Oswaldo Cruz. 2002;97(1):3–24.

[164] Brener Z. Trypanosoma cruzi: morfologia e ciclo evolutivo. JC Pinto Dias, JR. 1997;.

[165] Martins AV, Gomes AP, de Mendonça EG, Fietto JLR, Santana LA, de Almeida Oliveira MG, et al. Biology of Trypanosoma cruzi: An update. Infectio. 2012;16(1):45–58.

[166] Parasites - American Trypanosomiasis (also known as Chagas Disease) [homepage on the Internet]. Centers for Disease Control and Prevention; 2016 [updated 2016; cited March 2017]. Available from: https://www.cdc.gov/parasites/chagas/biology.html.

[167] Ley V, Andrews NW, Robbins ES, Nussenzweig V. Amastigotes of Trypanosoma cruzi sustain an infective cycle in mammalian cells. The Journal of experimental medicine. 1988;168(2):649–659.

[168] Picka MCM, Meira DA, Carvalho TBd, Peresi E, Marcondes-Machado J. Definition of a diagnostic routine in individuals with inconclusive serology for Chagas disease. Brazilian Journal of Infectious Diseases. 2007;11(2):226–233.

[169] Chagas C. Processos patojenicos da tripanozomiase americana. Memórias do Instituto Oswaldo Cruz. 1916;8(2):5–36.

[170] Coura JR, Borges-Pereira J. Chronic phase of Chagas disease: why should it be treated? A comprehensive review. Memorias do Instituto Oswaldo Cruz. 2011;106(6):641–645.

[171] Castro JA, deMecca MM, Bartel LC. Toxic side effects of drugs used to treat Chagas' disease (American trypanosomiasis). Human & experimental toxicology. 2006;25(8):471–479.

[172] Coura JR, Borges-Pereira J. Chagas disease: What is known and what should be improved: a systemic review. Revista da Sociedade Brasileira de Medicina Tropical. 2012;45(3):286–296.

[173] Coura J, De Abreu L, Willcox H, Petana W. Comparative controlled study on the use of benznidazole, nifurtimox and placebo, in the chronic form of Chagas' disease, in a field area with interrupted transmission. I. Preliminary evaluation. Revista da Sociedade Brasileira de Medicina Tropical. 1996;30(2):139–144.

[174] Molécula inibe crescimento de parasita da Doença de Chagas [homepage on the Internet]. USP; 2014 [updated 2014 Jun 13; cited 2017 Mar 07]. Available from: http://www.usp.br/agen/?p=179334.

[175] Cazzulo J. Proteinases of Trypanosoma cruzi: potential targets for the chemotherapy of Chagas disease. Current topics in medicinal chemistry. 2002;2(11):1261–1271.

[176] Powers JC, Asgian JL, Ekici ÖD, James KE. Irreversible inhibitors of serine, cysteine, and threonine proteases. Chemical reviews. 2002;102(12):4639–4750.

[177] Choe Y, Leonetti F, Greenbaum DC, Lecaille F, Bogyo M, Brömme D, et al. Substrate profiling of cysteine proteases using a combinatorial peptide library identifies functionally unique specificities. Journal of Biological Chemistry. 2006;281(18):12824–12832.

[178] Siklos M, BenAissa M, Thatcher GR. Cysteine proteases as therapeutic targets: does selectivity matter? A systematic review of calpain and cathepsin inhibitors. Acta Pharmaceutica Sinica B. 2015;5(6):506–519.

[179] Schechter I, Berger A. On the size of the active site in proteases. I. Papain. Biochemical and biophysical research communications. 1967;27(2):157–162.

[180] Turk B. Targeting proteases: successes, failures and future prospects. Nature reviews Drug discovery. 2006;5(9):785–799.

[181] Engel JC, Doyle PS, Hsieh I, McKerrow JH. Cysteine protease inhibitors cure an experimental Trypanosoma cruzi infection. Journal of Experimental Medicine. 1998;188(4):725–734.

[182] Scharfstein J, Schechter M, Senna M, Peralta JM, Mendonça-Previato L, Miles MA. Trypanosoma cruzi: characterization and isolation of a 57/51,000 mw surface glycoprotein (GP57/51) expressed by epimastigotes and bloodstream trypomastigotes. The Journal of Immunology. 1986;137(4):1336–1341.

