JÉSSICA COSTA DE OLIVEIRA

UMA ESTRATÉGIA COMPUTACIONAL BASEADA EM APRENDIZAGEM SUPERVISIONADA PARA PREDIÇÃO MOLÉCULAS PARA USO AGRÍCOLA

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de Magister Scientiae.

Orientador: Sabrina de Azevedo Silveira.

VIÇOSA - MINAS GERAIS
2023

Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus

T
O48e
2023

Oliveira, Jéssica Costa de, 1994-
Uma estratégia computacional baseada em aprendizagem supervisionada para predição moléculas para uso agrícola: / Jéssica Costa de Oliveira. - Viçosa, MG, 2023.
1 dissertação eletrônica (69 f.): il. (algumas color.).
Orientador: Sabrina de Azevedo Silveira
Dissertação (mestrado) - Universidade Federal de Viçosa, Departamento de Informática, 2023.
Referências bibliográficas: .
DOI: https://doi.org/10.47328/ufvbbt.2023.681
Modo de acesso: World Wide Web.

1. Aprendizado do computador; 2. Proteínas - Estrutura; 3. Bioinformática; I. Silveira, Sabrina de Azevedo II. Universidade Federal de Viçosa.. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação III. Título

CDD 23. ed. 006.31

Bibliotecário(a) responsável: EUZEBIO LUIZ PINTO CRB-6/3317

JÉSSICA COSTA DE OLIVEIRA

UMA ESTRATÉGIA COMPUTACIONAL BASEADA EM APRENDIZAGEM SUPERVISIONADA PARA PREDIÇÃO MOLÉCULAS PARA USO AGRÍCOLA

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 30 de junho de 2023.

Assentimento:

Jéssica Costa de Oliveira
Autora

Sabrina de Azevedo Silveira
Orientadora

Aos meus pais, irmão e amigos.

Agradecimentos

A Deus.

Aos meus pais.

À Universidade Federal de Viçosa, pela oportunidade de realizar a pós-graduação.

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (Capes).

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes), pela concessão da bolsa de estudos.

“Confia ao Senhor as tuas obras, e teus pensamentos serão estabelecidos.”
(Provérbios 16-3)

Resumo

OLIVEIRA, Jéssica Costa de, M.Sc., Universidade Federal de Viçosa, junho de 2023. **Uma Estratégia Computacional Baseada em Aprendizagem Supervisionada para Predição Moléculas para Uso Agrícola**. Orientadora: Sabrina de Azevedo Silveira.

O aumento da utilização de recursos computacionais em pesquisas científicas tem contribuído para uma maior aplicação dessas tecnologias nos trabalhos científicos na área da Bioinformática. Uma técnica computacional que tem sido bastante utilizada é a *virtual screening* ou triagem virtual de ligantes. Como resultado dessas contribuições, pode-se citar as descobertas de novos fármacos, as mutações em resíduos de proteínas, o alinhamento de sequências, entre outros. Além dos benefícios mencionados acima, existe a perspectiva de descobertas de medicamentos e vacinas com menores custos e com menor tempo de desenvolvimento desses fármacos. Nesse sentido, pode-se citar a descoberta, em caráter emergencial, da vacina contra a *Covid-19*, doença causada pelo *Sars-Cov-2*, nome oficial do novo coronavírus. O reposicionamento de fármacos é outra técnica utilizada. Por meio dessa, busca-se avaliar a eficácia de medicamentos já existentes para determinadas enfermidades em outros tipos de doenças. Esse recurso assemelha-se ao teste utilizado para desenvolver a vacina *Sars-Cov-2*. Neste trabalho, propôs-se a utilização de uma estratégia computacional em aprendizado supervisionado para caracterizar e prever ligantes que podem interagir com moléculas importante no contexto da agricultura. O cenário de aplicação é a soja e seu inseto praga, a lagarta *Anticarsia gemmatalis Hubner*. Assim, busca-se prever potenciais moléculas que possam inibir proteínas no intestino da lagarta e, consequentemente, o controle de pragas.

Palavras-chave: Triagem Virtual. Aprendizado de Máquina. Proteína-Ligante. Protease.

Abstract

OLIVEIRA, Jéssica Costa de, M.Sc., Universidade Federal de Viçosa, June 2023. **A Computational Strategy Based on Supervised Learning for Predicting Molecules for Agricultural Use**. Adviser: Sabrina de Azevedo Silveira.

The increased used of computational resources in scientific research has contributed to a greater application of these technologies in scientific work in the field of Bioinformatics. A computational technique that has been widely used is virtual screening of ligands. As a result of these contributions, we can mention the discoveries of new drugs, mutations in protein residues, sequence alignment, among others. In addition to the benefits mentioned above, there is the prospect of discovering drugs and vaccines at lower costs and with a shorter development time for these drugs. In this sense, one can mention the discovery, on an emergency basis, of the vaccine against Covid-19, a disease caused by Sars-Cov-2, the official name of the new coronavirus. Drug repositioning is another technique used. Through this, we seek to evaluate the effectiveness of existing drugs for certain diseases in other types of diseases. This feature is similar to the test used to develop the Sars-Cov-2 vaccine. In this work, we propose the use of a computational strategy in supervised learning to characterize and predict ligands that can interact with important molecules in the context of agriculture. The application scenario is soybean and its insect pest, the caterpillar Anticarsia gemmatalis Hubner. Thus, we seek to predict potential molecules that can inhibit proteins in the caterpillar's intestine and, consequently, pest control.

Keywords: Virtual Screening. Machine Learning. Protein-Ligand. Protease.

Lista de ilustrações

Figura 1 – Representação de algumas das principais áreas da Bioinformática. 16
Figura 2 – Formação estrutural do aminoácido Aspagarina. 18
Figura 3 – Estrutura geral de um aminoácido. O grupo *R*, ou cadeia lateral (roxo), ligado ao carbono (cinza) é diferente em cada aminoácido. 19
Figura 4 – Os 20 aminoácidos comuns de proteínas e algumas de suas propriedades. 20
Figura 5 – Níveis de estrutura nas proteínas. 21
Figura 6 – Formação de uma ligação peptídica entre dois aminoácidos, com liberação de uma molécula de água (condensação). 22
Figura 7 – Diferentes representações da estrutura secundária em α-hélice de uma proteína. 23
Figura 8 – Representação da estrutura terciária de uma proteína. 23
Figura 9 – Grupo *Heme* presente na Hemoglobina. 24
Figura 10 – Respectivas áreas de Aprendizado Máquina. 25
Figura 11 – Modelo ilustrativo do *Perceptron* para reconhecimento de padrões. . 26
Figura 12 – *Perceptron*. 27
Figura 13 – Rede com múltiplas camadas. 28
Figura 14 – Exemplo de Regressão Logística. 29
Figura 15 – Organograma da condução dos experimentos. 31
Figura 16 – Características da família *serine protease*. 32
Figura 17 – Exemplo de *serina protease*. 33
Figura 18 – *Trypsin-like serine protease* no *BLAST*. 33
Figura 19 – Alinhamento das sequências de proteínas no *BLAST*. 34
Figura 20 – Busca de dataset no *BindingDB*. 34
Figura 21 – Construção da *query* a partir da sequência. 35
Figura 22 – Resultado da *query* a partir da sequência da proteína. 36
Figura 23 – Tratamento da sequência de proteína, com a remoção de água e de íons no *Pymol*. 36
Figura 24 – Sequência de proteína com água e com íons removidos no *Pymol*. . 37
Figura 25 – Repositório de geração de *Decoys*. 37
Figura 26 – Molécula gerada pelo *software RDkit*. 38
Figura 27 – Parâmetros do algoritmo *LogisticRegression*. 39
Figura 28 – Parâmetros dos algoritmos *MLPClassifier*. 39
Figura 29 – Emprego do método de atracamento molecular na predição do modo de ligação do *GTP* ao seu sítio de ligação na proteína *c-H-ras p21*. 41
Figura 30 – Graus de flexibilidade do receptor. 41

Figura 31 – Proteína correspondente à família da protease *Subtilisin-like serine protease (Plasmodium falciparum)*. 44
Figura 32 – Ligante da proteína correspondente à família da protease *Subtilisin-like serine protease (Plasmodium falciparum)*. 45
Figura 33 – *Matrix Generator*. 45
Figura 34 – *Matrix Generator* gerada a partir do classificador *LogisticRegression*. 46
Figura 35 – *Matrix Generator* gerada a partir do classificador *MLPClassifier*. . . 47
Figura 36 – *Matrix Generator* gerada a partir do classificador *MLPClassifier*. . . 48
Figura 37 – *Matrix Generator* gerada a partir do classificador *MLPClassifier*. . . 49
Figura 38 – Exemplo de conversão de ligante em formato *.sdf* para *.pdbqt* no *Open Babel*. 52
Figura 39 – Visão geral da proteína no *Discovery Studio*. 52
Figura 40 – Região do ligante (em amarelo) na proteína. 53
Figura 41 – As coordenadas *XYZ* do ligante para formação da caixa. 53
Figura 42 – Estrutura do ligante da molécula no *Discovery Studio*. 54
Figura 43 – Estrutura da proteína carregada no *Autodock*. 54
Figura 44 – Preparação da caixa de simulação no *Autodock*. 55
Figura 45 – Interações obtidas no processo de *docking* da proteína: *4x2u*. . . . 56
Figura 46 – Interações obtidas no processo de *docking* da proteína: *4zw5*. . . . 57
Figura 47 – Interações obtidas no processo de *docking* da proteína: *4zw6*. . . . 57
Figura 48 – Interações obtidas no processo de *docking* da proteína: *4zw7*. . . . 58
Figura 49 – Interações obtidas no processo de *docking* da proteína: *4zw8*. . . . 58
Figura 50 – Interações obtidas no processo de *docking* da proteína: *4zx4*. . . . 59
Figura 51 – Interações obtidas no processo de *docking* da proteína: *4zx5*. . . . 59
Figura 52 – Interações obtidas no processo de *docking* da proteína: *4zx6*. . . . 60
Figura 53 – Interações obtidas no processo de *docking* da proteína: *5y1s*. . . . 60
Figura 54 – Interações obtidas no processo de *docking* da proteína: *5y3i*. 61
Figura 55 – Interações obtidas no processo de *docking* da proteína: *6ea1*. . . . 61
Figura 56 – Interações obtidas no processo de *docking* da proteína: *6ea2*. . . . 62
Figura 57 – Interações obtidas no processo de *docking* da proteína: *6eaa*. . . . 62
Figura 58 – Interações obtidas no processo de *docking* da proteína: *6eab*. . . . 63
Figura 59 – Interações obtidas no processo de *docking* da proteína: *6ee3*. . . . 63
Figura 60 – Interações obtidas no processo de *docking* da proteína: *6ee4*. . . . 64
Figura 61 – Interações obtidas no processo de *docking* da proteína: *6ee6*. . . . 64
Figura 62 – Interações obtidas no processo de *docking* da proteína: 6eed. . . . 65

Lista de tabelas

Tabela 1 – Exemplos de proteínas e suas funções. 21
Tabela 2 – Aspectos dos parâmetros característicos do *Perceptron*. 27
Tabela 3 – Matriz de Confusão para um problema de classificação binária. . . 40
Tabela 4 – Portais de acesso para alguns programas de *Docking Molecular*. . . 43
Tabela 5 – Resultados do classificador *LogisticRegression*. 47
Tabela 6 – Resultados do classificador *MLPClassifier*. 48
Tabela 7 – Resultados do classificador *MLPClassifier*. 49
Tabela 8 – Resultados do classificador *MLPClassifier*. 50
Tabela 9 – Métricas dos Classificadores *LogisticRegression* e *MLPClassifier*. . 50
Tabela 10 – Moléculas utilizadas no *Docking Molecular*. 51
Tabela 11 – Nomenclaturas dos aminoácidos. 56

Lista de abreviaturas e siglas

PDB	*Protein Data Bank*
BLAST	*Basic Local Alignment Search Tool*
VS	*Virtual Screening*

Lista de símbolos

α	Alfa
β	Beta
θ	Theta
π	PI
∞	Infinito

Sumário

1 INTRODUÇÃO 16
1.1 Objetivos 17
1.2 Organização do Trabalho 17
2 REVISÃO DE LITERATURA 18
2.1 Fundamentos Teóricos 18
2.1.1 Proteínas 18
2.1.2 Estrutura Primária 21
2.1.3 Estrutura Secundária 22
2.1.4 Estrutura Terciária 23
2.1.5 Estrutura Quaternária 24
2.2 Aprendizado de Máquina 25
2.2.1 Aprendizagem Supervisionada 26
2.2.1.1 *Multilayer Perceptron (MLP)* 26
2.2.1.2 Regressão Logística 28
2.3 Trabalhos Relacionados 30
3 MATERIAIS E MÉTODOS 31
3.1 Fonte de Dados 32
3.2 Preparação e Análise das Proteínas 35
3.3 Algoritmos de Aprendizado de Máquina 38
3.3.1 Pacotes *RDkit* 38
3.3.2 Parâmetros dos Algoritmos de Aprendizagem Supervisionada . 38
3.3.3 Métricas 39
3.4 *Docking Molecular* 41
3.4.1 Interações proteína-ligante 42
4 RESULTADOS E DISCUSSÃO 44
4.1 Matriz de Confusão 45
4.2 Classificadores 46
4.2.1 *LogisticRegression* 46
4.2.2 *MLPClassifier* 47
4.2.3 Métricas dos Classificadores 50
4.3 *Docking Molecular* 51
4.3.1 *Open Babel* 51
4.3.2 *Discovery Studio* 52
4.3.3 *Autodock* 54
4.3.4 Resultados do *Docking* 55
4.3.4.1 Resultados das Interações proteína-ligante 65

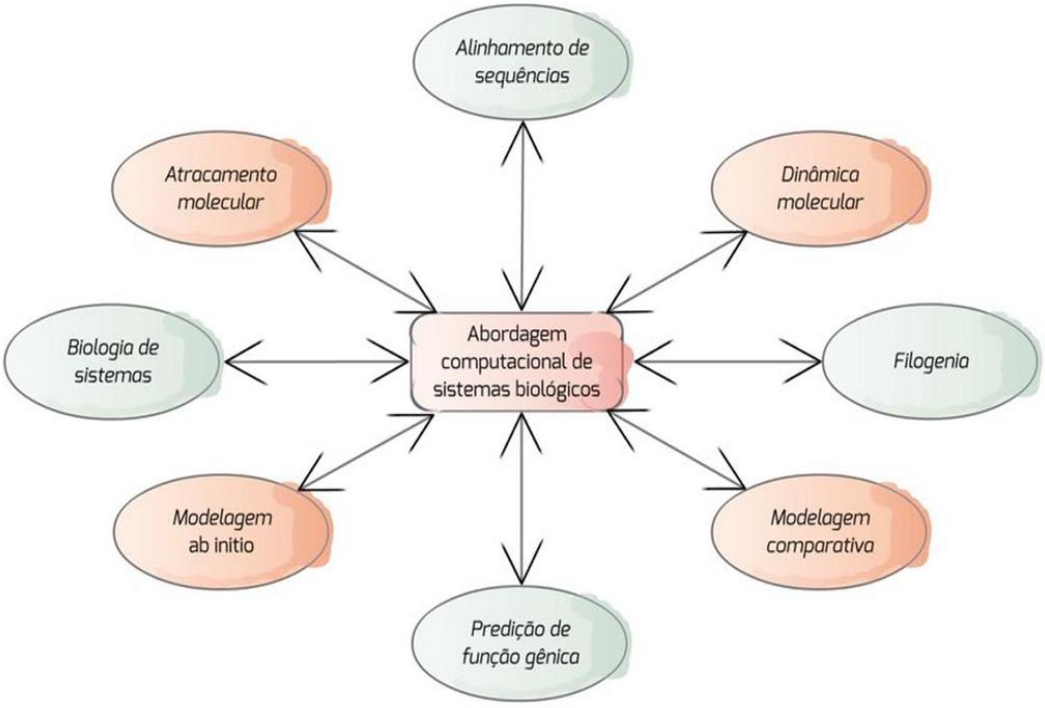
5 CONCLUSÃO **67**

Referências **68**

1 INTRODUÇÃO

A Bioinformática é um campo de estudo interdisciplinar utilizada para análises e interpretações significativas de dados biológicos, constituída por áreas da computação, matemática, estatística e biologia. Percebe-se que os avanços do poder computacional tem contribuído para o aumento das aplicações dessa ciência na pesquisa científica, tais como: sequenciamento de *DNA*, descobertas de novos fármacos, descobertas de genes, alinhamento de proteínas, entre outros (BAYAT, 2002). Na Figura 1, são apresentadas algumas das principais áreas da Bioinformática.

Figura 1 – Representação de algumas das principais áreas da Bioinformática.



Fonte: Figura extraída da referência (VERLI, 2014).

Um marco histórico considerado pelos pesquisadores como o nascimento da Bioinformática foi o sequenciamento do primeiro genoma, em 1995. Nesse momento, já existiam computadores capazes de lidar com o grande volume de material genético gerado pelo sequenciamento de genomas (DUCK et al., 2016; DRAGON et al., 2020).

O surgimento de ferramentas de cunho computacional fez a diversificação da pesquisa em Bioinformática se tornar realidade e ir para além do sequenciamento genômico. Verifica-se também que devido à robustez das simulações computacionais, a utilização de técnicas de Aprendizado de Máquina tornou-se cada vez mais constante (DUCK et al., 2016; DRAGON et al., 2020).

Uma perspectiva que pesquisadores da Bioinformática têm é a possibilidade de realizar simulações computacionais de dados ômicos e organismos pluricelulares, para simulação inteiramente virtual de um ser vivo (BAYAT, 2002; GAUTHIER et al., 2018).

1.1 Objetivos

O objetivo deste trabalho é projetar, implementar e avaliar as estratégias computacionais capazes de caracterizar e prever moléculas que possam inibir proteínas (também chamados de alvos) de interesse.

Os objetivos específicos foram:

- Revisar na literatura trabalhos científicos relacionados a protease da lagarta;
- Construir um conjunto de dados de protease da lagarta ou similar para realização do treinamento dos modelos de Aprendizagem Supervisionada;
- Executar os algoritmos de Aprendizagem Supervisionada para o conjunto de dados;
- Avaliar a eficácia e a qualidade dos modelos de Aprendizagem Supervisionada através das métricas *Precision* e *Recall*;
- Realizar o *Docking* para identificar as interações são mais favoráveis entre os modelos de Aprendizagem Supervisionada.

1.2 Organização do Trabalho

Esta dissertação é constituída de 4 (quatro) capítulos. O primeiro capítulo apresenta uma *Introdução* geral ao tema proposto. No segundo, *Revisão de Literatura*, são apresentados os fundamentos teóricos para entendimento dos elementos utilizados para estratégia de pesquisa. No terceiro, *Materiais e Métodos*, serão apresentados as ferramentas e as técnicas de Aprendizagem Supervisionada e *Docking Molecular* utilizadas para o desenvolvimento deste trabalho. O Capítulo 4, *Resultados e Discussão*, são discutidos os resultados obtidos por meio dos experimentos e, por fim, no quinto capítulo, *Conclusão*, são feitas as considerações finais e apresentam-se sugestões para trabalhos futuros.

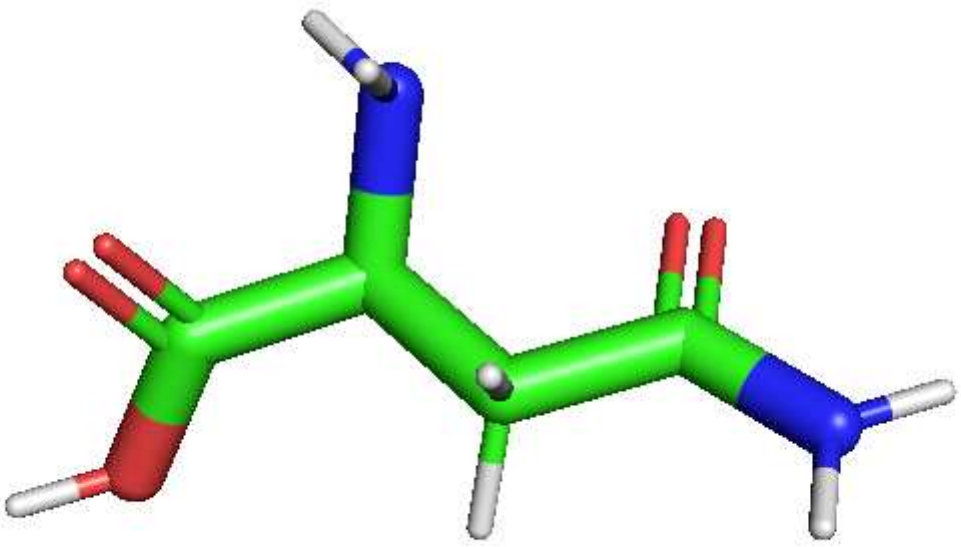
2 REVISÃO DE LITERATURA

2.1 Fundamentos Teóricos

2.1.1 Proteínas

Sabe-se que as proteínas são polímeros resultantes da desidatração de aminoácido, e cada resíduo de aminoácido liga-se ao seu vizinho através de uma ligação covalente (”resíduo”é a perda de água após a união de um aminoácido ao outro) (NELSON; COX, 2014). Destaca-se que o primeiro aminoácido descorberto nas proteínas foi a aspagarina, em 1806 (NELSON; COX, 2014). Na Figura 2, tem-se uma demonstração da estrutura da aspagarina.

