Trabalho de Conclusão de Curso

Caracterização de técnicas de aprendizado de máquina para classificação de reatividade entre compostos e proteínas

Danilo Vasconcelos Freire

orientado por
Prof. Dr. André Luiz Lins de Aquino

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
5 de novembro de 2024

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

CARACTERIZAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE REATIVIDADE ENTRE COMPOSTOS E PROTEÍNAS

Trabalho de Conclusão de Curso apresentado ao Instituto de Computação da Universidade Federal de Alagoas como requisito parcial para a obtenção do grau de Bacharel em ciência da Computação.

Danilo Vasconcelos Freire

Orientador: Prof. Dr. André Luiz Lins de Aquino

Banca Examinadora:

André Luiz Lins de Aquino	Prof. Dr., IC-UFAL
Amanda Lima Cunha	MSC, IQB-UFAL
Gean da Silva Santos	MsC, IC-UFAL

Maceió, Alagoas
5 de novembro de 2024

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Helena Cristina Pimentel do Vale – CRB4 - 661

F866c Freire, Danilo Vasconcelos.
Caracterização de técnicas de aprendizado de máquina para classificação de reatividade entre compostos e proteínas / Danilo Vasconcelos Freire. – 2024.
38 f.: il.

Orientador: André Luiz Lins de Aquino.
Monografia (Trabalho de Conclusão de Curso em Ciência da Computação) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2024.

Bibliografia: f. 36-38.

1. Classificação reativa. 2. Vetores de representação proteica. 3. Interações droga-alvo. 4. Aprendizado de máquina. I. Título.

CDU: 004.94

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Instituto de Computação – IC
Campus A. C. Simões - Av. Lourival de Melo Mota, BL 12
Tabuleiro do Martins, Maceió/AL - CEP: 57.072-970
Telefone: (082) 3214-1401

Trabalho de Conclusão de Curso - TCC

Formulário de Avaliação

Nome do Aluno
Danilo Vasconcelos Freire

Nº de Matrícula
19110977

Título do TCC (Tema)
Caracterização de técnicas de aprendizado de máquina para classificação de reatividade entre compostos e proteínas

Banca Examinadora

André Luiz Lins de Aquino
Nome do Orientador

ANDRE LUIZ LINS DE AQUINO:03235015400
Assinado de forma digital por ANDRE LUIZ LINS DE AQUINO:03235015400
Dados: 2024.12.06 14:19:04 -03'00'

Amanda Lima Cunha
Nome do Professor

Documento assinado digitalmente
gov.br AMANDA LIMA CUNHA
Data: 06/12/2024 14:21:44-0300
Verifique em https://validar.iti.gov.br

Assinatura

Gean da Silva Santos
Nome do Professor

Documento assinado digitalmente
gov.br GEAN DA SILVA SANTOS
Data: 06/12/2024 16:50:56-0300
Verifique em https://validar.iti.gov.br

Assinatura

Data da Defesa
06/12/2024

Nota Obtida
9,50 (Nove e meio)

Observações

Coordenador do Curso
De Acordo

Documento assinado digitalmente
gov.br MARIA CRISTINA TENORIO CAVALCANTE ESCAR
Data: 07/12/2024 10:36:52-0300
Verifique em https://validar.iti.gov.br

Assinatura

Dedicatória

Dedico este trabalho aos meus pais Fernanda e Wlademy.

Agradecimentos

Gostaria de agradecer UFAL e a todos os professores que fizeram parte dessa jornada, em especial ao professor André Aquino e ao LaCCAN/ORION por terem me dado tantas oportunidades e guiado nessa caminho, agradeço também ao pessoal do grupo de pesquisa Elias, Rodrigo e Anderson pelas discussões e aprendizados em conjunto. Gostaria de agradecer a minha namorada Ullyanne por ter me incentivado a continuar sempre e me apoiado nos momentos mais difíceis. Agradeço aos meus pais Wlademy e Fernanda, por sempre terem acreditado em mim e nunca terem me deixado desistir, agradeço aos meus avós Pedro, Terezinha, Olindina e Severino por terem sido de grande influência na minha educação e a toda minha família pelo apoio em tantos aspectos. Agradeço a todos os amigos que fiz no caminho no laboratório e na UFAL, nominalmente Jordan, Matheus Levi, Ascânio, João Ayalla, Gabriel Souza, Jorge, Carlos Eduardo, Rafael e tantos outros que tornaram esse processo mais leve. Agradeço aos meus amigos de colégio que sempre me ajudaram a manter a pouca sanidade que ainda me resta, Arthur Cerqueira, José Badú, Thiago de Menezes, Alice Suzuki e Antonio Bernardo que não está mais entre nós mas foi crucial na minha jornada.

Maceió, 1 de dezembro de 2024.

Resumo

Este trabalho teve como objetivo investigar a previsão binária de reatividade entre compostos químicos e proteínas, utilizando a abordagem de Interação Droga-Alvo (DTI) por meio de diferentes modelos de aprendizado de máquina. A metodologia envolveu a segmentação de índices de reatividade catalogados em grupos de reagentes e não reagentes, a construção de bases de dados para cada composto-alvo e a extração de vetores de características proteicas utilizando diversos padrões, como APAAC, CTDD, CTRiad, DDE, Geary, K-SEP PSSM, PFAM, QSO e SPMAP.