[183] Martinez J, Campetella O, Frasch A, Cazzulo JJ. The major cysteine proteinase (cruzipain) from Trypanosoma cruzi is antigenic in human infections. Infection and immunity. 1991;59(11):4275–4277.

[184] Fampa P, Lisboa C, Jansen A, Santos A, Ramirez M. Protease expression analysis in recently field-isolated strains of Trypanosoma cruzi: a heterogeneous profile of cysteine protease activities between TC I and TC II major phylogenetic groups. Parasitology. 2008;135(09):1093–1100.

[185] McGrath ME, Eakin AE, Engel JC, McKerrow JH, Craik CS, Fletterick RJ. The crystal structure of cruzain: a therapeutic target for Chagas' disease. Journal of molecular biology. 1995;247(2):251–259.

[186] Sajid M, Robertson SA, Brinen LS, McKerrow JH. Cruzain. In: Cysteine Proteases of Pathogenic Organisms. Springer; 2011. p. 100–115.

[187] Gillmor SA, Craik CS, Fletterick RJ. Structural determinants of specificity in the cysteine protease cruzain. Protein Science. 1997;6(8):1603–1611.

[188] Yang PY, Wang M, Li L, Wu H, He CY, Yao SQ. Design, Synthesis and Biological Evaluation of Potent Azadipeptide Nitrile Inhibitors and Activity-Based Probes as Promising Anti-Trypanosoma brucei Agents. Chemistry–A European Journal. 2012;18(21):6528–6541.

[189] Ettari R, Tamborini L, Angelo IC, Micale N, Pinto A, De Micheli C, et al. Inhibition of rhodesain as a novel therapeutic modality for human African trypanosomiasis. J Med Chem. 2013;56(14):5637–5658.

[190] Huang L, Brinen LS, Ellman JA. Crystal structures of reversible ketone-based inhibitors of the cysteine protease cruzain. Bioorganic & medicinal chemistry. 2003;11(1):21–29.

[191] Ferreira RS, Simeonov A, Jadhav A, Eidam O, Mott BT, Keiser MJ, et al. Complementarity between a docking and a high-throughput screen in discovering new cruzain inhibitors. Journal of medicinal chemistry. 2010;53(13):4891–4905.

[192] Wiggers HJ, Rocha JR, Fernandes WB, Sesti-Costa R, Carneiro ZA, Cheleski J, et al. Non-peptidic cruzain inhibitors with trypanocidal activity discovered by virtual screening and in vitro assay. PLoS Negl Trop Dis. 2013;7(8):e2370.

[193] Jones BD, Tochowicz A, Tang Y, Cameron MD, McCall LI, Hirata K, et al. Synthesis and evaluation of oxyguanidine analogues of the cysteine protease inhibitor WRR-483 against Cruzain. ACS medicinal chemistry letters. 2015;7(1):77–82.

[194] Schrödinger, LLC. The PyMOL Molecular Graphics System, Version 1.8; 2015.

[195] Stierand K, Rarey M. Drawing the PDB: protein- ligand complexes in two dimensions. ACS medicinal chemistry letters. 2010;1(9):540–545.

[196] Wallace AC, Laskowski RA, Thornton JM. LIGPLOT: a program to generate schematic diagrams of protein-ligand interactions. Protein engineering. 1995;8(2):127–134.

[197] Santos VC. Developing selective cruzain inhibitors through structure-based techniques [Dissertação, Mestrado em Bioquímica e Imunologia]. Instituto de Ciências Biológicas da UFMG; 2017.

[198] Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic acids research. 2016;44(D1):D1045–D1053.

[199] Du X, Hansell E, Engel JC, Caffrey CR, Cohen FE, McKerrow JH. Aryl ureas represent a new class of anti-trypanosomal agents. Chemistry & biology. 2000;7(9):733–742.