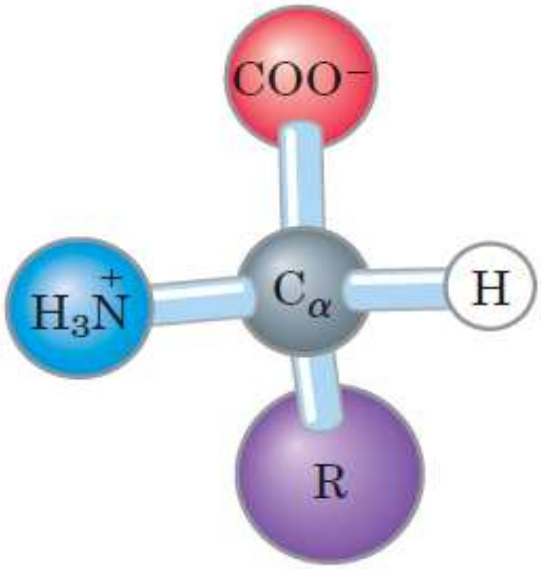
Figura 2 – Formação estrutural do aminoácido Aspagarina.



Fonte: Autor.

Salienta-se, também, os 20 (vintes) aminoácidos encontrados em proteínas contêm um grupo carboxila e um grupo amino ligado no mesmo átomo de carbono (NELSON; COX, 2014) e que esses grupos diferem-se entre si por meio do grupo *R* ou cadeias laterais (NELSON; COX, 2014). Com exceção da glicina, o carbono está ligado a quatro grupos diferentes: um grupo carboxila, um grupo amino, um grupo *R* e um átomo de hidrogênio, conforme apresentado na Figura 3 . Na Figura 4, representação dos 20 (vintes) aminoácidos e respectivas propriedades.

Figura 3 – Estrutura geral de um aminoácido. O grupo *R*, ou cadeia lateral (roxo), ligado ao carbono (cinza) é diferente em cada aminoácido.

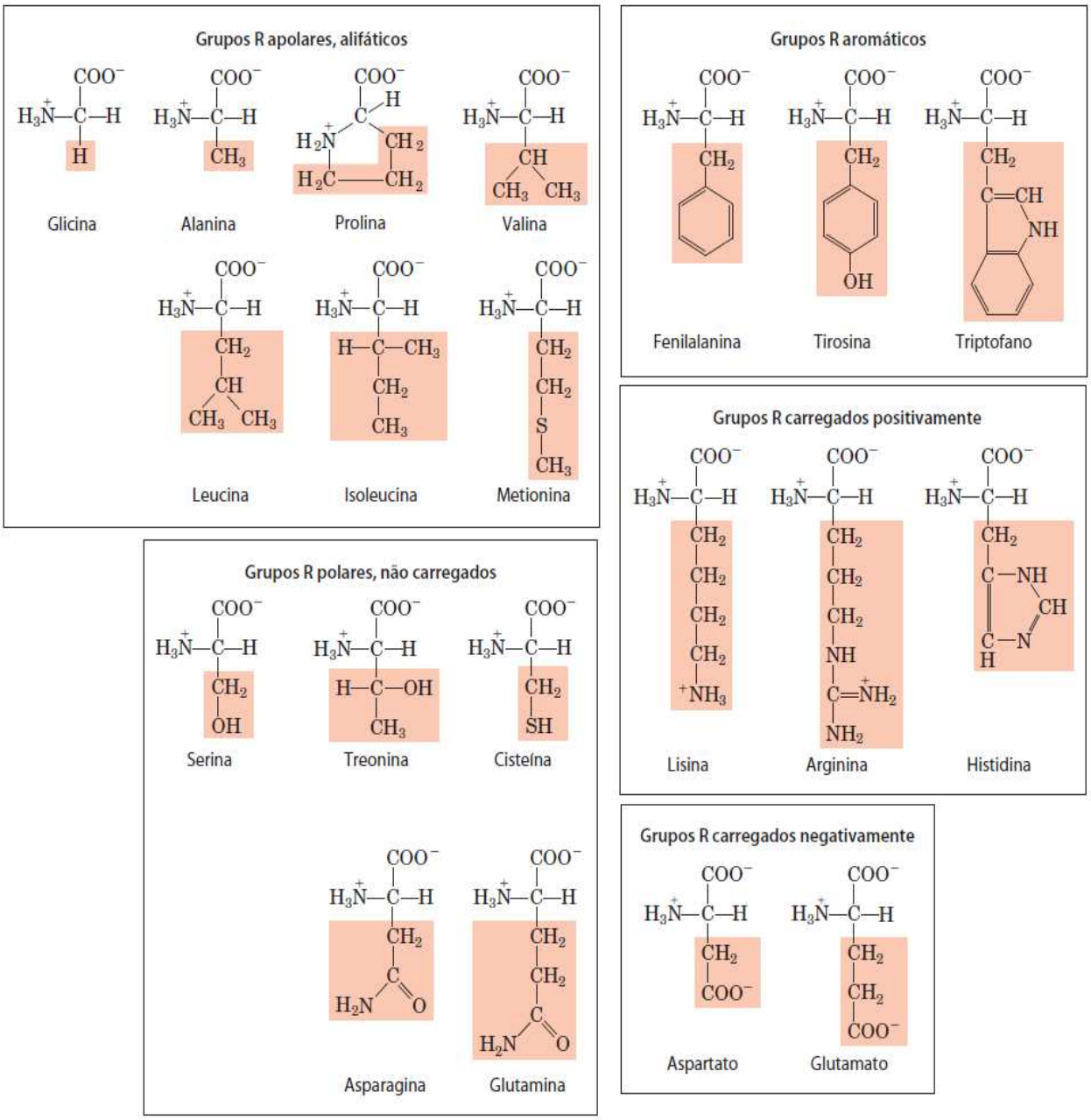


Fonte: Figura extraída da referência (NELSON; COX, 2014).

Os grupos *R* dos aminoácidos podem ser classificados da seguinte forma:

- **Grupo R apolares, alifáticos:** nesta classe, são apolares e hidrofóbicos. As cadeias laterais tendem a se agrupar no interior de proteínas, o que estabiliza a estrutura proteíca por meio de interações hidrofóbicas com exceção da **glicina** (NELSON; COX, 2014);
- **Grupo R aromáticos:** são apolares (hidrofóbicos). As cadeias laterais aromáticas e todos desta classe de aminoácido participam das interações hidrofóbicas (NELSON; COX, 2014);
- **Grupo R polares:** os grupo R desta classe são mais solúveis em água, ou mais hidrofílicos do que os apolares, pois contêm ligações de hidrogênio (NELSON; COX, 2014);
- **Grupo R carregados positivamente:** os grupos *R* desta classe de aminoácido tem uma carga positiva em pH 7.0. São: a lisina com um grupo amino na posição em cadeia alifática; a **arginina**, com um grupo guanidínio positivamente carregado; e a **histidina**, com um grupo imidazol aromático (NELSON; COX, 2014);
- **Grupo R carregados negativamente:** os aminoácidos que apresentam grupo *R* com carga negativa final em pH 7.0 são o aspartato e o glutamato (NELSON; COX, 2014).

Figura 4 – Os 20 aminoácidos comuns de proteínas e algumas de suas propriedades.



Fonte: Figura extraída da referência (NELSON; COX, 2014).

Segundo (NELSON; COX, 2002), as proteínas possuem funções importantes para o funcionamento do nosso corpo, dentre elas estão: transporte de oxigênio, contração muscular, estruturação e anabolismo muscular, proteção imunológica e outras funções (NELSON; COX, 2002). Na Tabela 1, uma representação de funções em nosso corpo.

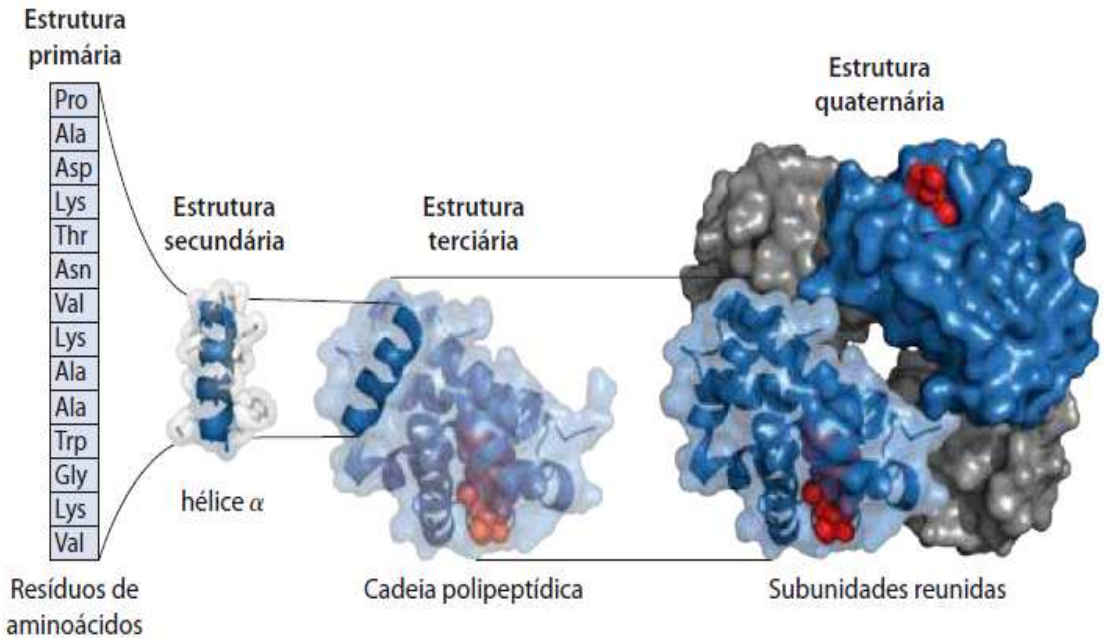
Tabela 1 – Exemplos de proteínas e suas funções.

Funções	Proteínas
Estruturação	Microtúbulos
Transporte	Albumina
Movimentação	Actiona e Miosina
Alimentação	Caseína
Defesa	Imunoglobinas
Coordenação	Insulina
Catalisador	Enzimas protéicas

Fonte: Autor.

Além apresentar diferentes funções, as proteínas dividem-se em 4 (quatro) estruturas, sendo elas: primária, secundária, terciária e quartenária, conforme na Figura 5.

Figura 5 – Níveis de estrutura nas proteínas.



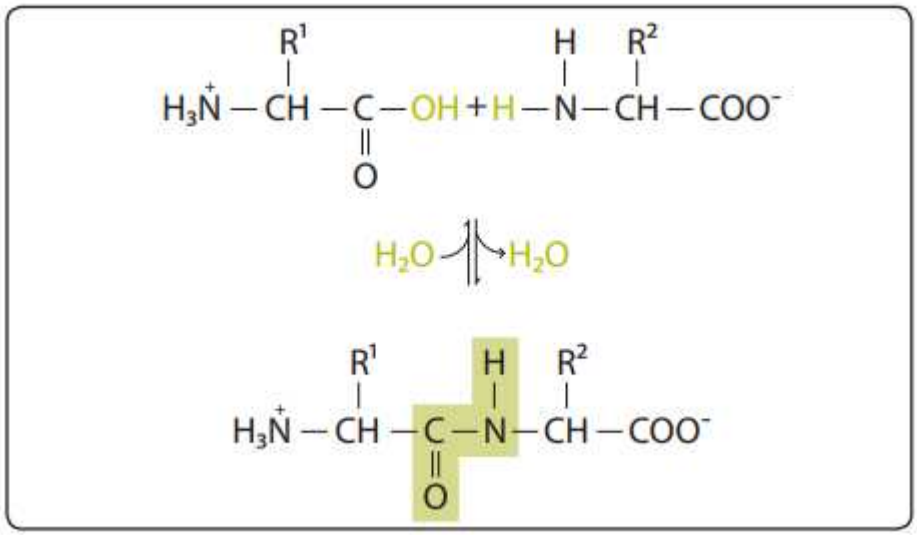
Fonte: Figura extraída da referência (NELSON; COX, 2014).

2.1.2 Estrutura Primária

A estrutura primária de uma proteína é, simplesmente, sua sequência de aminoácidos, também chamada de sequência primária (MARQUES, 2014; NELSON; COX, 2014).

Para entender uma estrutura primária, precisamos conhecer a ligação peptídica uma reação entre dois aminoácidos forma-se a ligação peptídica. Além disso, uma ligação peptídica, que é uma reação entre dois aminoácidos, e uma ligação covalente entre o grupo carboxila *(COO-)* e o grupo amino *(NH3+)* do aminoácido (MARQUES, 2014; NELSON; COX, 2014). Na Figura 6, uma representação de uma ligação peptídica.

Figura 6 – Formação de uma ligação peptídica entre dois aminoácidos, com liberação de uma molécula de água (condensação).



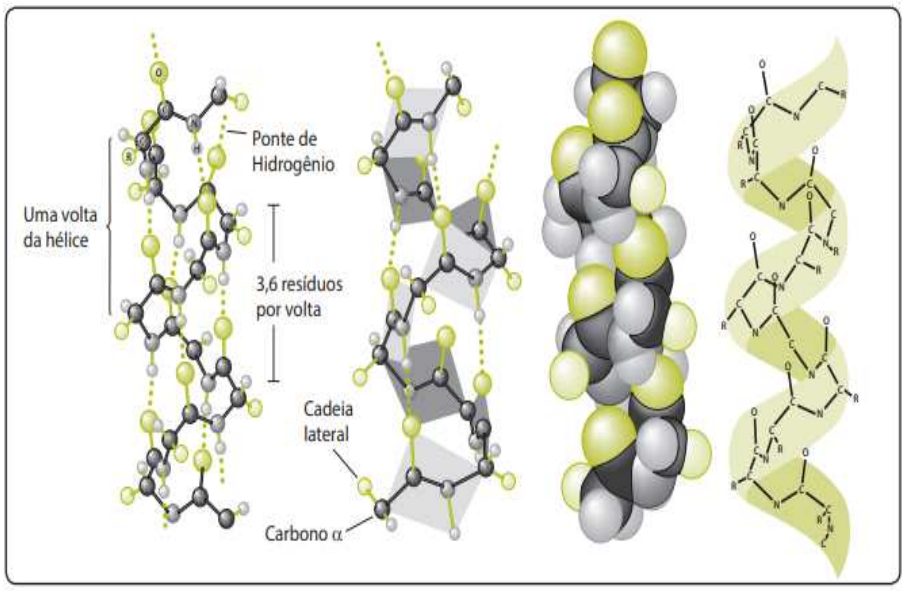
Fonte: Figura extraída da referência (MARQUES, 2014).

2.1.3 Estrutura Secundária

Considera-se estrutura secundária de uma proteína, a maneira como os aminoácidos se organizam no espaço (MARQUES, 2014). Os elementos mais comuns que compõem essa estrutura são:

- **hélices**: interações entre aminoácidos, as quais ocorrem em função da formação de uma espiral ou estrutura helicoidal (MARQUES, 2014). Na Figura 7, uma demonstração de diferentes representações da estrutura secundária em α-hélice.
- **folhas**: os esqueletos polipeptídicos da sequência primária dos aminoácidos de uma proteína organizados no espaço têm o formato “zigue-zague” (MARQUES, 2014).
- **laços**: conectam diferentes segmentos das proteínas, podendo mudar a direção da cadeia.

Figura 7 – Diferentes representações da estrutura secundária em α-hélice de uma proteína.



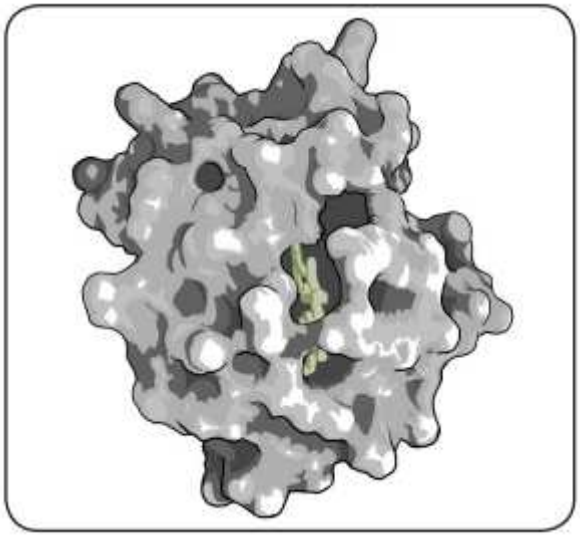
Fonte: Figura extraída da referência (MARQUES, 2014)

2.1.4 Estrutura Terciária

A estrutura terciária da proteína descreve os aspectos de enovelamento que leva à formação globular (MARQUES, 2014; NELSON; COX, 2014).

O enovelamento da cadeia polipeptídica permite que resíduos de aminoácidos apolares evitem o contato com a água, permanecendo posicionados na parte mais interna da estrutura globular, conforme na Figura 8 (MARQUES, 2014; NELSON; COX, 2014).

Figura 8 – Representação da estrutura terciária de uma proteína.



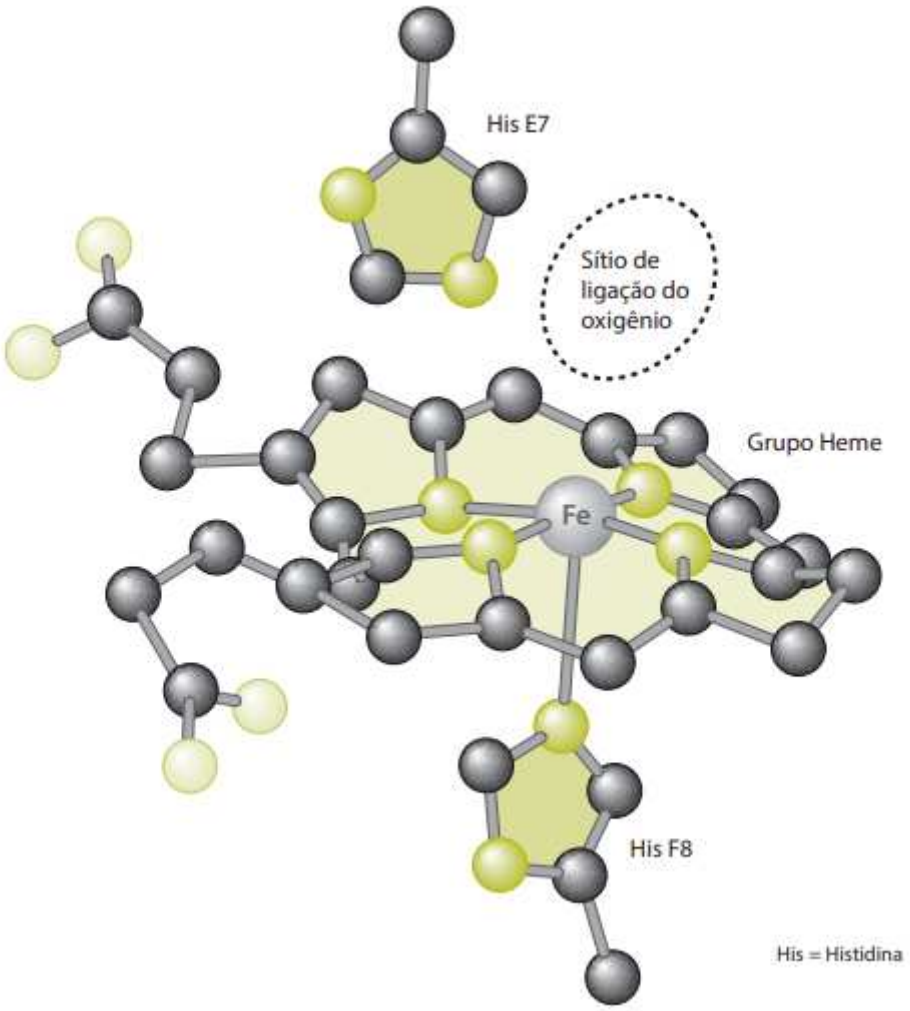
Fonte: Figura extraída da referência (MARQUES, 2014).

2.1.5 Estrutura Quaternária

Algumas proteínas, como a hemoglobina ou os anticorpos, apresentam um nível estrutural superior aos descritos até aqui. Esse nível de organização estrutural mais complexo, denominado estrutura quaternária, envolve a participação de pelo menos duas cadeias polipeptídicas na formação da proteína. Denomina-se de estrutura quaternária (MARQUES, 2014).

As proteínas formadas por duas ou mais cadeias polipeptídicas são denominadas de proteínas oligoméricas, e cada cadeia polipeptídica que forma a sua estrutura molecular (MARQUES, 2014). Um exemplo clássico de estrutura quaternária é a hemoglobina, conforme na Figura 9.

Figura 9 – Grupo *Heme* presente na Hemoglobina.



Fonte: Figura extraída da referência (MARQUES, 2014).