[200] Ferreira RS, Dessoy MA, Pauli I, Souza ML, Krogh R, Sales AI, et al. Synthesis, Biological Evaluation, and Structure–Activity Relationships of Potent Noncovalent and Nonpeptidic Cruzain Inhibitors as Anti-Trypanosoma cruzi Agents. Journal of medicinal chemistry. 2014;57(6):2380–2392.

[201] Rogers KE, Keränen H, Durrant JD, Ratnam J, Doak A, Arkin MR, et al. Novel cruzain inhibitors for the treatment of Chagas' disease. Chemical biology & drug design. 2012;80(3):398–405.

[202] Olsson MH, Søndergaard CR, Rostkowski M, Jensen JH. PROPKA3: consistent treatment of internal and surface residues in empirical p K a predictions. Journal of Chemical Theory and Computation. 2011;7(2):525–537.

[203] Yin RK. Estudo de caso: planejamento e métodos. Tradução Daniel Grassi. Porto Alegre: Bookman; 2005.

[204] Cstorer A, Ménard R. [33] Catalytic mechanism in papain family of cysteine peptidases. In: Proteolytic Enzymes: Serine and Cysteine Peptidases. vol. 244 of Methods in Enzymology. Academic Press; 1994. p. 486 – 500. Available from: http://www.sciencedirect.com/science/article/pii/0076687994440352.

[205] Otto HH, Schirmeister T. Cysteine proteases and their inhibitors. Chemical reviews. 1997;97(1):133–172.

[206] Sárkány Z, Szeltner Z, Polgár L. Thiolate-imidazolium ion pair is not an obligatory catalytic entity of cysteine peptidases: the active site of picornain 3C. Biochemistry. 2001;40(35):10601–10606.

[207] Shokhen M, Khazanov N, Albeck A. Challenging a paradigm: Theoretical calculations of the protonation state of the Cys25-His159 catalytic diad in free papain. Proteins: Structure, Function, and Bioinformatics. 2009;77(4):916–926.

[208] Dos Santos LHS. Developing selective cruzain inhibitors through structure-based techniques [Tese, Doutorado em Biologia Computacional e Sistemas]. Fundação Oswaldo Cruz; 2016.

[209] Athanasiadis E, Cournia Z, Spyrou G. ChemBioServer: A web-based pipeline for filtering, clustering and visualization of chemical compounds used in drug discovery. Bioinformatics. 2012;28(22):3002–3003.

[210] Berendsen HJ, van der Spoel D, van Drunen R. GROMACS: a message-passing parallel molecular dynamics implementation. Computer Physics Communications. 1995;91(1-3):43–56.

[211] Kellenberger E, Springael JY, Parmentier M, Hachet-Haas M, Galzi JL, Rognan D. Identification of nonpeptide CCR5 receptor agonists by structure-based virtual screening. Journal of medicinal chemistry. 2007;50(6):1294–1303.

[212] Li H, Robertson AD, Jensen JH. Very fast empirical prediction and rationalization of protein pKa values. Proteins: Structure, Function, and Bioinformatics. 2005;61(4):704–721.

[213] Schrödinger Release 2016-4: Maestro [homepage on the Internet]. Schrödinger; 2016 [updated 2016; cited 2017 Feb 07]. Available from: `https://www.schrodinger.com/freemaestro/`.

[214] Morrot A. The Role of Sialic Acid-Binding Receptors (Siglecs) in the Immunomodulatory Effects of Trypanosoma cruzi Sialoglycoproteins on the Protective Immunity of the Host. Scientifica. 2013;2013.

APÊNDICE A

A.1 Resumo publicado na ISCB - Latin American X-Meeting on Bioinformatics with BSB & SoiBio, 2014

A VIRTUAL SCREENING RESCORING SCHEME BASED ON DEEP LEARNING

J. C. Pereira[1], C. N. dos Santos[2], E. R. Caffarena[1]

[1]Programa de Computação Científica. Presidência/Fiocruz.
[2]IBM Research.