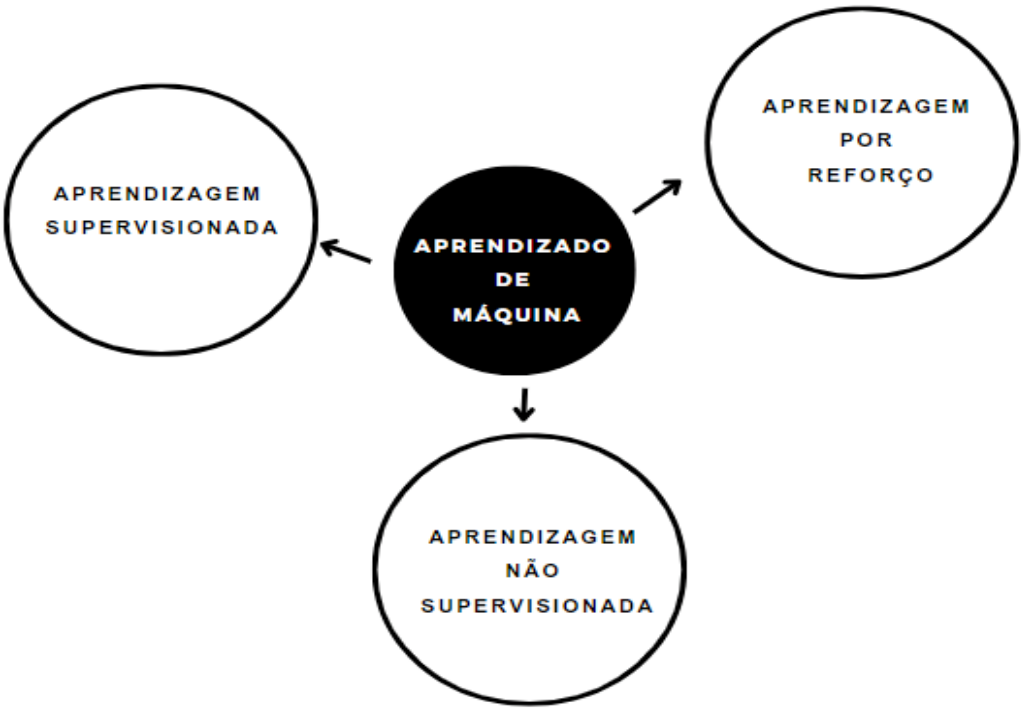
2.2 Aprendizado de Máquina

Nos últimos anos, tem-se vivenciado mudanças significativas impulsionadas pelo avanço da Inteligência Artificial *(IA)*. As máquinas, além de fazerem os trabalhos para os quais são programadas, também estão aprendendo, assim como os humanos, desempenhar novas tarefas. Surge, então, uma subárea da Inteligência Artificial – chamada Aprendizado de Máquina (KUBAT, 2017; MULLER; GUIDO, 2017).

Esse Aprendizado está relacionado à construção de modelos computacionais, ou seja, a criação de algoritmos capazes de possibilitar a máquina aprender, a partir de um grande volume de dados, desempenhar determinadas tarefas. Ressalta-se que já existem inúmeras funções com base nesse aprendizado, entre elas, tem-se: o reconhecimento de voz, a mineração de dados, o reconhecimento de padrões, além das encontradas na Bioinformática, objeto deste estudo (KUBAT, 2017; MULLER; GUIDO, 2017; GéRON, 2019).

Devido aos avanços tecnológicos, com os quais convivemos em nosso dia a dia, é inevitável não perceber a importância do Aprendizado de Máquina para pesquisa científica. Por meio de um grande volume de dados, o computador consegue gerar conhecimento, que são hipóteses construídas a partir dos dados (KUBAT, 2017; MULLER; GUIDO, 2017; GéRON, 2019). Destaca-se que existem três tipos principais de Apredizado de Máquina: Supervisionada; Não Supervisionada; e por Reforço. Na Figura 10, são apresentadas as referidas áreas.

Figura 10 – Respectivas áreas de Aprendizado Máquina.



Fonte: Autor.

Destaca-se que o foco deste trabalho é, especificamente, a Aprendizagem Supervisionada, ou seja, aquela em que os algoritmos de classificação que em conjunto com as métricas são capazes de avaliar a eficácia e a qualidade do modelo para, enfim, poder caracterizar e prever os ligantes no contexto da protease da lagarta.

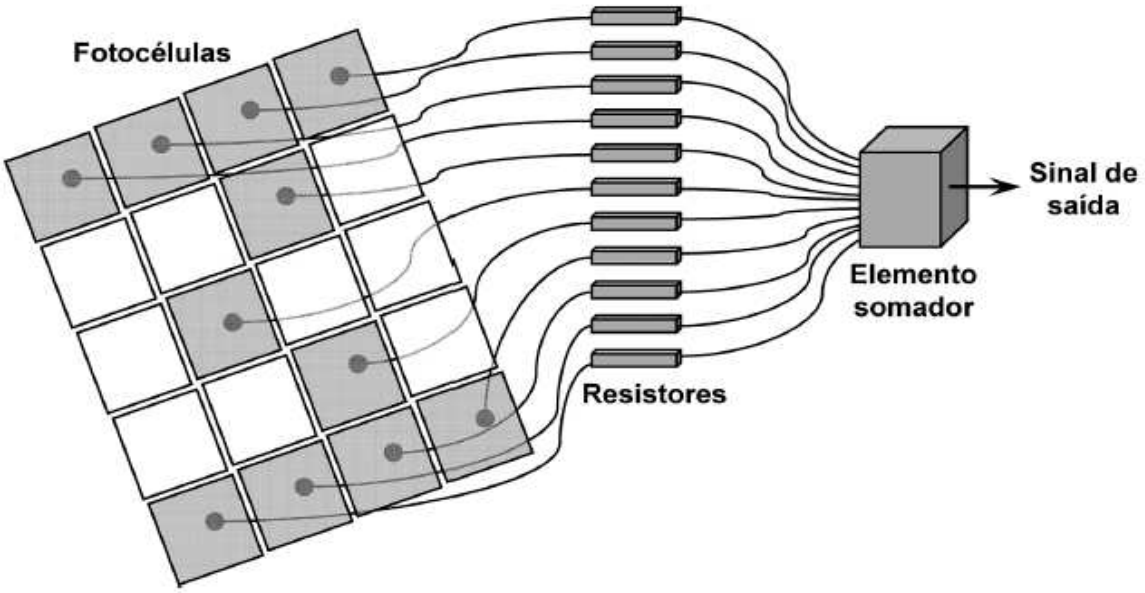
2.2.1 Aprendizagem Supervisionada

Nesta seção, serão abordados os fundamentos teóricos dos algoritmos de Aprendizagem Supervisionada, o *Multilayer Perceptron* e a Regressão Logística utilizados neste trabalho.

2.2.1.1 *Multilayer Perceptron (MLP)*

O *Multilayer Perceptron (MLP)* é um algoritmo de Aprendizagem Supervisionada baseado em camadas, ou seja, em uma camada de entrada, formada por neurônios, que se conecta às entradas; em uma ou mais camadas ocultas (intermediárias), cujas entradas e saídas se conectam a outros neurônios; e em uma camada de saída, que são as saídas dos neurônios. Cada entrada possui um peso, e cada saída é dada por uma função de ativação. Um *Perceptron* é a forma simples de configuração de uma rede neural artificial que, por meio da implementação de uma modelagem computacional, inspirado na retina (SILVA; SPATTI; FLAUZINO, 2010). A Figura 11 mostra uma concepção inicial do elemento *Perceptron*, em que os sinais elétricos advindos de fotocélulas mapeiam padrões geométricos por resistores sintonizáveis (SILVA; SPATTI; FLAUZINO, 2010).

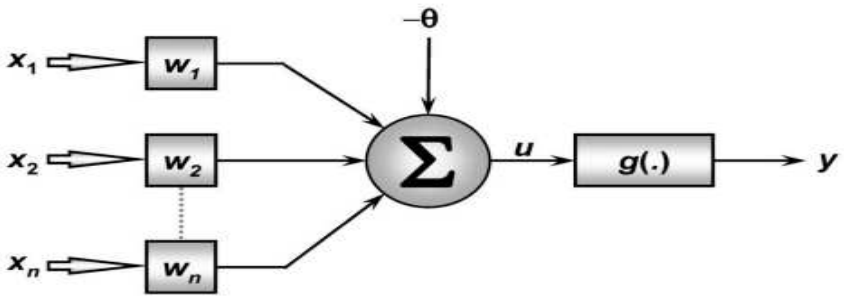
Figura 11 – Modelo ilustrativo do *Perceptron* para reconhecimento de padrões.



Fonte: Figura extraída da referência (SILVA; SPATTI; FLAUZINO, 2010).

Na Figura 12, uma representação de uma rede *Perceptron* constituída de *n* sinais de entrada, representativos do problema a ser mapeado, e somente uma saída, pois este tipo de rede é composto de um único neurônio (SILVA; SPATTI; FLAUZINO, 2010).

Figura 12 – *Perceptron*.



Fonte: Figura extraída da referência (SILVA; SPATTI; FLAUZINO, 2010).

Um *Perceptron* pode ser descrito pelas seguintes expressões das (Equação 2.1) e (Equação 2.2):

$$u = \sum_{i=1}^{n} w_i, x_i - \theta \tag{2.1}$$

$$y = g(u) \tag{2.2}$$

em que x_i são as entradas da rede, w_i é o peso associado à i-ésima entrada, θ é o limiar de ativação, *g(.)* é a função de ativação e *u* é o potencial de ativação (SILVA; SPATTI; FLAUZINO, 2010).

Resumindo, a Tabela 2 explicita os aspectos característicos dos parâmetros do *Perceptron*.

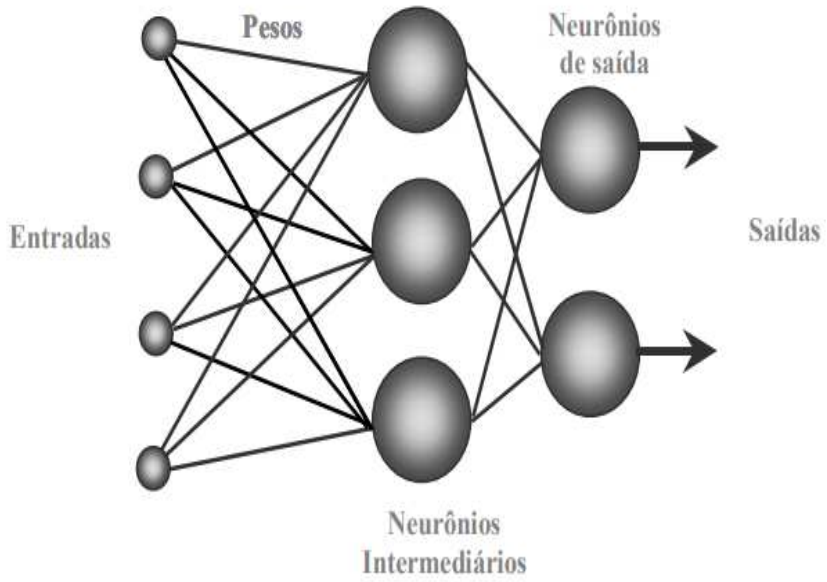
Tabela 2 – Aspectos dos parâmetros característicos do *Perceptron*.

Parâmetro	Variável representativa	Tipo característico
Entradas	x_i (i-ésima entrada)	Reais ou binárias (advindas externamente)
Pesos	w_i (associado a x_i)	Reais (iniciados aleatoriamente)
Limiar	θ	Real (iniciado aleatoriamente)
Saída	*y*	Binária
Função de ativação	*g(.)*	Degrau ou degrau bipolar
Processo de treinamento		Supervisionado
Regra de aprendizado		Regra de *Hebb*

Fonte: Tabela extraída da referência (SILVA; SPATTI; FLAUZINO, 2010).

Na Figura 13, uma representação da arquitetura *Multilayer Perceptron (MLP)*

Figura 13 – Rede com múltiplas camadas.



Fonte: Figura extraída da referência (FURTADO, 2019).

2.2.1.2 Regressão Logística

A Regressão Logística também é um algoritmo de Aprendizagem Supervisionada, uma técnica estatística que tem como objetivo realizar predições ou explicar a ocorrência de determinados fenômenos quando a variável é natureza binária (FáVERO et al., 2009).

Um dos primeiros estudos de relevância para a técnica de Regressão Logística foi o *Framingham Heart Study*. Esse trabalho, cujo objetivo foi identificar os fatores que levavam as pessoas, com idades entre 30 e 60 anos, a desenvolverem doenças cardiovasculares foi realizado em colaboração com a Universidade de *Boston* (FáVERO et al., 2009).

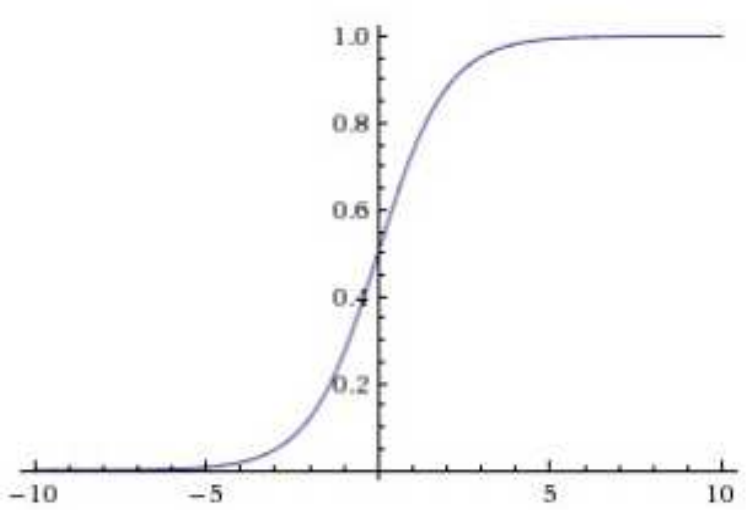
Cabe ainda destacar que essa técnica vem ganhando popularidade devido a uma série de aplicações, tais como: análise de crédito; ocorrência de uma doença; sinistro (seguradora) e, em especial, *Credit Scoring* (FáVERO et al., 2009).

Além das possibilidades acima mencionadas, a técnica de Regressão permite a flexibilização de pressupostos, o que possibilita a ampliação de suas aplicações (FáVERO et al., 2009). Seu modelo é definido pela regressão logística, dado pela seguinte função (Equação 2.3):

$$f(Z) = \frac{1}{1 + e^{-z}} \tag{2.3}$$

assume valores entre 0 e 1, para qualquer Z entre -∞ e + ∞. Na Figura 14, uma ilustração de um exemplo de Regressão Logística.

Figura 14 – Exemplo de Regressão Logística.



Fonte: Autor.

Sendo Z (Equação 2.4):

$$z = log(\frac{p}{1-p}) \tag{2.4}$$

em que z, lê-se (Equação 2.5):

$$\text{z= } \alpha \text{ + } \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \tag{2.5}$$

em que *p* indica probabilidade de ocorrência de determinado evento, *X* vetor de variáveis explicativas (ou independentes) e α e β são os parâmetros do modelo (FáVERO et al., 2009). O termo log($\frac{p}{1-p}$) é chamado de logit e o termo ($\frac{p}{1-p}$) representa a chance *(odds)* de ocorrência do evento de interesse (FáVERO et al., 2009).

Logo, se calcular a probabilidade de ocorrência é *p*= ($\frac{odds}{1-odds}$), substituindo a função (Equação 2.6) (FáVERO et al., 2009):

$$f(Z) = \frac{1}{1 + e^{-x}} \tag{2.6}$$

substituindo x da função (Equação 2.6) pela expressão (Equação 2.7):

$$x = (\alpha + \sum \beta_i X_i) \tag{2.7}$$

Matematicamente pode ser representada da seguinte forma (Equação 2.8) (FáVERO et al., 2009):

$$P(Y) = f(Y = 1 \mid x_1, X_2, ..., X_n) = \frac{1}{1+e^{-(}}\alpha + \sum \beta_i X_i) \tag{2.8}$$

2.3 Trabalhos Relacionados

Vários trabalhos sobre interação proteína-ligante com estratégias computacionais para caracterizar e predizer ligantes têm sido propostos; especificamente, o Aprendizado de Máquina. Para a realização deste trabalho, contou-se com uma grande contribuição do *GReMLIN* (SANTANA et al., 2016) baseado em mineração em grafos para encontrar padrões de interação proteína-ligante.

Além disso, utilizou-se do *visGReMLIN* (RIBEIRO et al., 2020), também baseado em grafos, com a estratégia de encontrar padrões de proteína-ligante, porém com a proposta de uma ferramenta web, interativa e visual, para poder explorar a interface proteína-ligante.

O *ppiGReMLIN* (QUEIROZ et al., 2020), foi utilizado neste trabalho, como uma estratégia no contexto proteína-proteína, para abordagem da Aprendizagem Não Supervisionada. Porém, a proposta de desenvolvimento é relacionada à mineração em grafos para encontrar padrões com base nas propriedades físico-químicas de interface com as proteínas.

3 MATERIAIS E MÉTODOS

Propõe-se neste trabalho a utilização de recursos computacionais capazes de caracterizar e de prever o surgimento de moléculas capazes de inibir proteínas, também chamadas de alvos de interesse. Destaca-se que a revisão da literatura contribuiu para o desenvolvimento da pesquisa, em que foram abordados os fundamentos teóricos dos elementos necessários para a condução dos experimentos. Neste capítulo, as seções são organizadas da seguinte forma: na primeira seção, *Fonte de Dados*, estão descritos os materiais utilizados na coleta e na análise dos dados necessários para o desenvolvimento dos experimentos. Em *Preparação e Análise das Proteínas* , foram feitas as análises dos dados das sequências de proteínas; a limpeza dos dados por meio do *Pymol* e; e a geração dos *Decoys*. Na terceira seção, *Algoritmos de Aprendizado de Máquina*, são apresentados os algoritmos e seus respectivos parâmetros para extrair informações que conseguem medir o desempenho dos classificadores e apresentar as métricas *Recall* e *Precision*. A última seção, *Docking Molecular*, apresenta uma visão geral e demonstra a importância de se identificar as interações proteína-ligante que são mais eficazes quando baseadas nos algoritmos de Aprendizagem Supervisionada. A Figura 15 mostra o passo a passo da condução dos experimentos.

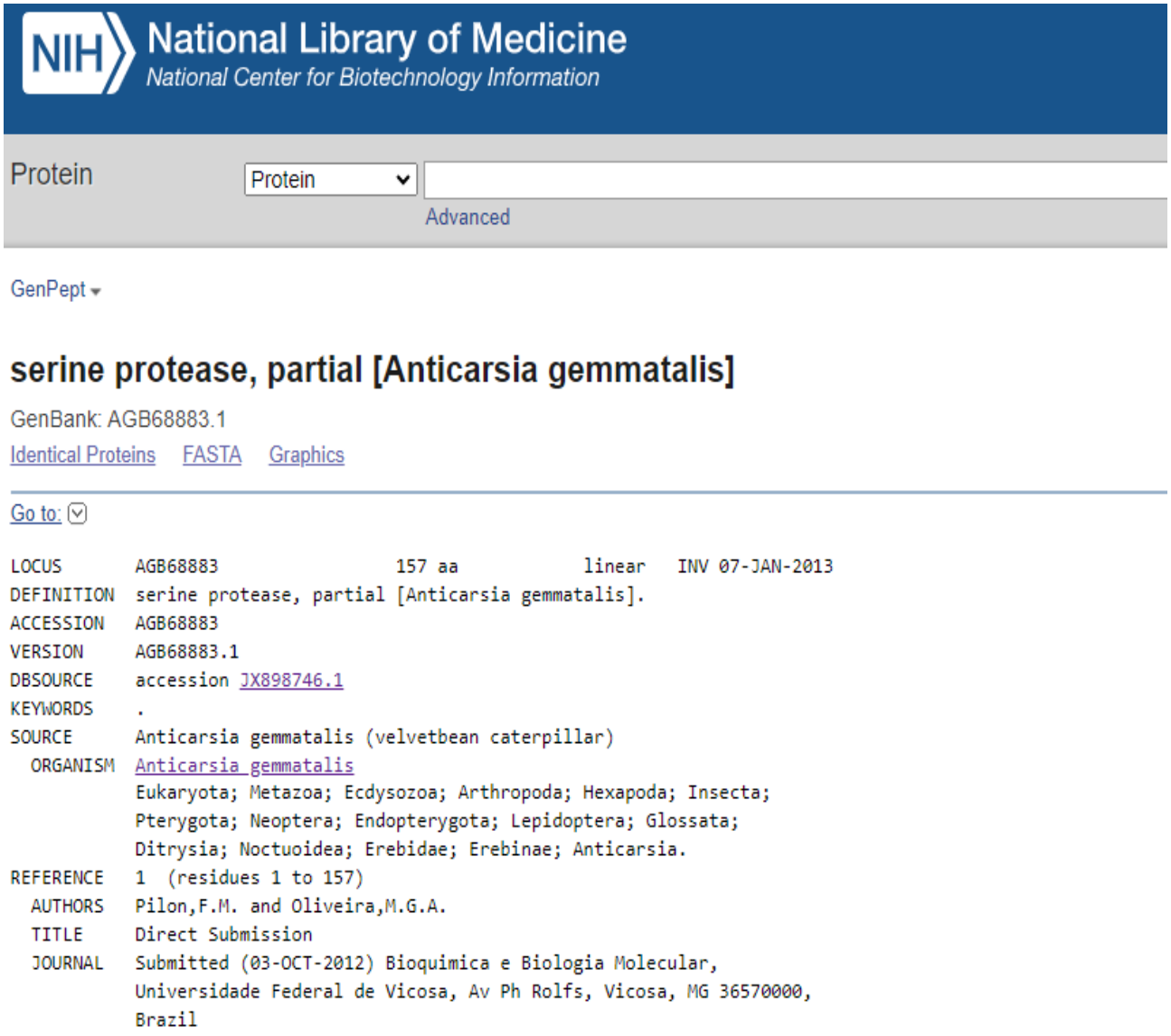
Figura 15 – Organograma da condução dos experimentos.

Fonte: Autor.

3.1 Fonte de Dados

Para construção do conjunto de dados necessários para a realização deste trabalho, procurou-se na literatura pesquisas referentes às famílias de proteases; verificou-se se o sítio ativo era conservado e buscou-se ligantes ancorados a proteínas semelhantes às da mesma família. Com o auxílio do *BLAST*, sugerido pelo próprio *GenBank*, foram retornadas 6 (seis) sequências mais similares para protease de lagarta, que pertence ao domínio *Trypsin- like serine protease*. A Figura 16 demonstra a família de protease *serine protease*.

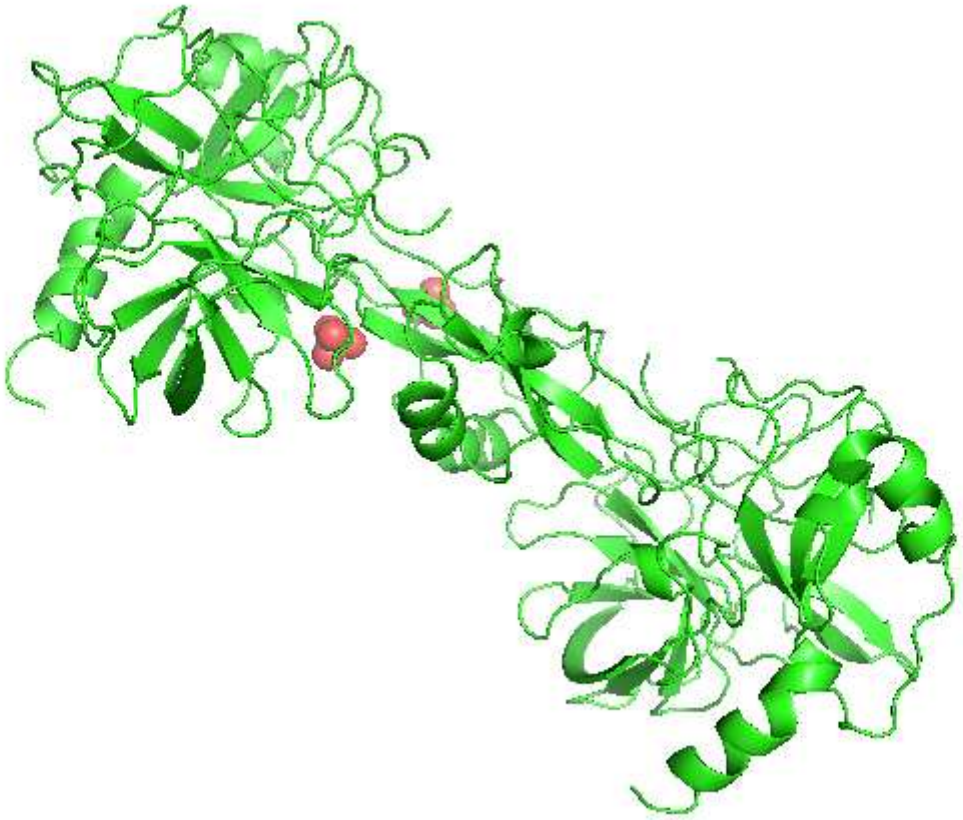
Figura 16 – Características da família *serine protease*.



Fonte: Autor.

As proteases *serine* são um grupo diverso de enzimas caracterizadas pela presença de três aminoácidos: histidina, aspartato e serina (DAVIES et al., 1998; TRIPATHI; SOWDHAMINI, 2006; CERA, 2009). A Figura 17 traz a representação de uma serina protease.

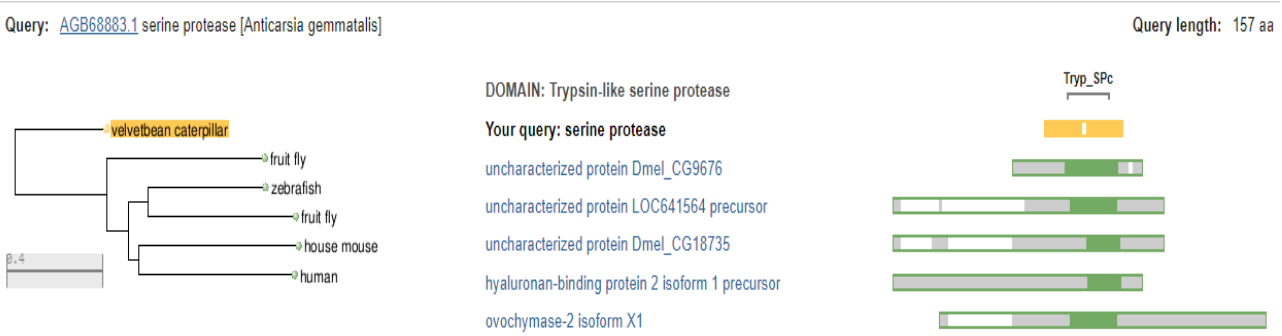
Figura 17 – Exemplo de *serina protease*.



Fonte: Autor.

Uma classe bem conhecida de serina protease é a tripsina, que é reconhecida pelo seu papel na função digestiva dos alimentos (SZMOLA; KUKOR; SAHIN-TOTH, 2003; WANG; LUO; REISER, 2008; KOISTINEN; KOISTINEN; ZHANG, 2009). Ressalta-se que ela é produzida pela clivagem do tripsinogênio por meio dos enteroquinases no trato digestivo (LUO; WANG; REISER, 2007; WANG; LUO; REISER, 2008; KOISTINEN; KOISTINEN; ZHANG, 2009). A Figura 18 representa a análise da *Trypsin-like serine protease*.

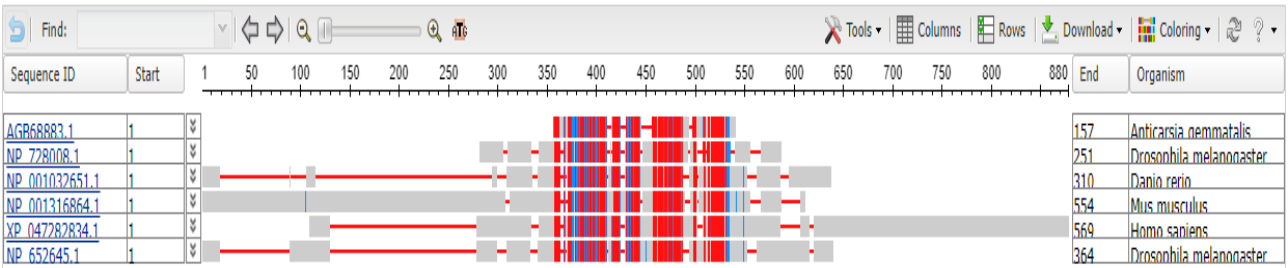
Figura 18 – *Trypsin-like serine protease* no *BLAST*.



Fonte: Autor.

A Figura 19, representa a análise do alinhamento das sequências de proteínas. Percebemos que existem várias regiões bastante conservadas (em vermelho) e algumas com conservação relativa (em azul).

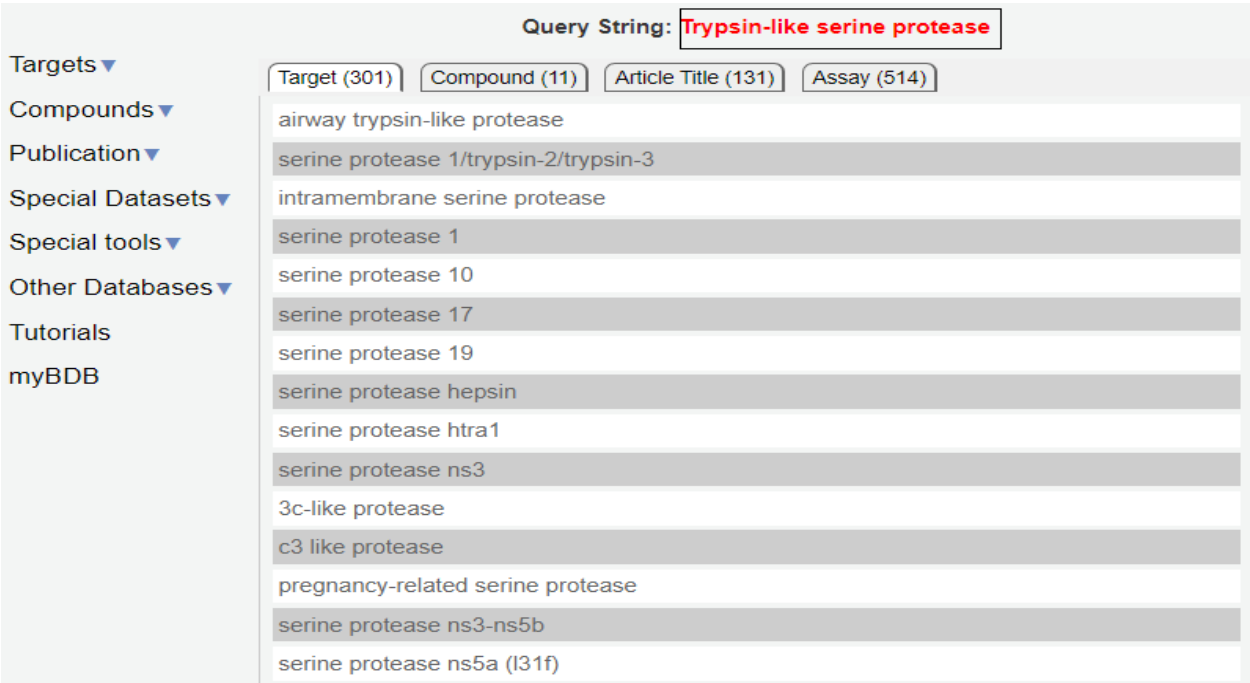
Figura 19 – Alinhamento das sequências de proteínas no *BLAST*.



Fonte: Autor.

Os experimentos foram conduzidos com proteases similares à protease de lagarta, chamada *Trypsin-like serine protease*. Na Figura 20, tem-se uma demonstração da busca de *dataset*, que corresponde à família serine protease no *BindingDB*, também conhecido como *Binding Database*, um banco de dados com diversas proteínas consideradas alvos de interesse (CHEN; LIN; GILSON, 2001). Nota-se que uma das palavras-chave utilizadas foi *Trypsin-like serine protease*, a qual foi identificada como palavra-chave no alinhamento realizado anteriormente.

Figura 20 – Busca de dataset no *BindingDB*.



Fonte: Autor.

Encontrou-se, então, por meio *BindingDB*, as famílias de proteínas, que se assemelham à protease, tais como:

- **Transmembrane protease serine 11D (Homo sapiens) :** só retornou uma estrutura, 2E7V e teve bom alinhamento no *Pymol*;
- **Subtilisin-like serine protease (Plasmodium falciparum) :** retornaram 945 estruturas;
- **Transmembrane protease serine 2 (Homo sapiens) :** retornaram 53195 estruturas;
- **Kallikrein 4 e 8, Hepsin e serine protease HTRA1 :** retornaram 53195 estruturas;
- **Serine protease NS3 (Hepatitis C virus genotype 1) :** retornaram 945 estruturas.

Por meio da ferramenta *Pymol* (PYMOL, 2023), conseguiu-se chegar ao *Subtilisin-like serine protease (Plasmodium falciparum)*.

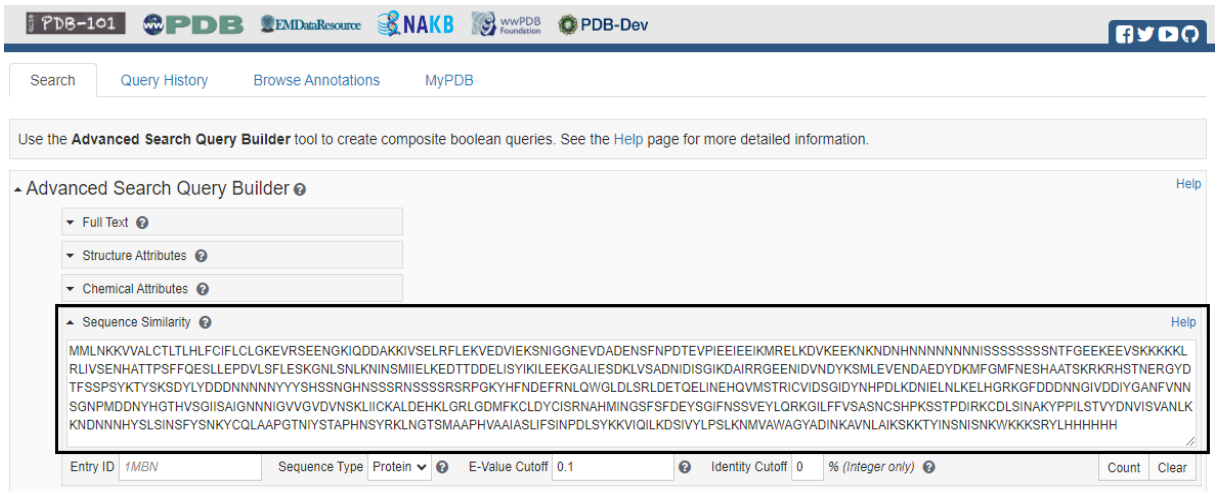
3.2 Preparação e Análise das Proteínas

Nesta seção, apresenta-se a preparação da construção do conjunto de dados a partir da família da protease *Subtilisin-like serine protease (Plasmodium falciparum)*.

1. Inicialmente, foram obtidas as sequências das proteínas, através do *BindingDB*. Porém, para encontrar as respectivas proteínas correspondente das sequências de proteínas utilizamos o *Protein Data Bank (PDB)* (JIN et al., 2015; PDB, 2023), conforme na Figura 22.

   Na Figura 21, em destaque apresenta a sequência da proteína.

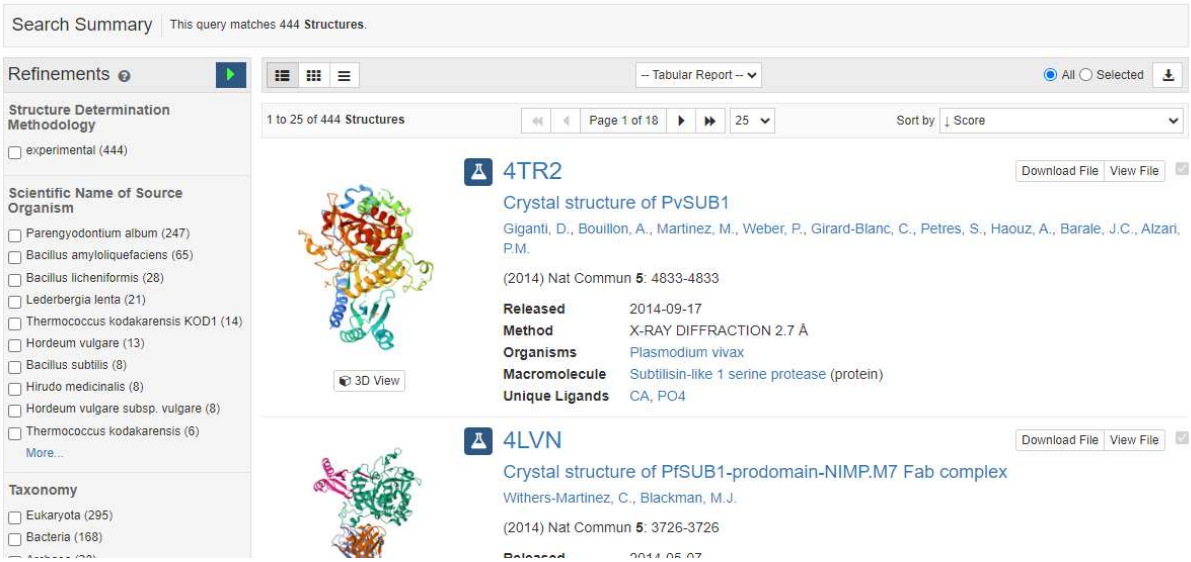
Figura 21 – Construção da *query* a partir da sequência.



Fonte: Autor.

Na Figura 22, representa o resultado gerado pela a *query* a partir da sequência da proteína.

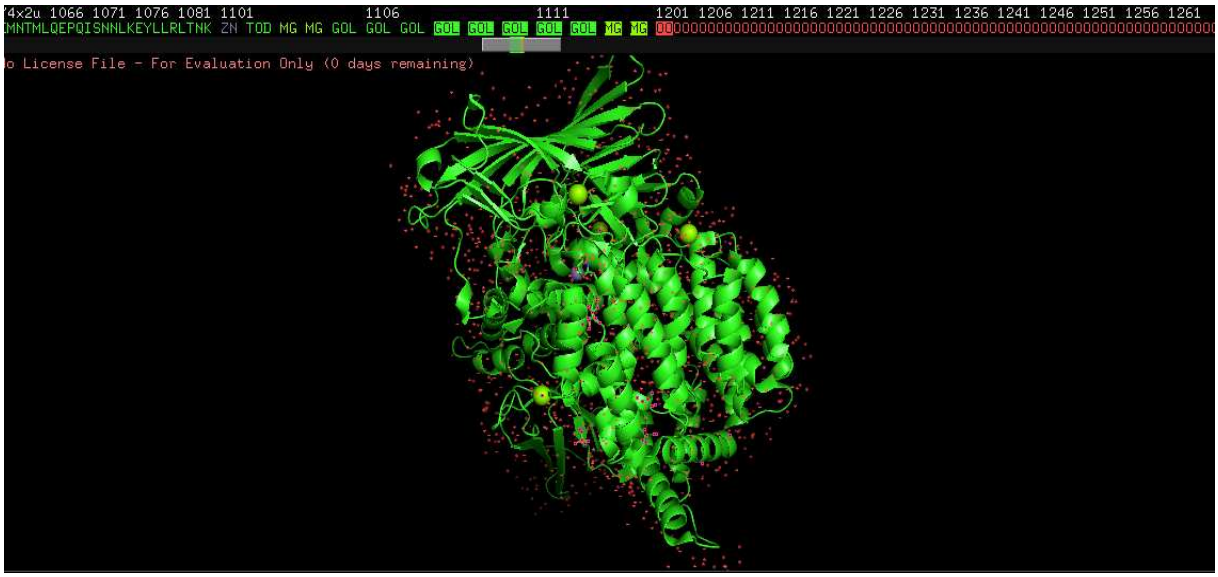
Figura 22 – Resultado da *query* a partir da sequência da proteína.



Fonte: Autor.

2. Remoção das moléculas de água e de íons presente nas proteínas. Após o resultado da *query*, obteve-se um conjunto de proteínas correspondente à família da protease. Com apoio do *Pymol*, executou-se o comando para exibir a sequência. A Figura 23 apresenta uma proteína já com alguns íons em destaque selecionados e com diversos pontos em vermelho, que são as moléculas de água

Figura 23 – Tratamento da sequência de proteína, com a remoção de água e de íons no *Pymol*.



Fonte: Autor.

Na Figura 24, uma demonstração com todas moléculas de água e íons removidos.

Figura 24 – Sequência de proteína com água e com íons removidos no *Pymol*.

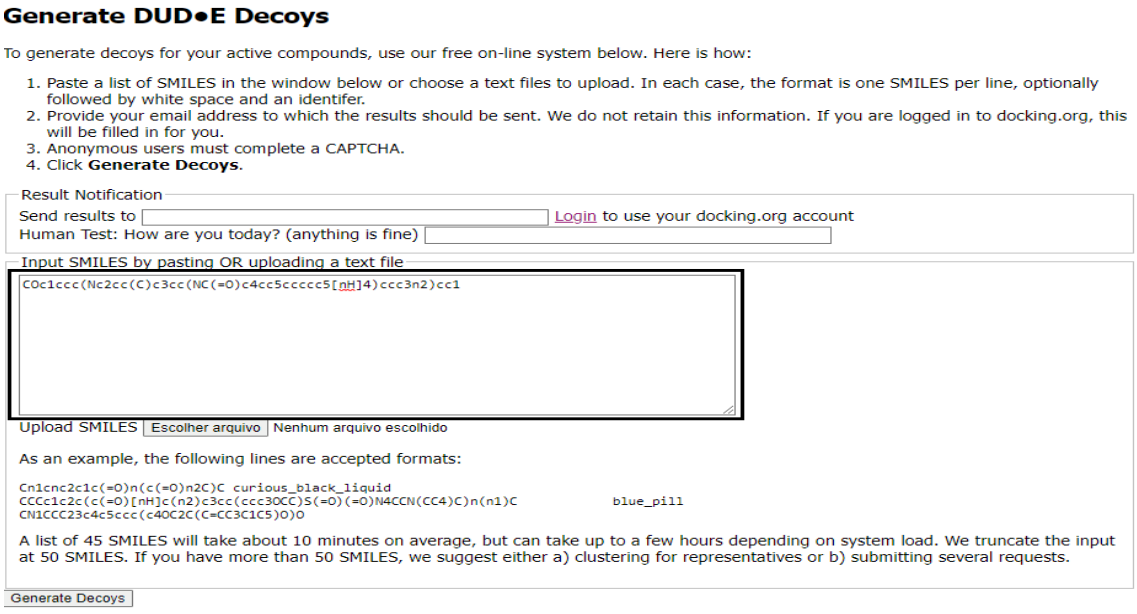
Fonte: Autor.

3. Geração dos *Decoys* (DECOYS, 2023). Novamente, com o apoio *BindingDB* encontrou-se, também, os *Smiles*, que são uma notação química de uma estrutura molecular, conforme o exemplo:

   COc1ccc(Nc2cc(C)c3cc(NC(=O)c4cc5ccccc5[nH]4)ccc3n2)cc1

   A Figura 25 mostra a geração de *Decoys*, a partir da informação do *Smiles* em destaque.