Background: Virtual screening (VS) is a low cost alternative to assist the process of new drug development. This work focuses in the structured-based VS method, which selects the most active ligands by using docking of a target with known 3D structure and each ligand in a library of ligands. The docking method is divided into two steps: First, computational models of interaction test all possibilities of ligand-target binding, which is an exhaustive search procedure; In the second step, it is performed a biding affinity test between the ligand and the target using one or more mathematical algorithms, which are normally called *scoring functions*. However, docking scoring functions are known to be poor predictors of biding affinity, which consists in a weak point of docking. In recent years, some research have been done with the focus on improving scoring functions (rescoring), and the best reported results are currently based on machine learning approaches. In these approaches, in order to feed the machine learning algorithm with information that is important for the task, researchers normally analyze the docking output and extract features by hand. Although this process can be effective at some extent, handcrafting features is an arduous process, it does not scale and the researcher can always miss information whose importance is not easy to figure out, which can lead to a feature set that do not explain all the complexity of the problem. In this work, we propose a Deep Learning based method to perform rescoring. Deep Learning algorithms are known to be good at automatic feature extraction and are currently the state-of-the-art for computer vision, image processing, speech recognition and many natural language processing tasks. The main idea in our approach is to give the complete output of the docking step to a Deep Neural Network (DNN) and let it learn the features that are important for creating effective scoring functions. In order to validate our approach, we use the DUD (Directory of Useful Decoys) dataset to train and test a DNN for the task of biding affinity scoring. This dataset contains 40 targets, 2950 known ligands and 95316 decoys. **Results**: In our experiments, we use the DOCK 6.6 software to perform the VS and divide the dataset into 80% for training and 20% for test. In order to assess the DNN performance we use well established metrics such as enrichment factor (EF) and the area under the ROC curve. We hope to produce a system that achieves state-of-the-art results without using handcrafted features. **Conclusion**: Since DNNs are able to cope with tasks that involve complex feature interactions, we believe it is a suitable approach to improve docking scoring functions in virtual screening.

Supported by: Fiocruz, Capes.

A.2 Resumo publicado no VII Fórum Discente Fiocruz, 2015

Uma estratégia baseada em *Deep Learning* para melhoramento de *Virtual Screening* baseado em estrutura.

J. C. Pereira[1], C. N. dos Santos[2], E. R. Caffarena[1]

[1]Grupo de Biofísica Computacional e Modelagem Molecular, Programa de Computação Científica/Fiocruz.
[2] IBM Research.
janaina.pereira@ioc.fiocruz.br

Introdução: *Virutal screening* (VS) baseado em estrutura consiste na seleção de ligantes mais ativos usando a estratégia de *docking*. Um dos maiores entraves na metodologia de *docking* são as funções de pontuação. Essas funções não são capazes, por si só, de classificar de forma confiável ligantes docados. Nesse trabalho propomos um novo método baseado em *deep learning* para melhoramento de funções de pontuação em VS. **Metodologia e Resultados:** Nossa abordagem usa a saída do *docking* para treinar uma rede neural profunda (*deep neural network)* do tipo convolucional, nomeada como DeepVS, para identificar de forma automática características (*features*) que são boas preditoras da afinidade de ligação. Para verificarmos a eficácia da DeepVS, executamos experimentos de *docking* com o programa DOCK 6.6 sobre o subconjunto de dados do DUD com cargas parciais atômicas corrigidas, definido por 10 receptores, 431 ligantes anotados e 17.571 *decoys*. O desempenho da DeepVS é avaliado usando a abordagem de validação cruzada *leave-one-out* em conjunto com as métricas fator de enriquecimento e área sob a curva ROC (AUC). Em todos os casos a DeepVS melhorou expressivamente o resultado obtido pelo DOCK6.6 (*Grid Score*), tendo uma AUC média de 0,78 enquanto que a AUC média do DOCK6.6 é de 0,54. **Discussão e Conclusões:** Para verificar como a DeepVS se compara a outros métodos estado-da-arte em reclassificação de ligantes, comparamos nossos resultados com os resultados da rede neural DDFA-ALL. Em 60% dos casos a DeepVS supera o resultado da DDFA-ALL e em 20% dos casos a DeepVS obteve resultado igual ao obtido pela DDFA-ALL. Em média, a DeepVS (AUC 0,78) possui um desempenho melhor para todos os sistemas (proteínas) se comparado a DDFA-ALL (AUC 0,74). Os resultados da DeepVS também foram comparados com os resultados do estado-da-arte em redes neurais de *rescoring* de função de pontuação NNScore1 e NNScore2. A pesar dos autores do NNScore utilizarem em seus experimentos um conjunto diferente e mais simples de *decoys*, a DeepVS obteve resultados superiores (AUC média 0,78) à NNScore1 (AUC média 0,74) e à NNScore2 (AUC média 0,73), mesmo ranqueando um conjunto mais complexo de moléculas. Nossos experimentos preliminares demonstram que a DeepVS é um método eficaz para realização de VS, produzindo ranques de ligantes e *decoys* de maior qualidade do que os do programa DOCK6.6 e de outros métodos reportados na literatura.