Figura 25 – Repositório de geração de *Decoys*.



Fonte: Autor.

3.3 Algoritmos de Aprendizado de Máquina

Nesta seção, são apresentados os algoritmos de Aprendizagem Supervisionada, bem como suas características e parâmetros que contribuíram para a obtenção de um melhor desempenho perante os demais classificadores. Destacam-se aqui a *LogisticRegression* e a *MLPClassifier* com suas respectivas características.

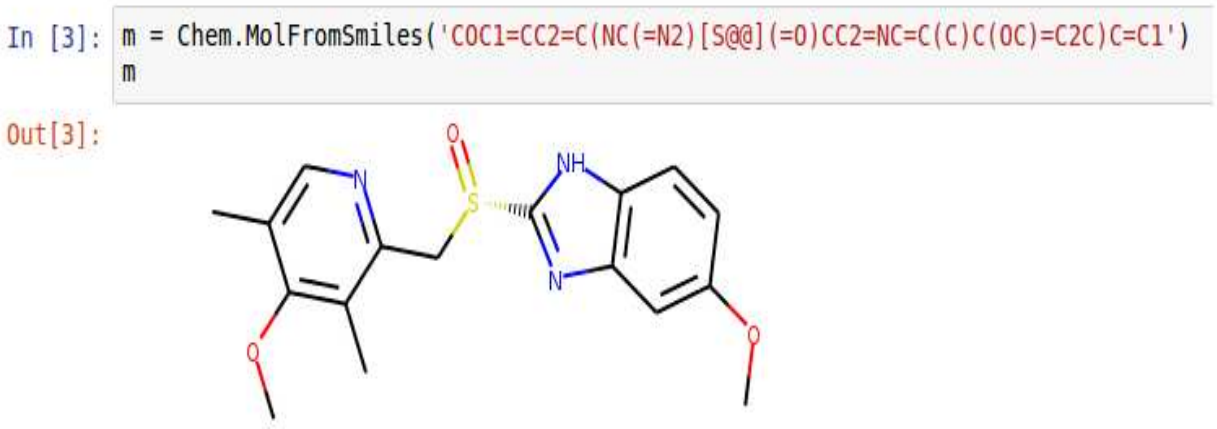
Ressalta-se que as métricas são importantíssimas para se avaliar a qualidade do modelo e encontrar o alvo de interesse; no caso, a proteína. Para isso, foram utilizadas as seguintes métricas: *Precision* e *Recall*, que possibilitam identificar entre os algoritmos, qual deles obteve melhor performance com base nas medições estatísticas e na validação cruzada.

3.3.1 Pacotes *RDkit*

Para a condução deste trabalho, contou-se com a contribuição do *RDKit*, que é um *software* de código aberto voltado para quimioinformática e sua interface de programação nas linguagens de programação *Python*, *Java*, *C++* e *C#* (RDKIT, 2023). Ele inclui diversas funcionalidades, tais como: geração de descritor de Aprendizado de Máquina, moléculas em *2D* e *3D*, *fingerprints* (impressão digital), reações químicas, entre outros (RDKIT, 2023).

Esse *software* foi desenvolvido pelo *Greg Landrum* e conta com inúmeras contribuições da comunidade para aperfeiçoar e implementar melhorias (RDKIT, 2023). Na Figura 26, tem-se uma demonstração de molécula gerada por ele.

Figura 26 – Molécula gerada pelo *software RDkit*.



Fonte: Autor.

3.3.2 Parâmetros dos Algoritmos de Aprendizagem Supervisionada

Os algoritmos de Aprendizado Supervisionada abordados neste trabalho foram o *Multilayer Perceptron* e Regressão Logística. Portanto, serão descritos seus

respectivos parâmetros e sua importância para condução dos experimentos.

Na Figura 27, tem-se a representação do algoritmo *LogisticRegression*, com o seguinte parâmetro:

- **random-state:** produziu os mesmos resultados em diferentes execuções do algoritmo (SCIKIT, 2023).

Figura 27 – Parâmetros do algoritmo *LogisticRegression*.

```
LogisticRegression(random_state=0)
```

Fonte: Autor.

Na Figura 28, tem-se a representação dos algoritmos *MLPClassifier*, com os seguintes parâmetros:

- **alpha:** constante que determina a intensidade de regularização do algoritmo. A regularização é uma técnica capaz de reduzir a possibilidade de erro ao realizar o treinamento do algoritmo (SCIKIT, 2023);
- **learning-rate-init:** parâmetro de ajuste que determina o tamanho e o peso a cada iteração do algoritmo (SCIKIT, 2023);
- **max-iter:** número máximo de iterações necessárias para convergir (SCIKIT, 2023).
- **hidden-layer-size:** número de neurônios na camada oculta (SCIKIT, 2023).

Figura 28 – Parâmetros dos algoritmos *MLPClassifier*.

```
MLPClassifier(alpha=0.0001, learning_rate_init=0.1, max_iter=500, hidden_layer_sizes=(100,75,50,25,))
MLPClassifier(alpha=0.1, learning_rate_init=0.001, max_iter=500)
MLPClassifier(alpha=0.01, learning_rate_init=0.01, max_iter=500)
```

Fonte: Autor.

3.3.3 Métricas

As métricas são importantíssimas para se estimar o desempenho dos algoritmos de classificação. Elas são baseadas em predições corretas e incorretas dentro de um grupo de teste. Para a realização dessa estimativa, foram utilizadas as seguintes métricas: *Precision* e *Recall*, que possibilitam identificar qual algoritmo é capaz de encontrar os ligantes para a proteína-alvo de interesse. De acordo com (SILVA; PERES; BOSCARIOLI, 2016), as predições podem ter as seguintes classes:

- **Verdadeiro positivo (VP) :** quando a classificação correta na classe positiva, pertence à classe **positiva**, e o classificador vê que a classe era realmente **positiva** (SILVA; PERES; BOSCARIOLI, 2016);
- **Verdadeiro negativo (VN) :** quando a classificação correta na classe negativa, pertence à classe **negativa**, e o classificador vê que a classe era realmente **negativa** (SILVA; PERES; BOSCARIOLI, 2016);
- **Falso positivo (FP) :** quando a classificação incorreta na classe positiva, pertence à classe **positiva**, e o classificador vê que a classe era realmente **positiva** (SILVA; PERES; BOSCARIOLI, 2016);
- **Falso negativo (FN) :** quando a classificação incorreta na classe negativa, pertence à classe **negativa**, e o classificador vê que a classe era realmente **negativa** (SILVA; PERES; BOSCARIOLI, 2016);

Uma maneira simples de entender as predições e as possíveis classes é por meio da Matriz de Confusão, conforme na Tabela 3.

Tabela 3 – Matriz de Confusão para um problema de classificação binária.

		Valores Previstos	
		Positivo	Negativo
Valores Reais	Positivo	VP	FP
	Negativo	FN	VN

Fonte: Autor.

Para se avaliar adequadamente os algoritmos de classificação, ressaltam-se duas métricas utilizadas para esse fim:

- **Precision:** porcentagem de verdadeiros positivos que foram realmente classificados como positivos (Equação 3.1) (SILVA; PERES; BOSCARIOLI, 2016);

$$Precision = \frac{VP}{VP + FP} \tag{3.1}$$

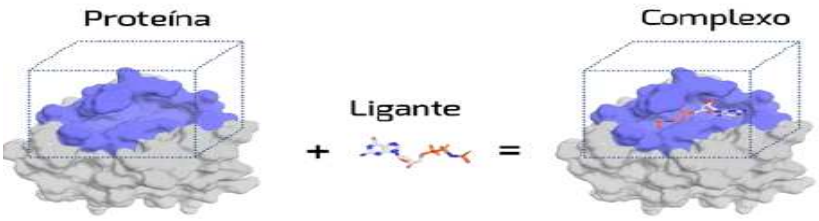
- **Recall:** porcentagem de verdadeiros positivos, cujo resultado esperado é a classe positiva (Equação 3.2) (SILVA; PERES; BOSCARIOLI, 2016).

$$Recall = \frac{VP}{VP + FN} \tag{3.2}$$

3.4 *Docking Molecular*

O *Docking Molecular*, também conhecido como Atracamento Molecular é uma técnica computacional, cuja a finalidade é prever os modos de ligação entre o ligante e o receptor. A Figura 29 mostra o atracamento entre ligante e proteína.

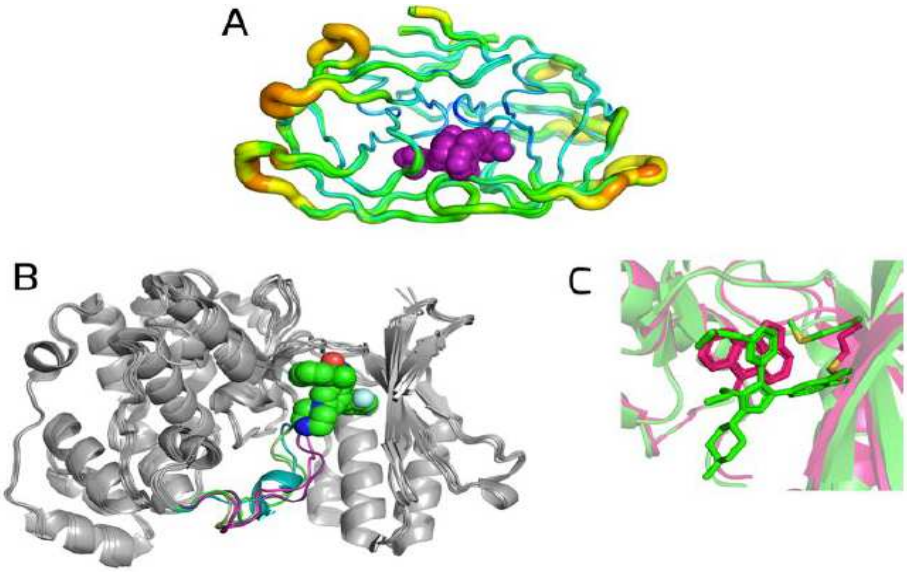
Figura 29 – Emprego do método de atracamento molecular na predição do modo de ligação do *GTP* ao seu sítio de ligação na proteína *c-H-ras p21*.



Fonte: Figura extraída da referência (VERLI, 2014).

Destaca-se que o *Docking*, uma técnica baseada ”chave-fechadura”, foi proposta por *Emil Fisher* (VERLI, 2014). Nessa técnica, o receptor é associado a uma fechadura, e seu sítio de ativação ao ”buraco- fechadura”. A possível fechadura é o ligante, e a interação entre esse e a proteína está relacionada à possibilidade de ”abrir ou fechar ”a porta (VERLI, 2014). Durante o processo de interação entre proteína-ligante, tanto o ligante quanto a proteína são flexíveis, o que modifica a sua conformação durante o processo de formação do complexo receptor-ligante (VERLI, 2014). A Figura 30 representa os graus de flexibilidade do receptor.

Figura 30 – Graus de flexibilidade do receptor.



Fonte: Figura extraída da referência (VERLI, 2014).

Os ligantes e as proteínas possuem características em comum, tais como:

- a superfície de contato molecular é definida por *van der Waals* (VERLI, 2014);
- alta complementaridade de propriedades associadas às superfícies de contato moleculares (VERLI, 2014);
- o ligante geralmente se liga por meio de conformação energeticamente favorável (VERLI, 2014);
- minimização de interações repulsivas entre ligantes e proteínas (VERLI, 2014).

3.4.1 Interações proteína-ligante

Os principais tipos de interações entre proteína-ligante, incluem (VERLI, 2014):

- ligações de hidrogênio (VERLI, 2014);
- interações de *van der Waals* (VERLI, 2014);
- interações iônicas (VERLI, 2014);
- interações hidrofóbicas (VERLI, 2014);
- interações do tipo cátion-π (VERLI, 2014);
- interações envolvendo anéis aromáticos(VERLI, 2014);
- coordenação com íons metálicos (VERLI, 2014).

As interações entre o sítio ativo e as partes apolares do ligante dão origem ao efeito hidrofóbico (VERLI, 2014), que causa a liberação e a desorganização das moléculas de água, durante a interação proteína-ligante, e o aumento de entropia; o que, consequentemente, contribui para a formação de complexo da proteína-ligante (VERLI, 2014).

Além disso, o efeito hidrofóbico tem um papel importante no processo de *docking*, pois algumas dessas moléculas de água podem ser consideradas moléculas estruturais (VERLI, 2014). É importante ressaltar que essas moléculas estão associadas ao sítio ativo, porém são conservadas em sítios de ligação das proteínas (VERLI, 2014).

Consequentemente, essas moléculas podem interferir no acesso do ligante ao sítio ativo e modificar a formação de ligações de hidrogênio, o que contribui para o *docking* (VERLI, 2014).

Os efeitos podem ser estimados pela energia livre de ligação que está relacionado à constante de equilíbrio de ligação K_{eq}, que pode ser medida da seguinte maneira (Equação 3.3) (VERLI, 2014):

$$\triangle G_{lig} = \triangle H - T\triangle S = -RT \log K_{eq} \tag{3.3}$$

em que $\triangle H$ é a variação de entalpia, T é a temperatura absoluta, $\triangle S$ é a variação de entropia e R é a constante universal dos gases (VERLI, 2014).

A constante de equilíbrio pode ser representada pela constante de dissociação K_d (Equação 3.4) ou de associação K_a (Equação 3.5), o que chega na seguinte representação:

$$K_d = \frac{([R][L])}{[RL]} \tag{3.4}$$

$$K_a = \frac{([RL])}{[R][L]} \tag{3.5}$$

onde *[R]*, *[L]* e *[RL]* são as concentrações de receptor, do ligante e do receptor-ligante respectivamente (VERLI, 2014).

Existem diversos programas que auxilia na execução *Docking Molecular*, conforme na Tabela 4.

Tabela 4 – Portais de acesso para alguns programas de *Docking Molecular*.

Potal	Programa de *Docking Molecular*
SwissDock	*EADock DSS*
DockingServer	*AutoDock*
DockThor Portal	*DockThor*
1-Click Docking	*AutoDock Vina*
DOCK Blaster	*DOCK*
Docking At UTMB	*AutoDock Vina*
ParDOCK	Método de *Monte Carlo*
PATCHDOCK	*PatchDock*
MEDock	*MEDock*

Fonte: Tabela extraída da referência (VERLI, 2014).

4 RESULTADOS E DISCUSSÃO

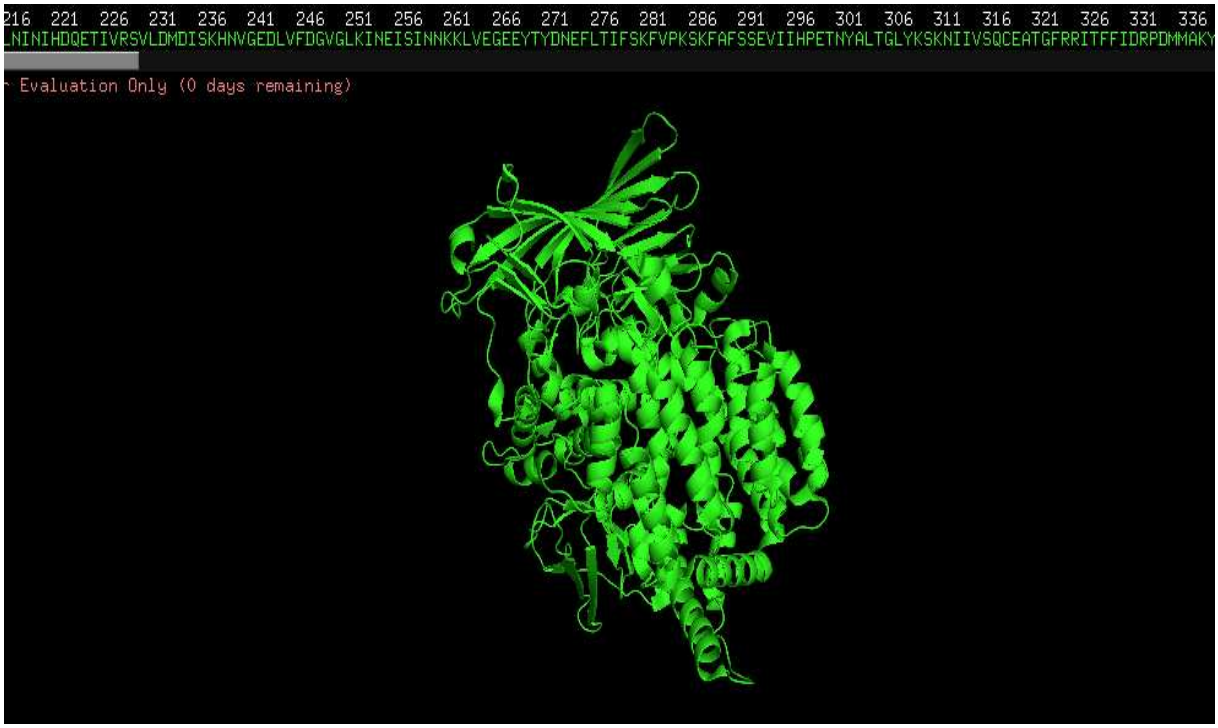
Este capítulo traz, de forma prática, os resultados obtidos nos experimentos conduzidos na realização deste trabalho, os quais contribuíram para prever os ligantes (também chamados de alvos). Evidenciou-se que os algoritmos de Aprendizado de Máquina e suas respectivas características, colaboraram se obter um melhor desempenho em relação aos demais algoritmos e também para se demonstrar a técnica de *Docking de Molecular*.

Constatou-se também que a aplicação de boas práticas na condução de experimentos pode impactar positivamente o desempenho e os resultados, tanto no tempo quanto no custo de sua execução.

Objetivou-se evidenciar quais fatores permitiram a obtenção de melhores resultados. Para isso, baseou-se em suas respectivas características e nas métricas utilizadas para se comprovar a eficácia dos modelos; a partir dos quais, executou-se o *docking* para se identificar os padrões de interação proteína-ligante.

Primeiramente, demonstrou-se uma das proteínas que compõe o conjunto de dados dos experimentos. Ela pertence à família de protease *Subtilisin-like serine protease (Plasmodium falciparum)*. Na Figura 31, encontra-se uma representação de sua estrutura.

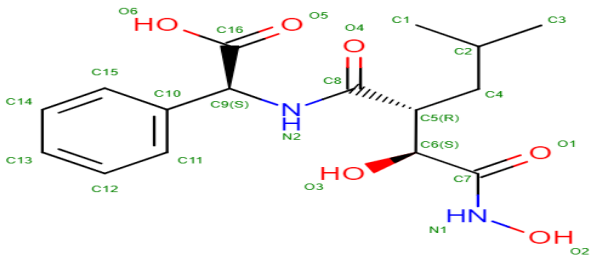
Figura 31 – Proteína correspondente à família da protease *Subtilisin-like serine protease (Plasmodium falciparum)*.



Fonte: Autor.

Na Figura 32, uma representação do ligante correspondente à proteína.

Figura 32 – Ligante da proteína correspondente à família da protease *Subtilisin-like serine protease (Plasmodium falciparum)*.



Fonte: Autor.

Iniciou-se, então, o treinamento dos algoritmos de Aprendizado de Máquina e ao *Docking Molecular*.

4.1 Matriz de Confusão

Para evidenciar a estratégia proposta neste trabalho e mostrar sua aplicação prática, foram coletados dados estruturais de proteínas com ligantes no *PDB (Protein Data Bank)* correspondentes à família de protease *Subtilisin-like serine protease (Plasmodium falciparum)*.

Inicialmente, realizou-se o treinamento dos classificadores com os ligantes e *decoys*, para identificar sua afinidade com a base *drugbank* (DRUGBANK, 2023), que é a base de dados que combina detalhes químicos e farmacológicos com informações relevantes dos seus alvos, ou seja, as proteínas. Na Figura 33, representa a matriz de confusão.

Figura 33 – *Matrix Generator*.

name	0	1	2	3	4	5	6	7	8	9	...	2206	2207	2208	2209	2210	2211	2212	2213	2214	class
ZINC29336179	1	0	1	0	1	1	0	1	0	0	...	1	1	1	1	1	1	1	1	0	1
ZINC29338041	1	1	1	0	0	1	0	1	0	0	...	1	1	0	1	1	1	1	1	0	1
ZINC29333180	1	1	0	0	1	0	1	0	1	1	...	0	1	0	0	1	1	1	1	0	1
ZINC29327748	0	1	1	0	0	0	0	1	0	0	...	1	1	1	1	0	0	1	0	0	1
ZINC01490698	0	1	1	0	0	0	0	1	0	0	...	1	1	1	1	0	0	1	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
C85341116	1	0	1	1	0	1	1	1	1	1	...	0	1	1	0	1	1	1	1	0	0
C13536720	1	0	1	0	1	1	1	0	1	0	...	1	1	1	1	1	1	1	1	0	0
C13508104	1	1	0	1	0	0	0	0	1	1	...	1	1	0	1	1	1	1	1	0	0
C97963564	0	1	1	1	1	1	1	0	1	1	...	1	1	1	1	1	1	1	1	0	0
C98002536	1	0	1	0	0	1	0	1	1	0	...	1	1	0	1	1	1	1	1	0	0

Fonte: Autor.

Na figura 33, tem-se a representação da Matrix Generator, em que a coluna *name* indica o código do ligante e a *class* representa a classe à qual pertence, ou seja, 0 valores negativos (*decoys*) e 1 valores positivos (ligantes).

4.2 Classificadores

Os experimentos foram conduzidos com classificadores *LogisticRegression* e *MLPClassifier*, com diferentes parâmetros e características. Os resultados obtidos representam a distribuição dos compostos químicos da base do *drugbank*. O eixo X representa as moléculas, e o eixo Y suas probabilidades das respectivas moléculas.

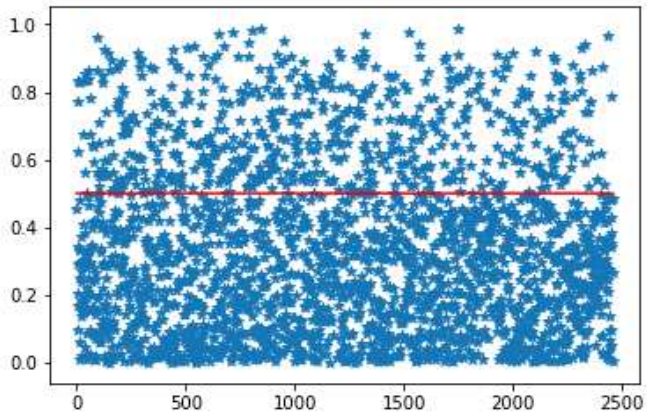
Além disso, conseguiu-se identificar quais medicamentos correspondem aos classificadores.

4.2.1 *LogisticRegression*

A Figura 34 representa o classificador *LogisticRegression* com o parâmetro **(random-state = 0)**: a distribuição uniforme dos compostos químicos e suas respectivas probabilidades demonstraram um excelente resultado. À medida que a quantidade de compostos aumentavam, a probabilidade se mantinha estável. A expressão dessa ação é dada da seguinte forma:

LogisticRegression (random-state = 0)

Figura 34 – *Matrix Generator* gerada a partir do classificador *LogisticRegression*.



Fonte: Autor.

Na Tabela 5, observa-se que os medicamentos obtidos a partir do classificador *LogisticRegression* tiveram resultados acima de 0.90.

Tabela 5 – Resultados do classificador *LogisticRegression*.

Name	*Value*	*Generic Name*
DB01009	0.987215	*Ketoprofen*
DB09214	0.987215	*Dexketoprofen*
DB00963	0.979776	*Bromfenac*
DB06802	0.977622	*Nepafenac*
DB00870	0.974952	*Suprofen*
DB00803	0.970743	*Colistin*
DB04838	0.969710	*Cyclandelate*
DB00781	0.967954	*Polymyxin B*
DB00209	0.962970	*Trospium*
DB01123	0.952035	*Proflavine*
DB00946	0.940734	*Phenprocoumon*
DB04120	0.940734	*4-Methyl-1,2-Benzenediol*
DB08869	0.939650	*Egrifta*

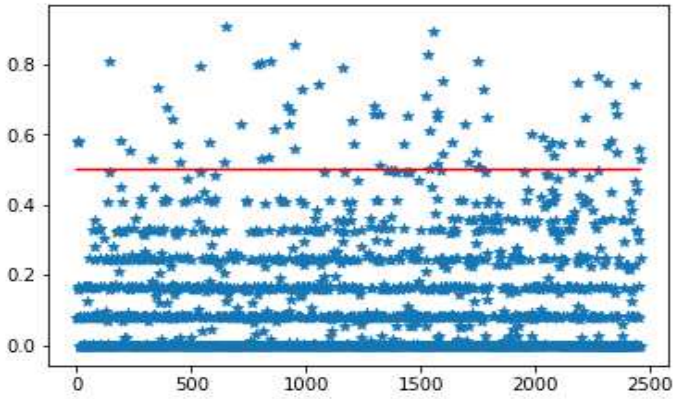
Fonte: Autor.

4.2.2 *MLPClassifier*

A Figura 35 representa o classificador *MLPClassifier* com os parâmetros **(alpha = 0.0001, learning-rate-init = 0.1, max-iter = 500, hidden-layer-sizes = (100, 75, 50, 25,))** ou seja, com distribuições desproporcionais dos compostos químicos. Isso dificultou medir a eficácia do classificador, e os resultados obtidos ficaram abaixo do esperado. A expressão dessa ação é dada da seguinte forma:

MLPClassifier(alpha = 0.0001, learning-rate-init = 0.1, max-iter = 500, hidden-layer-sizes = (100, 75, 50, 25,))

Figura 35 – *Matrix Generator* gerada a partir do classificador *MLPClassifier*.



Fonte: Autor.

Na Tabela 6, observa-se que os medicamentos obtidos a partir do classificador *MLPClassifier* tiveram resultados entre 0,75 e 0,90, considerando o alpha de 0.0001.

Tabela 6 – Resultados do classificador *MLPClassifier*.

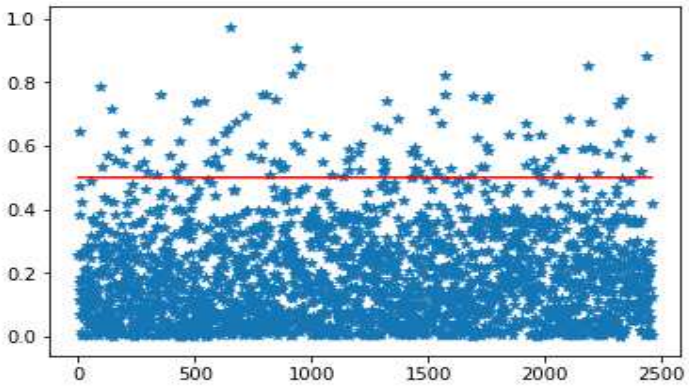
Name	*Value*	*Generic Name*
DB00803	0.907164	*Colistin*
DB08820	0.894923	*Ivacaftor*
DB01123	0.856811	*Proflavine*
DB06816	0.828154	*Pyrvinium*
DB00266	0.810361	*Dicoumarol*
DB09214	0.808084	*Dexketoprofen*
DB01009	0.808084	*Ketoprofen*
DB00963	0.806521	*Bromfenac*
DB00946	0.799978	*Phenprocoumon*
DB00682	0.793855	*Warfarin*
DB01418	0.791995	*Acenocoumarol*
DB13931	0.768379	*Netarsudil*
DB08907	0.750643	*Canagliflozin*

Fonte: Autor.

A Figura 36 representa o classificador *MLPClassifier* com os parâmetros **(alpha = 0.1, learning-rate-init = 0.001, max-iter = 500)**, com distribuição uniforme dos compostos químicos. Isso demonstrou um excelente resultado. À medida que a quantidade de compostos aumentavam, a probabilidade mantinha-se estável. A expressão dessa ação é dada da seguinte forma:

MLPClassifier(alpha = 0.1, learning-rate-init = 0.001, max-iter = 500)

Figura 36 – *Matrix Generator* gerada a partir do classificador *MLPClassifier*.



Fonte: Autor.

Na Tabela 7, observa-se que os medicamentos obtidos a partir do classificador *MLPClassifier* tiveram resultados entre 0.74 e próximo de 1 (um).

Tabela 7 – Resultados do classificador *MLPClassifier*.

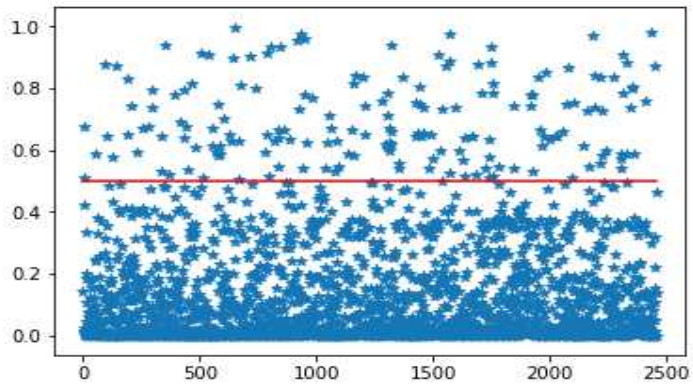
Name	*Value*	*Generic Name*
DB00803	0.973272	*Colistin*
DB01111	0.909206	*Colistimethate*
DB00781	0.881615	*Polymyxin B*
DB01123	0.850477	*Proflavine*
DB13170	0.849646	*Plecanatide*
DB01087	0.826991	*Primaquine*
DB08869	0.823831	*Tesamorelin*
DB00209	0.788782	*Trospium*
DB00946	0.762845	*Phenprocoumon*
DB00963	0.761432	*Bromfenac*
DB08875	0.760676	*Cabozantinib*
DB00484	0.759753	*Brimonidine*
DB09115	0.755411	*Diiodohydroxyquinoline*
DB09214	0.748042	*Dexketoprofen*
DB01009	0.748042	*Ketoprofen*

Fonte: Autor.

A Figura 37 representa o classificador *MLPClassifier* com os parâmetros **(alpha = 0.01, learning-rate-init = 0.01, max-iter = 500)**, ou seja, com distribuição uniforme dos compostos químicos. Nessa caso, os resultados foram muito bons; porém, não obtiveram a mesma qualidade do modelo com alpha de 0.1. A expressão dessa ação é dada da seguinte forma:

MLPClassifier(alpha = 0.01, learning-rate-init = 0.01, max-iter = 500)

Figura 37 – *Matrix Generator* gerada a partir do classificador *MLPClassifier*.



Fonte: Autor.

Na Tabela 8, observa-se que os medicamentos obtidos a partir do classificador *MLPClassifier* tiveram resultados acima de 0,90, considerando alpha de 0.01.

Tabela 8 – Resultados do classificador *MLPClassifier*.

Name	*Value*	*Generic Name*
DB00803	0.997074	*Colistin*
DB00781	0.982835	*Polymyxin B*
DB01111	0.977692	*Colistimethate*
DB08869	0.976216	*Tesamorelin*
DB13170	0.97100	*Plecanatide*
DB01123	0.957962	*Proflavine*
DB01087	0.957485	*Primaquine*
DB00484	0.941784	*Brimonidine*
DB04838	0.939444	*Cyclandelate*
DB00963	0.934958	*Bromfenac*
DB09214	0.933790	*Dexketoprofen*
DB01009	0.933790	*Ketoprofen*
DB00946	0.914486	*Phenprocoumon*
DB00643	0.912732	*Mebendazole*
DB00682	0.910971	*Warfarin*
DB06802	0.910662	*Nepafenac*
DB14086	0.905979	*Cianidanol*
DB00870	0.905979	*Suprofen*

Fonte: Autor.

4.2.3 Métricas dos Classificadores

As métricas de *Precision* e *Recall* obtiveram resultados relevantes nos algoritmos de *LogisticRegression* e *MLPClassifier*. No caso do *MLPClassifier*, com parâmetro **alpha de 0.1**. Ressalta-se que os parâmetros têm grande influência na qualidade e desempenho do classificador, o que pode ser observado no caso do *MLPClassifier* com parâmetro alpha de 0.1. Na Tabela 9, estão representados os algoritmos de maior de relevância obtidos dos experimentos.

Tabela 9 – Métricas dos Classificadores *LogisticRegression* e *MLPClassifier*.

Classificadores	*Precision*	*Recall*
LogisticRegression	0.9858	0.9896
MLPClassifier(alpha = 0.1, learning-rate-init = 0.001, max-iter = 500)	0.9710	0.9730

Fonte: Autor.

4.3 *Docking Molecular*

Nessa etapa, observou-se inicialmente que a contribuição do treinamento com os classificadores de Aprendizado de Máquina permitiu identificar quais proteínas seriam utilizadas no processo de *Docking Molecular*. Na Tabela 10 são apresentas proteínas da família de protease *Subtilisin-like serine protease (Plasmodium falciparum)*.

Contou-se também com o apoio de *softwares* auxiliares para condução dos experimentos, tais como: *Open Babel*, *Discovery Studio* e *Autodock*.

Tabela 10 – Moléculas utilizadas no *Docking Molecular*.

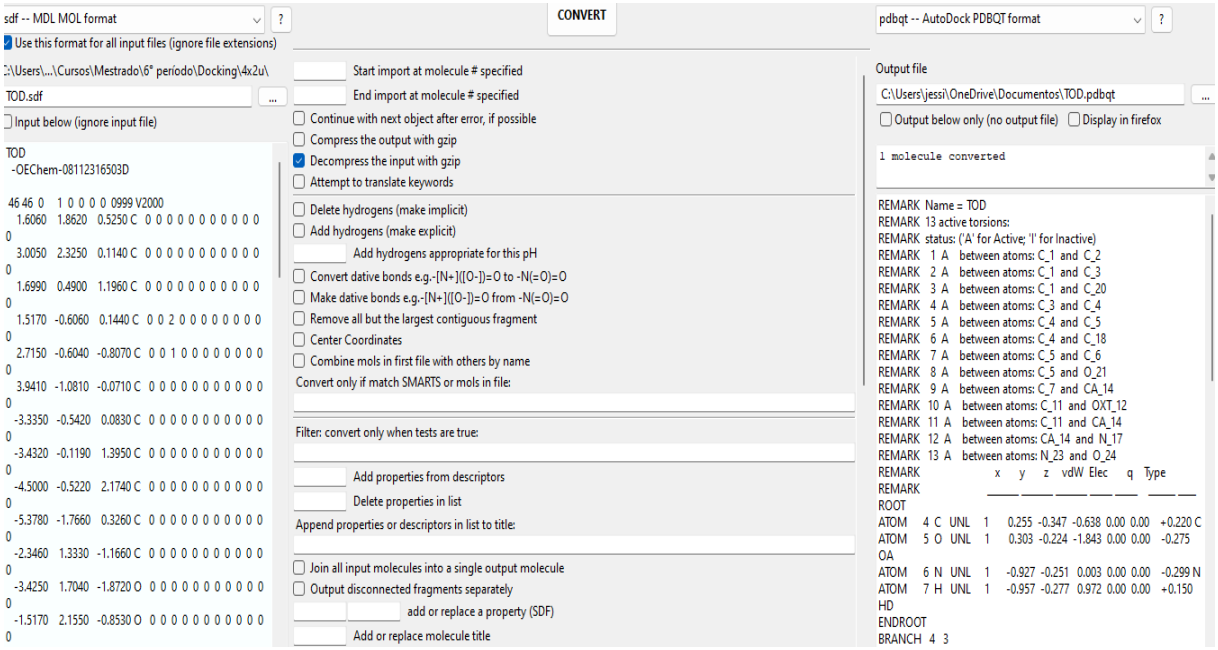
Código da Proteína	Organismo
4x2u	*Plasmodium falciparum*
4zw5	*Plasmodium falciparum*
4zw6	*Plasmodium falciparum*
4zw8	*Plasmodium falciparum*
4zx4	*Plasmodium falciparum*
4zx5	*Plasmodium falciparum*
4zx6	*Plasmodium falciparum*
4zw5	*Plasmodium falciparum*
5y1s	*Plasmodium falciparum*
5y3i	*Plasmodium falciparum*
6ea1	*Plasmodium falciparum*
6ea2	*Plasmodium falciparum*
6eaa	*Plasmodium falciparum*
6eab	*Plasmodium falciparum*
6ee3	*Plasmodium falciparum*
6ee4	*Plasmodium falciparum*
6ee6	*Plasmodium falciparum*
6eed	*Plasmodium falciparum*

Fonte: Autor.

4.3.1 *Open Babel*

O *Open Babel* é um *software* cuja finalidade é interconverter os arquivos das estruturas químicas em diversos formatos (O'BOYLE et al., 2011). Origina-se da versão do *OELib* lançada como *software Open source* pela *OpenEye Scientific* sob a *GPL* (Licença Pública *GNU*) (O'BOYLE et al., 2011). A Figura 38 apresenta uma conversão do arquivo de *.sdf* para *.pdbqt* do ligante de uma proteína.

Figura 38 – Exemplo de conversão de ligante em formato *.sdf* para *.pdbqt* no *Open Babel*.



Fonte: Autor.

4.3.2 *Discovery Studio*

O *Discovery Studio* é um *software* que permite realizar simulações computacionais de sistemas de pequenas moléculas e modelagem molecular, cuja finalidade é investigar as estruturas moleculares e seus complexos (BIOVIA, 2023).

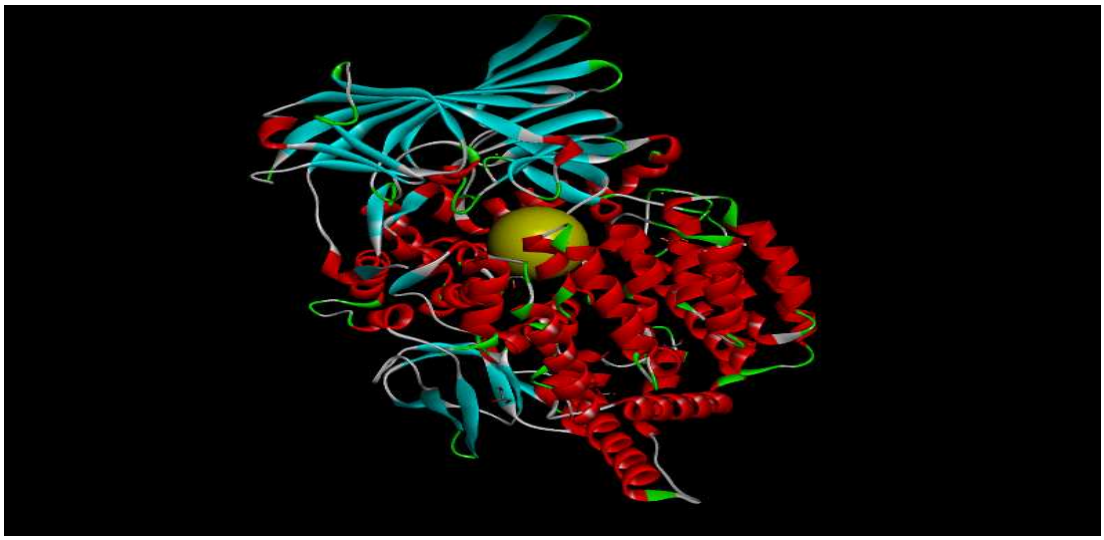
1. Na Figura 39, tem-se uma visão geral da proteína com as moléculas de água e íons removidos após o processo de preparação da proteínas.

Figura 39 – Visão geral da proteína no *Discovery Studio*.

Fonte: Autor.

2. Definição da região do ligante na proteína. O *Discovery Studio* possui um recurso que forma uma espécie de zona em volta do ligante, conforme mostrado na Figura 40.

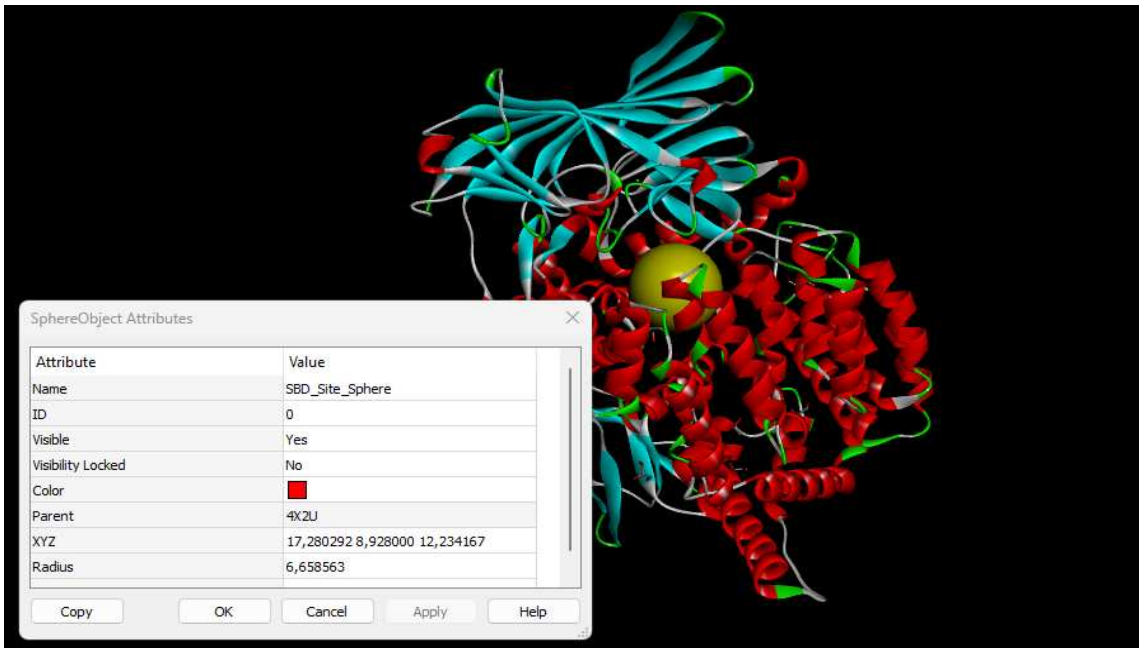
Figura 40 – Região do ligante (em amarelo) na proteína.



Fonte: Autor.

3. Após a definição da região do ligante, executou-se um comando para abrir uma janela que contém as informações de coordenadas do ligante, o que contribuiu para a formação da caixa no *AutoDock* e por fim realizar o *docking*, conforme na Figura 41.

Figura 41 – As coordenadas *XYZ* do ligante para formação da caixa.



Fonte: Autor.

4. Uma demonstração da estrutura do ligante correspondente à molécula gerada pelo *Discovery Studio*. Na Figura 42, representa a estrutura do ligante.

Figura 42 – Estrutura do ligante da molécula no *Discovery Studio*.