A.3 Artigo publicado na revista Journal of Chemical Information and Modeling

JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

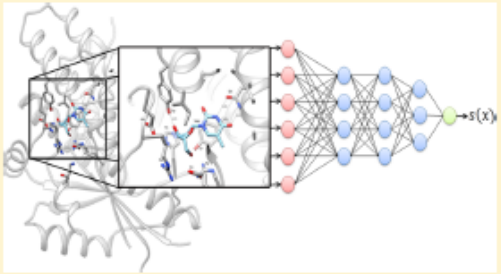
Boosting Docking-Based Virtual Screening with Deep Learning

Janaina Cruz Pereira,[*,†] Ernesto Raúl Caffarena,[*,†] and Cicero Nogueira dos Santos[*,‡]

[†]Fiocruz, 4365 Avenida Brasil, Rio de Janeiro, RJ 21040 900, Brazil

[‡]IBM Watson, 1101 Kitchawan Rd., Yorktown Heights, New York 10598, United States

ABSTRACT: In this work, we propose a deep learning approach to improve docking-based virtual screening. The deep neural network that is introduced, DeepVS, uses the output of a docking program and learns how to extract relevant features from basic data such as atom and residues types obtained from protein−ligand complexes. Our approach introduces the use of atom and amino acid embeddings and implements an effective way of creating distributed vector representations of protein−ligand complexes by modeling the compound as a set of atom contexts that is further processed by a convolutional layer. One of the main advantages of the proposed method is that it does not require feature engineering. We evaluate DeepVS on the Directory of Useful Decoys (DUD), using the output of two docking programs: Autodock Vina1.1.2 and Dock 6.6. Using a strict evaluation with leave-one-out cross-validation, DeepVS outperforms the docking programs, with regard to both AUC ROC and enrichment factor. Moreover, using the output of Autodock Vina1.1.2, DeepVS achieves an AUC ROC of 0.81, which, to the best of our knowledge, is the best AUC reported so far for virtual screening using the 40 receptors from the DUD.



INTRODUCTION

Drug discovery process is a time-consuming and expensive task. The development and even the repositioning of already-known compounds is a difficult chore.[1] The scenario gets worse if we take into account the thousands or millions of molecules capable of being synthesized in each development stage.[2,3]

In the past, experimental methods such as high-throughput screening (HTS) could help making this decision through the screening of large chemical libraries against a biological target. However, the high cost of the entire process associated with a low success rate makes this method inaccessible to academia.[2−4]

In order to overcome these difficulties, the use of low-cost computational alternatives is extensively encouraged, and it was adopted routinely as a way to aid in the development of new drugs.[3−5]

Computational virtual screening works basically as a filter (or a prefilter) consisting of the virtual selection of molecules, based on a particular predefined criterion of potentially active compounds against a determined pharmacological target.[2,4,6]

Two variants of this method can be adopted: ligand-based virtual screening and structure-based virtual screening. The first one involves the similarity and the physicochemical analysis of active ligands to predict the activity of other compounds with similar characteristics. The second is utilized when the three-dimensional (3D) structure of the target receptor was already elucidated somehow (experimentally or computationally modeled). This approach is used to explore molecular interactions between possible active ligands and residues of the binding site. Structure-based methods present a better performance, when compared to methods based solely on the structure of the ligand focusing on the identification of new compounds with therapeutic potential.[1,7−9]