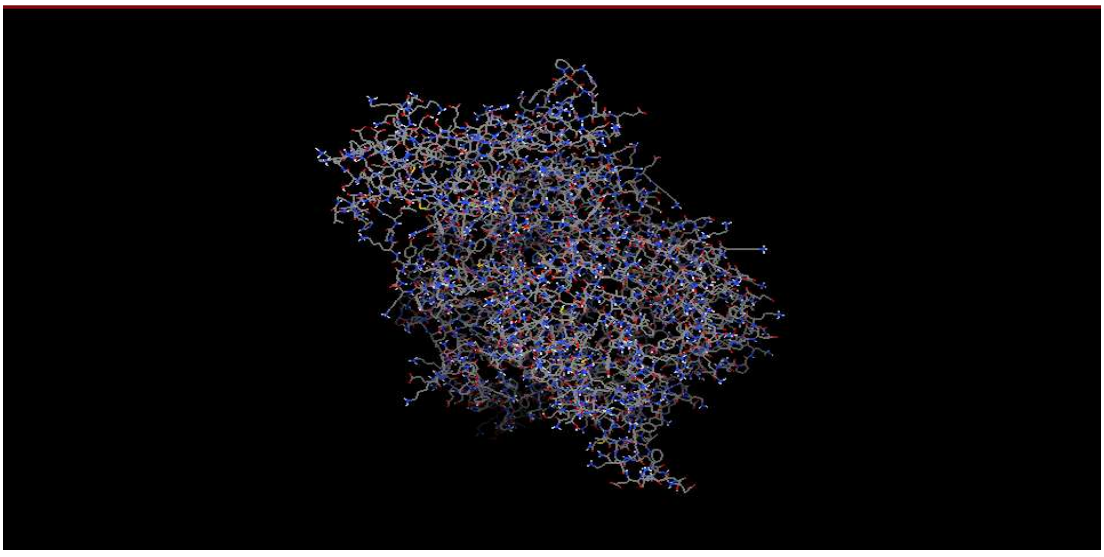
Fonte: Autor.

4.3.3 *Autodock*

O *Autodock* é um *software* de simulação e modelagem molecular, eficaz para prever as moléculas ou proteína (também chamado de alvos) de interesse, que se ligam a um receptor de estrutura *3D* (TROTT; OLSON, 2010). Além disso, também é capaz de realizar oacoplamento do ligante a uma proteína alvo.

1. Antes de iniciá-lo, é necessário preparar a molécula para evitar problemas durante o *docking*. Um desses preparos é verificar a existência de água na estrutura da molécula e, caso haja, tirá-la e adicionar hidrogênios, conforme mostrado na Figura 43.

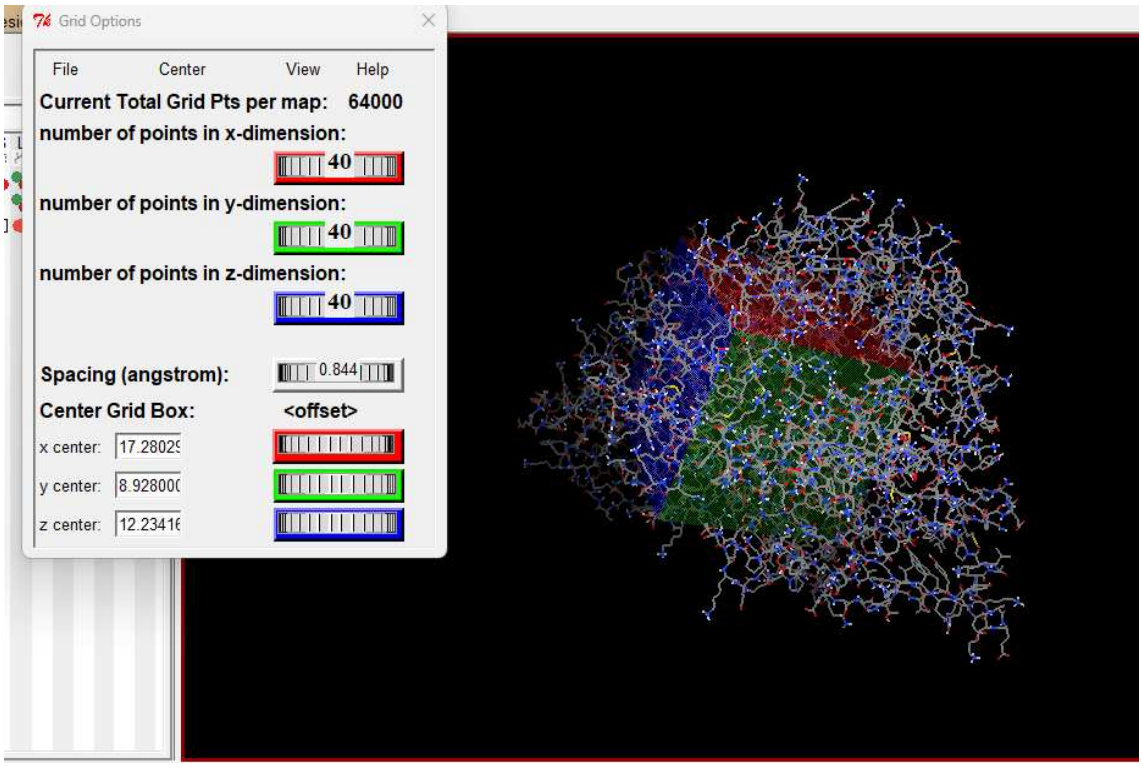
Figura 43 – Estrutura da proteína carregada no *Autodock*.



Fonte: Autor.

2. Criação da caixa foram definidas as coordenadas *XYZ* gerada no *Discovery Studio* para busca no *Autodock*. Aqui se demonstra como setar as coordenadas na caixa, conforme apresentado na Figura 44.

Figura 44 – Preparação da caixa de simulação no *Autodock*.



Fonte: Autor.

4.3.4 Resultados do *Docking*

O *Docking Molecular* é resultante da interação proteína-ligante, mas também pode ser constituído por aminoácidos. Neste trabalho, o objetivo foi esclarecer os resultados obtidos por meio de seu uso e entender suas interações. Também é importante ressaltar os códigos dos aminoácidos e seus respectivos significados nas interações obtidas nesse processo. A Tabela 11 apresenta as nomenclaturas dos aminoácidos.

Além disso, apresentar os tipos de interações provenientes do *docking* e suas respectivas caracteríticas para se entender, em detalhes, o que corresponde cada interação proteína-ligante presente.

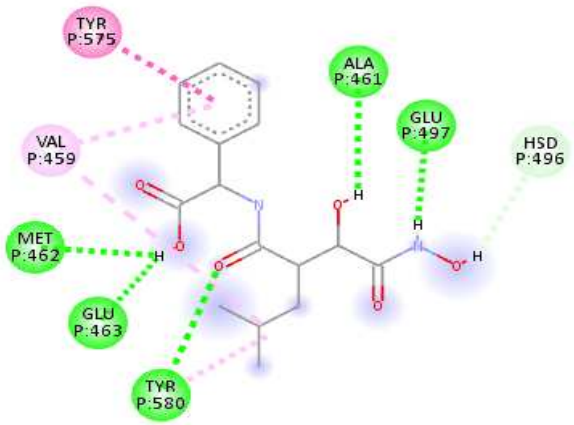
Tabela 11 – Nomenclaturas dos aminoácidos.

Aminoácido	Código
Glicina	*Gly*
Alanina	*Ala*
Leucina	*Leu*
Valina	*Val*
Isoleucina	*Ile*
Prolina	*Pro*
Fenilalanina	*Phe*
Serina	*Ser*
Treonina	*Thr*
Cisteína	*Cys*
Tirosina	*Tyr*
Asparagina	*Asn*
Glutamina	*Gln*
Aspartato	*Asp*
Glutamato	*Glu*
Arginina	*Arg*
Lisina	*Lys*
Histidina	*His*
Triptofano	*Trp*
Metionina	*Met*

Fonte: Autor.

Na Figura 45, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, π-*Donor Hydrogen Bond*, π-π *Stacked*, *Alkyl* e π-*Alkyl*.

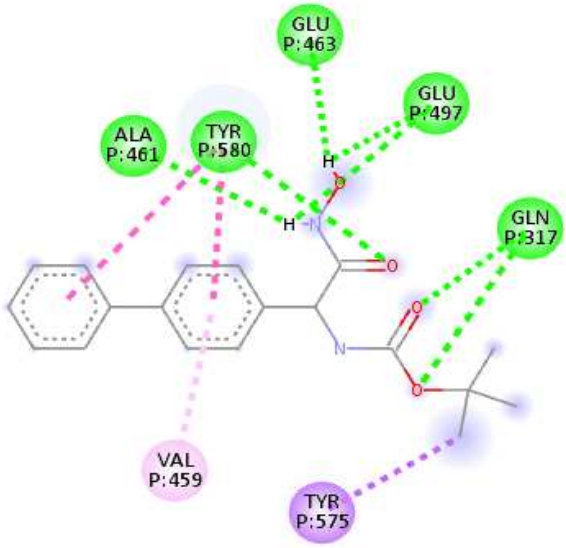
Figura 45 – Interações obtidas no processo de *docking* da proteína: *4x2u*.



Fonte: Autor.

Na Figura 46, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, π-*Sigma*, π-π *Stacked*, π-π *T shaped* e π-*Alkyl*.

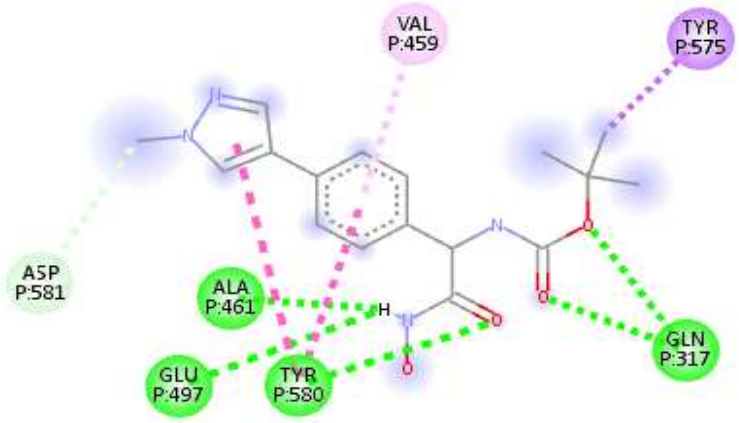
Figura 46 – Interações obtidas no processo de *docking* da proteína: *4zw5*.



Fonte: Autor.

Na Figura 47, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, π-*Sigma*, π-π *Stacked*, π-π *T shaped* e π-*Alkyl*.

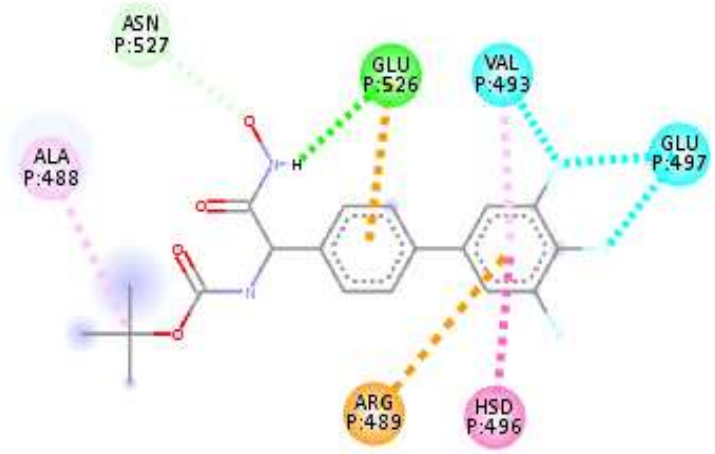
Figura 47 – Interações obtidas no processo de *docking* da proteína: *4zw6*.



Fonte: Autor.

Na Figura 48, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, *Halogen (Fluorine)*, π-*Cation*, π-*Anion*, π-π *Stacked*, *Alkyl* e π-*Alkyl*.

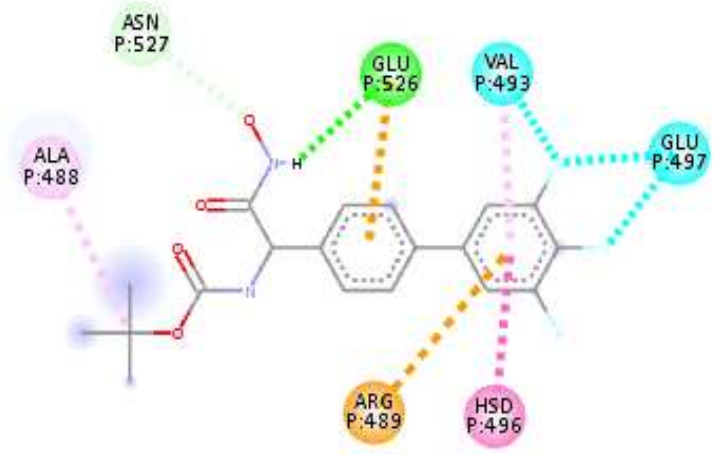
Figura 48 – Interações obtidas no processo de *docking* da proteína: *4zw7*.



Fonte: Autor.

Na Figura 49, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, π-*Anion*, π-*Sigma* e π-*Sulfur*.

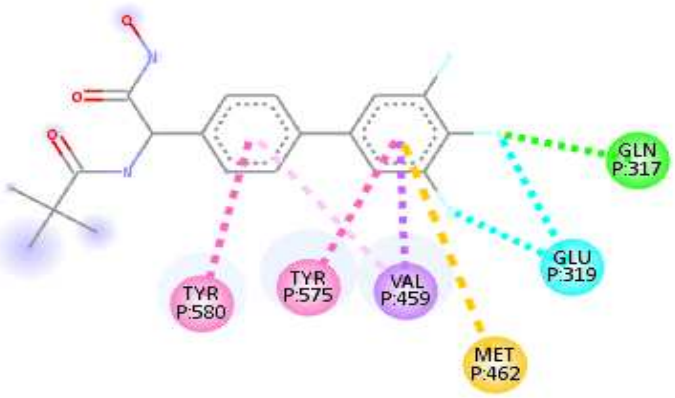
Figura 49 – Interações obtidas no processo de *docking* da proteína: *4zw8*.



Fonte: Autor.

Na Figura 50, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Halogen (Fluorine)*, π-*Sigma*, π-*Sulfur*, π-π *Stacked* e π-*Alkyl*.

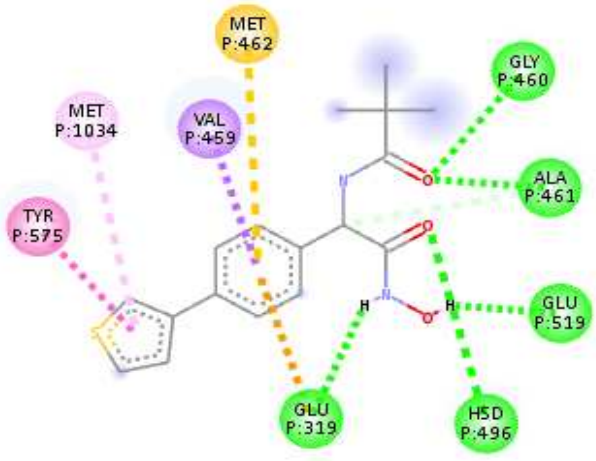
Figura 50 – Interações obtidas no processo de *docking* da proteína: *4zx4*.



Fonte: Autor.

Na Figura 51, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, π-*Anion*, π-*Sigma*,π-*Sulfur*, π-π *T shaped* e π-*Alkyl*.

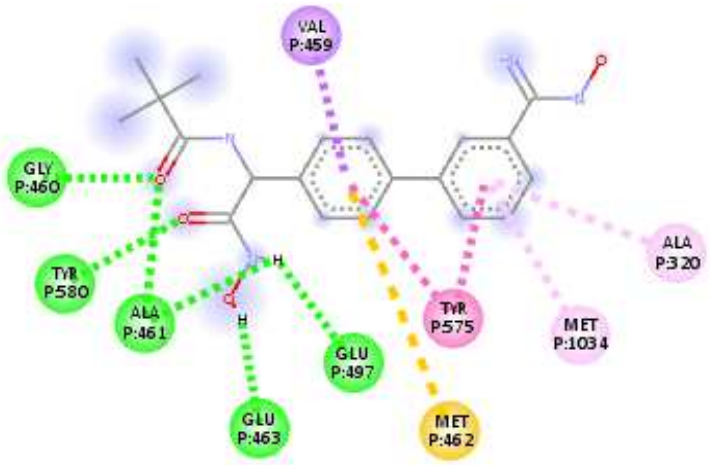
Figura 51 – Interações obtidas no processo de *docking* da proteína: *4zx5*.



Fonte: Autor.

Na Figura 52, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, π-*Sigma*, π-*Sulfur*, π-π *T shaped* e π-*Alkyl*.

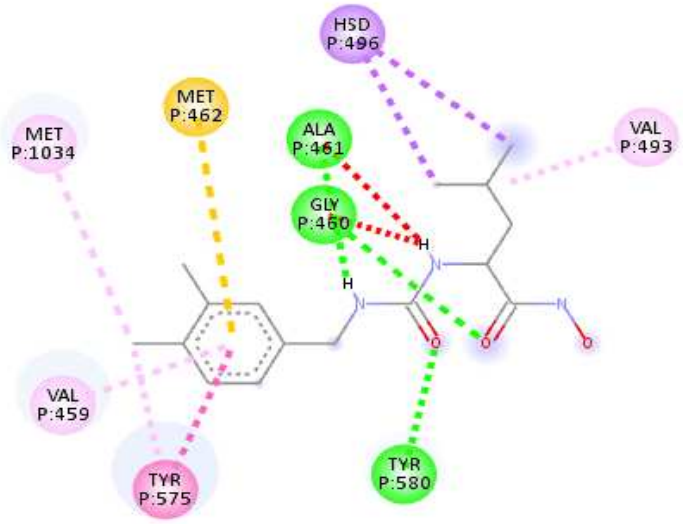
Figura 52 – Interações obtidas no processo de *docking* da proteína: *4zx6*.



Fonte: Autor.

Na Figura 53, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Unfavorable Donor-Donor*, π-*Sigma*, π-*Sulfur*, π-π *Stacked*, *Alkyl* e π-*Alkyl*.

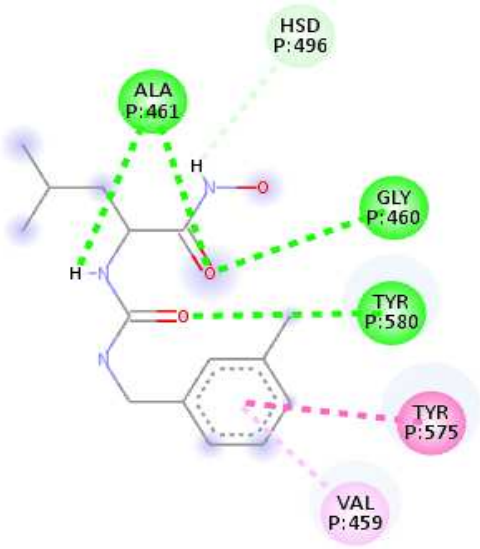
Figura 53 – Interações obtidas no processo de *docking* da proteína: *5y1s*.



Fonte: Autor.

Na Figura 54, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Unfavorable Donor-Donor*, π-*Donor Hydrogen Bond*, π-π *Stacked* e π-*Alkyl*.

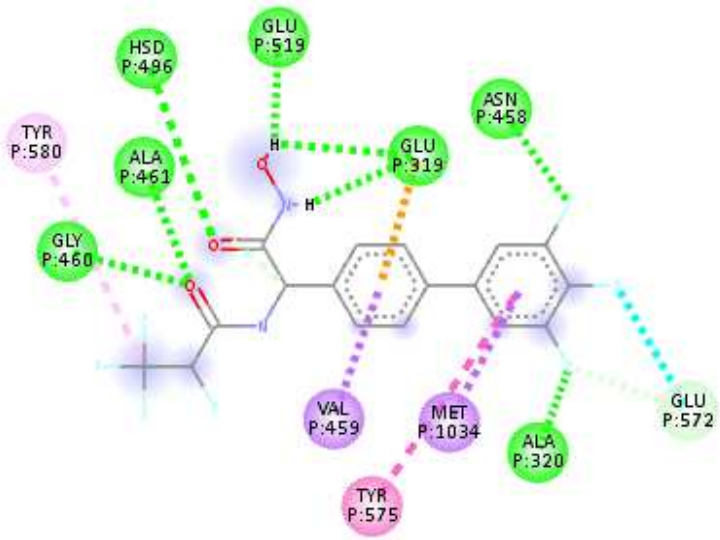
Figura 54 – Interações obtidas no processo de *docking* da proteína: *5y3i*.



Fonte: Autor.

Na Figura 55, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, *Halogen (Fluorine)*, π-*Anion*, π-*Sigma*, π-π *T shaped* e π-*Alkyl*.

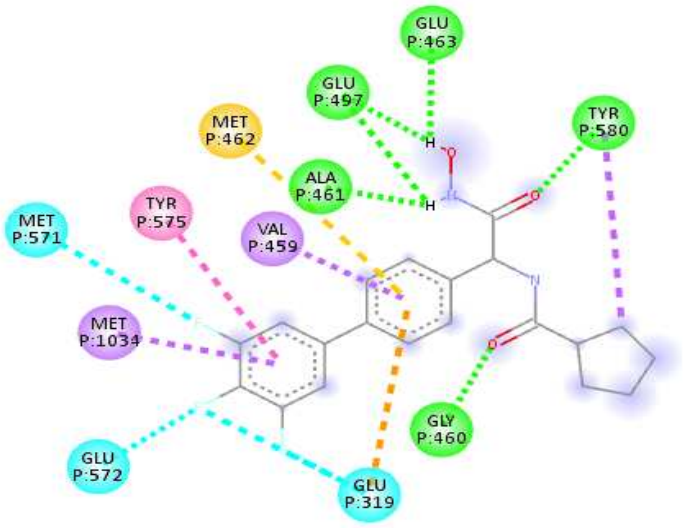
Figura 55 – Interações obtidas no processo de *docking* da proteína: *6ea1*.



Fonte: Autor.

Na Figura 56, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Halogen (Fluorine)*, π-*Anion*, π-*Sigma*, π-*Sulfur* e π-π *T shaped*.

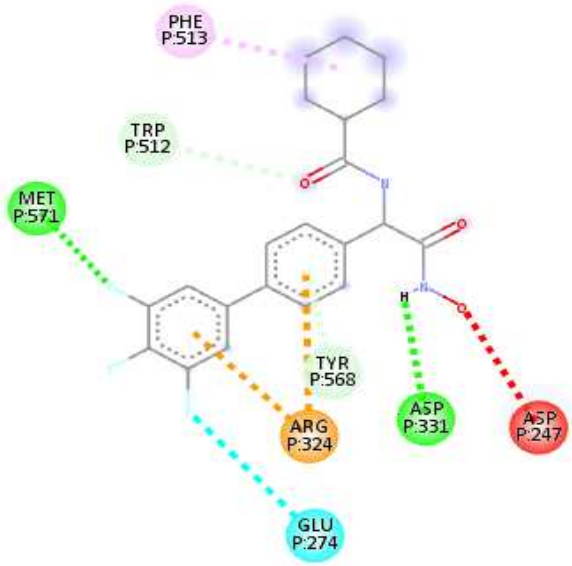
Figura 56 – Interações obtidas no processo de *docking* da proteína: *6ea2*.



Fonte: Autor.

Na Figura 57, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, *Halogen (Fluorine)*, *Unfavorable Donor-Donor*, π-*Cation*, π-*Donor Hydrogen Bond* e π-π *T shaped*.

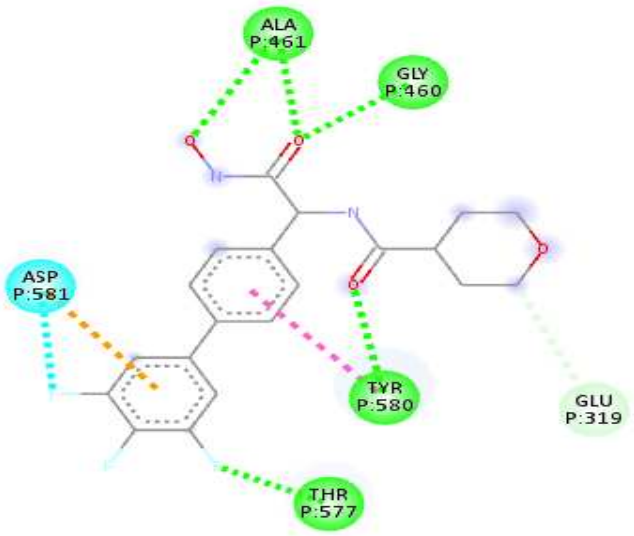
Figura 57 – Interações obtidas no processo de *docking* da proteína: *6eaa*.



Fonte: Autor.

Na Figura 58, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, *Halogen (Fluorine)*, π-*Anion* e π-π *Stacked*.

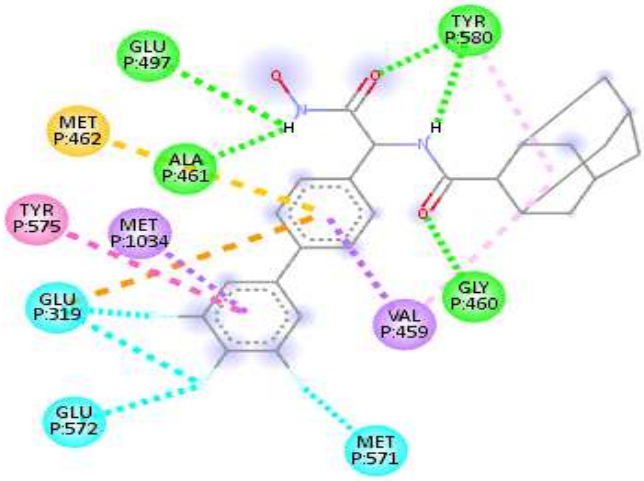
Figura 58 – Interações obtidas no processo de *docking* da proteína: *6eab*.



Fonte: Autor.

Na Figura 59, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Halogen (Fluorine)*, π-*Anion*, π-*Sigma*, π-*Sulfur*, π-π *T shaped*, *Alkyl* e π-*Alkyl*.

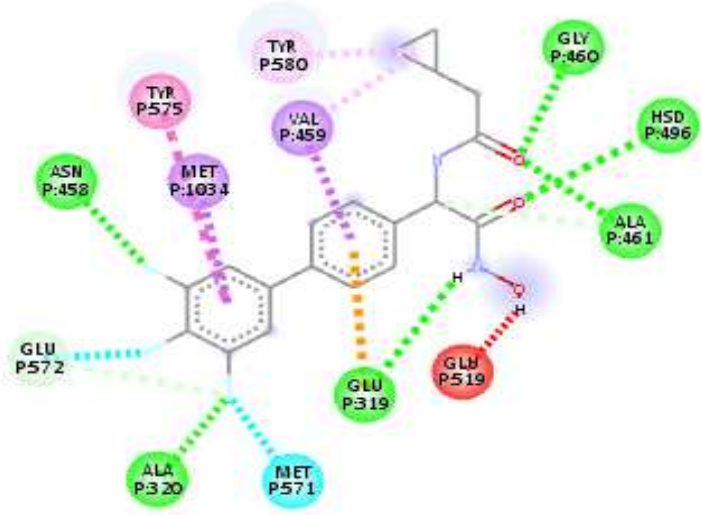
Figura 59 – Interações obtidas no processo de *docking* da proteína: *6ee3*.



Fonte: Autor.

Na Figura 60, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Unfavorable Donor-Donor*, *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, *Halogen (Fluorine)*, π-*Anion*, π-*Sigma*, π-*Sulfur*, π-π *T shaped*, *Alkyl* e π-*Alkyl*.

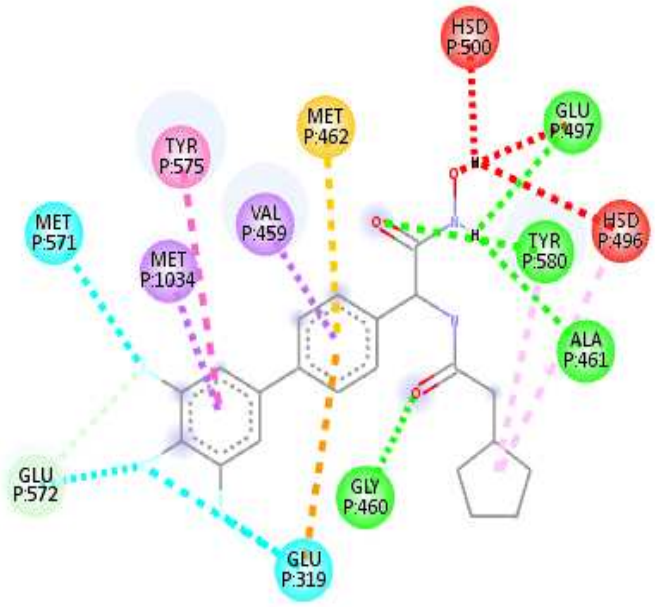
Figura 60 – Interações obtidas no processo de *docking* da proteína: *6ee4*.



Fonte: Autor.

Na Figura 61, encontram-se as interações resultantes do processo de *docking*, sendo elas:*Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, *Halogen (Fluorine)*, *Unfavorable Donor-Donor*, *Unfavorable Acceptor-Acceptor*, π*-Anion*, π*-Sigma*, π*-Sulfur*, π-π *T shaped* e π*-Alkyl*.

Figura 61 – Interações obtidas no processo de *docking* da proteína: *6ee6*.

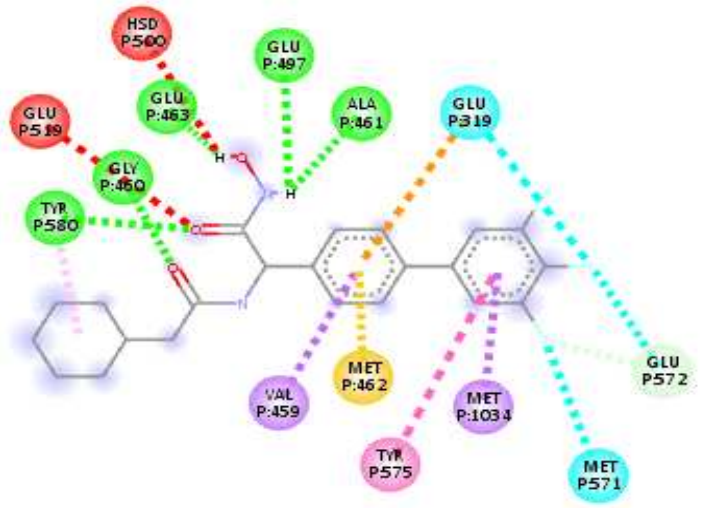


Fonte: Autor.

Na Figura 62, encontram-se as interações resultantes do processo de *docking*, sendo elas: *Conventional Hydrogen Bond*, *Carbon Hydrogen Bond*, *Halogen (Flu-*

orine), *Unfavorable Donor-Donor*, *Unfavorable Acceptor-Acceptor*, π*-Anion*, π*-Sigma*, π*-Sulfur*, π-π *T shaped* e π*-Alkyl*.

Figura 62 – Interações obtidas no processo de *docking* da proteína: 6eed.



Fonte: Autor.

4.3.4.1 Resultados das Interações proteína-ligante

As interações proteína-ligante encontradas no *docking* neste trabalho, são:

- **Interação Alkyl**: interação entre o grupo *Alkyl* liga-se a outros átomos ou a um grupo de átomos (GIESE; ALBRECHT, 2020);
- **Interação Carbon Hydrogen Bond**: interação covalente em que o átomo de carbono liga-se ao átomo de hidrogênio (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação Conventional Hydrogen Bond**: interação em que o átomo de hidrogênio liga-se ao átomo eletronegativo (como *N* (Nitrogênio) ou *O* (Oxigênio)) (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação Halogen (Fluorine)**: interação em que um átomo de halogênio é atraído por uma carga parcialmente negativa (LIN; MACKERELL, 2017);
- **Interação π-Alkyl**: interação não covalente entre o sistema π (rico em elétrons) que permite que os complexos orgânicos se liguem aos metais (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012; GIESE; ALBRECHT, 2020);

- **Interação π-Anion**: interação não covalente entre o sistema π (rico em elétrons) e um átomo, neste caso, o *Ânion* (carga negativa) (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação π-Cation**: interação não covalente entre o sistema π (rico em elétrons) e um átomo, neste caso, o *Cátion* (carga positiva) (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação π-Donor Hydrogen Bond**: interação não covalente entre o sistema π (rico em elétrons) e o átomo de hidrogênio (doador) (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação π-Sigma**: interação não covalente entre o sistema π (rico em elétrons) e metal com ligação *sigma* (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação π-π Stacked**: interação não covalente entre o sistema π (rico em elétrons) e e moléculas aromáticas (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação π-Sulfur**: interação não covalente entre o sistema π (rico em elétrons) e um metal, neste caso, enxofre (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação π-π T shaped**: interação não covalente entre o sistema π (rico em elétrons) e dois grupos aromáticos de moléculas (TSUZUKI; FUJII, 2008; MIROSLAV; BLUDSKý, 2012; TSUZUKI, 2012);
- **Interação Unfavorable Acceptor-Acceptor**: interação em que o aceitador-aceitador são desfavoráveis;
- **Interação Unfavorable Donor-Donor**: interação em que o doador-doador são desfavoráveis.

5 CONCLUSÃO

Este trabalho visou mostrar a importância das estratégias computacionais para caracterizar e prever os ligantes, utilizando-se do Aprendizado de Máquina. Buscou-se alinhar as ferramentas tradicionais com os recursos computacionais para alcançar resultados promissores em trabalhos com proteínas e com outros elementos, o que apresentou grande potencial para estudos futuros.

Para a realização deste, contou-se com os recursos tradicionais *BLAST* para alinhamento e análise das proteínas para construção da base de dados. Além disso, mas não menos importante o *BindingDB* foi crucial para se chegar à família da protease *Subtilisin-like serine protease (Plasmodium falciparum)* similar, que é à protease da lagarta.

Além disso, submeteu-se essa base de dados a diferentes contextos de algoritmos de Aprendizado de Máquina, compostos por características e parâmetros que podem influenciar os resultados de forma positiva ou negativa. Conseguiu-se também identificar qual apresenta melhor desempenho, considerando a influência das métricas de classificação, tais como: *Recall* e *Precision*. Os algoritmos que obtiveram melhor desempenho e destacaram foram: *LogisticRegression* e *MLPClassifier* com suas respectivas características.

Por fim, aplicou-se o *Docking Molecular*, uma técnica computacional capaz de prever as interação proteína-ligante e encontrar seus respectivos alvos. Com o auxílio do Aprendizado de Máquina pode-se aperfeiçoar e encontrar resultados significativos para realizar o *Docking* com mais eficácia. Os resultados obtidos mostraram que, recursos computacionais aliados às ferramentas tradionais no contexto de proteínas tendem a contribuir paraa redução de custo e de tempo, assim como melhorar e diminuir os processos que antes demandavam muito mais recursos para o desenvolvimento de trabalho como esta proposta de pesquisa.

Os avanços na Bioinformática tem contribuído para o aperfeiçoamento dos processos para descobertas de novas fórmulas, compostos e seus alvos (ou seja, as proteínas) em um período menor. Dessa forma, constatou-se que é possível conduzir experimentos com os mais diferentes tipos de compostos. Notou-se, ainda, que o *virtual screening* ou triagem virtual tem ganhado notoriedade.

Enfim, destaca-se que as áreas farmacológica e biomédica têm futuros promissores com as novas descobertas da ciência, o que pode ser revertido em benefícios para a sociedade. Este trabalho tem grande relevância para a Bioinformática e para a ciência como um todo e, a partir dele, novas pesquisas podem ser desenvolvidas e aplicadas para a descoberta e novos fármacos.

Referências

BAYAT, A. Science, medicine, and the future: Bioinformatics. **BMJ**, BMJ, v. 324, n. 7344, p. 1018–1022, abr. 2002. Disponível em: <https://doi.org/10.1136/bmj.324.7344.1018>.

BIOVIA, D. S. Dassault systemes biovia: Software discovery studio. 2023. Disponível em: <https://discover.3ds.com/discovery-studio-visualizer-download>.

CERA, E. D. Serine proteases. **IUBMB Life**, 2009.

CHEN, X.; LIN, Y.; GILSON, M. K. The binding database: overview and user's guide. **Biopolymers**, 2001.

DAVIES, B. J. et al. Serine proteases in rodent hippocampus. **J Biol Chem**, 1998.

DECOYS. Dud-e: A database of useful decoys: Enhanced. 2023. Disponível em: <https://dude.docking.org/>.

DRAGON, J. A. et al. Bioinformatics core survey highlights the challenges facing data analysis facilities. **Journal of Biomolecular Techniques : JBT**, Association of Biomolecular Resource Facilities, p. jbt.20–3102–005, jun. 2020. Disponível em: <https://doi.org/10.7171/jbt.20-3102-005>.

DRUGBANK. Drugbank: Database for drug and drug target info. 2023. Disponível em: <https://go.drugbank.com/>.

DUCK, G. et al. A survey of bioinformatics database and software usage through mining the literature. **PLOS ONE**, Public Library of Science (PLoS), v. 11, n. 6, p. e0157989, jun. 2016. Disponível em: <https://doi.org/10.1371/journal.pone.0157989>.

FURTADO, M. I. V. **Redes neurais artificiais : uma abordagem para sala de aula**. Ponta Grossa(PR): Atena Editora: [s.n.], 2019.

FáVERO, L. P. L. et al. **Análise de dados: modelagem multivariada para tomada de decisões**. Rio de Janeiro: Elsevier: [s.n.], 2009.

GAUTHIER, J. et al. A brief history of bioinformatics. **Briefings in Bioinformatics**, v. 20, n. 6, p. 1981–1996, 08 2018. ISSN 1477-4054. Disponível em: <https://doi.org/10.1093/bib/bby063>.

GIESE, M.; ALBRECHT, M. Alkyl-alkyl interactions in the periphery of supramolecular entities: From the evaluation of weak forces to applications. **Chempluschem**, 2020.

GéRON, A. **Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. O'Reilly Media, Incorporated: [s.n.], 2019.

JIN, X. et al. Db-explorer: a web-based interactive map of the protein data bank in shape space. **BMC Bioinformatics**, 2015. Disponível em: <https://www.rdkit.org>.

KOISTINEN, H.; KOISTINEN, R.; ZHANG, W. M. Nexin-1 inhibits the activity of human brain trypsin. **Neuroscience**, 2009.

KUBAT, M. **An Introduction to Machine Learning**. Springer International Publishing: [s.n.], 2017.

LIN, F. Y.; MACKERELL, A. Do halogen–hydrogen bond donor interactions dominate the favorable contribution of halogens to ligand–protein binding? **The Journal of Physical Chemistry B**, 2017.

LUO, W.; WANG, Y.; REISER, G. Protease-activated receptors in the brain: Receptor expression, activation, and functions in neurodegeneration and neuroprotection. **Brain Res Rev**, 2007.

MARQUES, M. R. F. **Bioquímica**. Florianópolis : BIOLOGIA/EAD/UFSC: 1. ed revisada. 2a. ed., 2014.

MIROSLAV, R.; BLUDSKý, O. Intermolecular π–π interactions in solids. **Phys. Chem. Chem.Phys.**, 2012. Disponível em: <http://dx.doi.org/10.1039/B718656H>.

MULLER, A.; GUIDO, S. **Introduction to Machine Learning with Python**. O'Reilly Media, Incorporated: [s.n.], 2017.

NELSON, D. L.; COX, M. M. **Lehninger Princípios de Bioquímica**. São Paulo: Sarvier: 3. ed., 2002.

NELSON, D. L.; COX, M. M. **Princípios de Bioquímica de Lehninger**. Porto Alegre: Artmed: 6. ed., 2014.

O'BOYLE, N. M. et al. Open babel: An open chemical toolbox. **Journal of Cheminformatics**, 2011. Disponível em: <https://doi.org/10.1186/1758-2946-3-33>.

PDB. Pdb: Protein data bank. 2023. Disponível em: <https://www.rcsb.org/>.

PYMOL. Pymol. 2023. Disponível em: <https://pymol.org/2/>.

QUEIROZ, F. C. et al. ppigremlin: a graph mining based detection of conserved structural arrangements in protein-protein interfaces. **BMC Bioinformatics**, p. 15–21, 2020. Disponível em: <https://doi.org/10.1186/s12859-020-3474-1>.

RDKIT. Rdkit: Software de cheminformática de código aberto. 2023. Disponível em: <https://www.rdkit.org>.

RIBEIRO, V. S. et al. visgremlin: graph mining-based detection and visualization of conserved motifs at 3d protein-ligand interface at the atomic level. **BMC Bioinformatics**, n. 80, 2020. Disponível em: <https://doi.org/10.1186/s12859-020-3347-7>.

SANTANA, C. A. et al. Gremlin: A graph mining strategy to infer protein-ligand interaction patterns. **2016 IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE)**, p. 28–35, 2016.

SCIKIT. Scikit learn: Glossary of common terms and api elements. 2023. Disponível em: <https://scikit-learn.org/stable/glossary.html#>.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais para engenharia e ciências aplicadas**. São Paulo: Artliber: 1. ed, 2010.

SILVA, L. A.; PERES, S. M.; BOSCARIOLI, C. **Introdução à mineração de dados: com aplicações em R**. Rio de Janeiro: Elsevier: 1. ed, 2016.

SZMOLA, R.; KUKOR, Z.; SAHIN-TOTH, M. Human mesotrypsin is a unique digestive protease specialized for the degradation of trypsin inhibitors. **J Biol Chem**, 2003.

TRIPATHI, L. P.; SOWDHAMINI, R. Cross genome comparisons of serine proteases in arabidopsis and rice. **BMC Genomics**, 2006. Disponível em: <https://doi.org/10.1186/1471-2164-7-200>.

TROTT, O.; OLSON, A. J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. **J Comput Chem**, 2010.

TSUZUKI, S. Ch/π interactions. **Annu. Rep. Prog. Chem., Sect. C: Phys. Chem**, 2012. Disponível em: <http://dx.doi.org/10.1039/B718656H>.

TSUZUKI, S.; FUJII, A. Nature and physical origin of ch/π interaction: significant difference from conventional hydrogen bonds. **Phys. Chem. Chem. Phys.**, 2008. Disponível em: <http://dx.doi.org/10.1039/B718656H>.

VERLI, H. **Bioinformática da Biologia à flexibilidade molecular**. Porto Alegre: 1. ed., 2014.

WANG, Y.; LUO, W.; REISER, G. Trypsin and trypsin-like proteases in the brain: proteolysis and cellular functions. **Cell. Mol. Life Sci**, 2008.