One of the computational methodologies extensively used to investigate these interactions is molecular docking.[5,8,10] The selection of more-potent ligands using Docking-based Virtual Screening (DBVS) is made by performing the insertion of each compound from a compound library into a particular region of a target receptor with an elucidated 3D structure. In the first stage of this process, a heuristic search is carried out in which thousands of possible insertions are regarded. In the second, the quality of the insertion is described via some mathematical functions (scoring functions) that give a clue about the energy complementarity between the compound and the target.[10,11] The last phase became a challenge to the computational scientists, considering that it is easier to recover the proper binding mode of a compound within an active site than to assess a low energy score to a determined pose. This hurdle constitutes a central problem to the docking methodology.[12]

Systems based on machine learning (ML) have been successfully used to improve the outcome of DBVS for both, increasing the performance of score functions and constructing binding affinity classifiers.[1,3] The main strategies used in virtual screening are neural networks (NN),[13] support vector machines (SVM),[12] and random forest (RF).[14] One of the main advantages of employing ML is the capacity to explain the nonlinear dependence of the molecular interactions between the ligand and the receptor.[3]

Received: June 15, 2016
Published: November 4, 2016

DOI: 10.1021/acs.jcim.6b00355
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

APÊNDICE B

B.1 Arquivo para construção da grid Dock6.6

compute_grids	yes
grid_spacing	0.3
output_molecule	no
contact_score	no
energy_score	yes
energy_cutoff_distance	9999
atom_model	a
attractive_exponent	6
repulsive_exponent	12
distance_dielectric	yes
dielectric_factor	4
bump_filter	yes
bump_overlap	0.75
receptor_file	rec.mol2
box_file	rec_box.pdb
vdw_definition_file	vdw_AMBER_parm99.defn
score_grid_prefix	grid

B.2 Arquivo de configuração do *Virtual Screening* Dock6.6

ligand_atom_file	all_ligands_Gasteiger.mol2
limit_max_ligands	no
skip_molecule	no
read_mol_solvation	no
calculate_rmsd	yes
use_rmsd_reference_mol	no
use_database_filter	no
orient_ligand	yes
automated_matching	yes
receptor_site_file	selected_spheres.sph
max_orientations	500
critical_points	no
chemical_matching	no
use_ligand_spheres	no
use_internal_energy	yes
internal_energy_rep_exp	12
flexible_ligand	yes
user_specified_anchor	no
limit_max_anchors	no
min_anchor_size	40
pruning_use_clustering	yes
pruning_max_orients	100
pruning_clustering_cutoff	100
pruning_conformer_score_cutoff	25.0
use_clash_overlap	no
write_growth_tree	no
bump_filter	no
score_molecules	yes

contact_score_primary	no
contact_score_secondary	no
grid_score_primary	yes
grid_score_secondary	no
grid_score_rep_rad_scale	1
grid_score_vdw_scale	1
grid_score_es_scale	1
grid_score_grid_prefix	grid
multigrid_score_secondary	no
dock3.5_score_secondary	no
continuous_score_secondary	no
descriptor_score_secondary	no
gbsa_zou_score_secondary	no
gbsa_hawkins_score_secondary	no
SASA_descriptor_score_secondary	no
amber_score_secondary	no
minimize_ligand	yes
minimize_anchor	yes
minimize_flexible_growth	yes
use_advanced_simplex_parameters	no
simplex_max_cycles	1
simplex_score_converge	0.1
simplex_cycle_converge	1.0
simplex_trans_step	1.0
simplex_rot_step	0.1
simplex_tors_step	10.0
simplex_anchor_max_iterations	500
simplex_grow_max_iterations	500
simplex_grow_tors_premin_iterations	0
simplex_random_seed	0

simplex_restraint_min	no
atom_model	all
vdw_defn_file	vdw_AMBER_parm99.defn
flex_defn_file	flex.defn
flex_drive_file	flex_drive.tbl
ligand_outfile_prefix	virtual_flex
write_orientations	no
num_scored_conformers	1
rank_ligands	no

APÊNDICE C

C.1 Arquivo de configuração do *Virtual Screening* Autodockvina1.2

receptor = rec.pdbqt

center_x = 5.1780

center_y = 17.2920

center_z = -13.4420

size_x = 27

size_y = 27

size_z = 27

energy_range = 10

num_modes = 1

cpu = 5

exhaustiveness = 16

seed = -16807

APÊNDICE D

Tabela D.1: Informações adicionais dos ligantes conhecidos para a Cruzaína.

Ligante[a]	**Bibliografia**	**Mw**	**CLogP**	**IC_{50}(μM)**	**Ki(μM)**	**% Inibição Cruzaína(100μM)**
ZINC1026484	Du *et al.* (2000)	439	5,55	2,9	ND[a]	ND[a]
ZINC1033017	Du *et al.* (2000)	470	5,62	<10	ND	ND
ZINC1038200	Du *et al.* (2000)	434	5,75	3,1	ND	ND
ZINC1040170	Du *et al.* (2000)	387	3,62	3,7	ND	ND
ZINC1042930	Du *et al.* (2000)	398	6,14	1,2	ND	ND
ZINC2161654	Du *et al.* (2000)	436	7,08	<10	ND	ND
ZINC1043567	Du *et al.* (2000)	405	6,43	<10	ND	ND
ZINC1047389	Du *et al.* (2000)	446	5,66	<10	ND	ND
ZINC2148801	Du *et al.* (2000)	466	6,49	<10	ND	ND
ZINC2161657	Du *et al.* (2000)	437	6,61	<10	ND	ND
ZINC3106209	Du *et al.* (2000)	363	3,1	2,7	ND	ND
ZINC5223994	Du *et al.* (2000)	382	4,14	4,8	ND	ND
ZINC1035011	Du *et al.* (2000)	446	5,06	10	ND	ND
ZINC1035011	Rogers *et al.* (2012)	ND	ND	16,0	ND	ND
ZINC1035011	Rogers *et al.* (2012)	ND	ND	66,0	ND	ND
ZINC1035011	Rogers *et al.* (2012)	ND	ND	63,0	ND	ND
ZINC00943080	Ferreira *et al.* (2010)	ND	ND	0,4	2,0	ND
ZINC01852276	Ferreira *et al.* (2010)	ND	ND	0,7	ND	ND
ZINC02652325	Ferreira *et al.* (2010)	ND	ND	3,0	ND	ND
ZINC03242874	Ferreira *et al.* (2010)	ND	ND	7,0	0,07	ND
ZINC03363866	Ferreira *et al.* (2010)	ND	ND	7,0	6,0	ND
ZINC05061372	Ferreira *et al.* (2010)	ND	ND	18,0	ND	ND
ZINC05212600	Ferreira *et al.* (2010)	ND	ND	0,4	2,0	ND
ZINC09580294	Ferreira *et al.* (2010)	ND	ND	13,0	ND	ND
ZINC08693977	Ferreira *et al.* (2010)	ND	ND	13,0	ND	ND
ZINC00002334	Ferreira *et al.* (2014)	ND	ND	0,21	ND	92,0
ZINC02592566	Ferreira *et al.* (2014)	ND	ND	3,0	ND	92,0
ZINC08778859	Ferreira *et al.* (2014)	ND	ND	2,7	ND	96,0
ZINC71749919	Ferreira *et al.* (2014)	ND	ND	1,6	ND	96,0
ZINC02933983	Ferreira *et al.* (2014)	ND	ND	13,2	ND	93,0
ZINC32556196	Ferreira *et al.* (2014)	ND	ND	38,4	ND	66,0
ZINC00130100	Ferreira *et al.* (2014)	ND	ND	ND	ND	25,0
ZINC00124029	Ferreira *et al.* (2014)	ND	ND	ND	ND	25,0
ZINC12538929	Ferreira *et al.* (2014)	ND	ND	5,2	ND	84,0
ZINC12628932	Ferreira *et al.* (2014)	ND	ND	3,0	ND	90,0
ZINC22313068	Ferreira *et al.* (2014)	ND	ND	5,3	ND	92,0
ZINC169307576	Ferreira *et al.* (2014)	ND	ND	0,21	ND	92,0
ZINC169307573	Ferreira *et al.* (2014)	ND	ND	0,6	ND	90,0

[a] Não identificado em bibliografia.