Os dados foram coletados a partir da base ChEMBL, e a representação proteica foi obtida da base UniProt. A classificação binária foi realizada separando os valores de reatividade em reagentes e não reagentes, com a conversão em 1 e 0, respectivamente. Para a análise, foram aplicados modelos de classificação, incluindo Random Forest, KNN, SVM, MLP, XGBoost e Dummy Classifier, com otimização de hiperparâmetros utilizando o algoritmo Grid Search.

Os resultados mostraram que, embora nenhum modelo tenha se destacado consistentemente em desempenho, alguns pares de modelo-característica apresentaram resultados promissores. O XGBoost com o vetor CTriad obteve uma média de acurácia de 72,78%, enquanto o DDE com Random Forest alcançou 71,08%. O modelo MLP, por outro lado, apresentou desempenho abaixo da média, possivelmente devido à escassez de dados na base.

A análise detalhada dos resultados revelou que o XGBoost e o Random Forest se destacaram em termos de performance, especialmente em conjuntos de dados com menor quantidade de amostras, como o composto ChEMBL104. O KNN e o K-SEP PSSM também mostraram resultados acima da média, enquanto o SVM, utilizando o vetor Geary, apresentou uma das melhores performances em muitos compostos, embora tenha enfrentado limitações em alguns casos.

Esses achados sugerem que a escolha do modelo e do vetor de características é crucial para a eficácia da predição de reatividade entre compostos e proteínas. A pesquisa destaca a importância de uma investigação mais aprofundada sobre as interações entre os modelos e as características, bem como a necessidade de um maior volume de dados para melhorar a generalização e a consistência dos modelos mais complexos.

***Palavras-chave*: classificação, Interações Droga-Alvo, Aprendizado de Máquina, vetores de representação proteica.**

Abstract

This work aimed to investigate the binary prediction of reactivity between chemical compounds and proteins, using the Drug-Target Interaction (DTI) approach through different machine learning models. The methodology involved segmenting cataloged reactivity indices into groups of reactive and non-reactive compounds, constructing databases for each target compound, and extracting protein feature vectors using various patterns, such as APAAC, CTDD, CTRiad, DDE, Geary, K-SEP PSSM, PFAM, QSO, and SPMAP.

The data were collected from the ChEMBL database, and the protein representation was obtained from the UniProt database. The binary classification was performed by separating the reactivity values into reactive and non-reactive, converting them to 1 and 0, respectively. For the analysis, classification models were applied, including Random Forest, KNN, SVM, MLP, XGBoost, and Dummy Classifier, with hyperparameter optimization using the Grid Search algorithm.

The results showed that, although no model consistently outperformed the others, some model-feature pairs presented promising results. XGBoost with the CTriad vector achieved an average accuracy of 72.78%, while DDE with Random Forest reached 71.08%. The MLP model, on the other hand, exhibited below-average performance, possibly due to the scarcity of data in the dataset.

A detailed analysis of the results revealed that XGBoost and Random Forest excelled in terms of performance, especially in datasets with a smaller number of samples, such as the ChEMBL104 compound. KNN and K-SEP PSSM also showed above-average results, while SVM, using the Geary vector, demonstrated one of the best performances across many compounds, although it faced limitations in some cases.

These findings suggest that the choice of model and feature vector is crucial for the effectiveness of predicting reactivity between compounds and proteins. The research highlights the importance of further investigation into the interactions between models and features, as well as the need for a larger volume of data to improve the generalization and consistency of more complex models.

***Keywords*: *classification*, *Drug-Target Interactions Machine Learning*, *protein feature vectors*.**

Lista de Figuras

Lista de Abreviações

PCM *Proteochemometric modeling*

APAAC *Amphiphilic pseudo amino acid composition*

CTDD *Composition transition distribution distribution*

CTRIAD *Conjoint triad*

CTF *Conjoint triad feature*

DDE *Dipeptide deviation from expected mean*

K-Sep_PSSM *k separated bigrams on Position Specific Scoring Matrix*

QSO *quasi-sequence order*

SPMap *subsequence profile map*

TAAP *total amino acid properties*

KNN *K nearest neighbors*

MLP *Multi layer perceptron*

ReLU *Rectified Linear Unit*

Sumário

Capítulo 1

Introdução

1.1 Contextualização e motivação

A descoberta de medicamentos é um processo químico complexo e dispendioso, tanto em tempo quanto em recursos financeiros. O financiamento para a realização de um grande número de experimentos laboratoriais muitas vezes se mostra inviável. Nesse cenário, aumentar as chances de sucesso na identificação de compostos, seja para reações ou inibições, torna-se crucial. Segundo Ou-Yang et al. (2012), o ciclo típico de descoberta e desenvolvimento de medicamentos, desde a concepção até a comercialização, leva cerca de 14 anos, com custos entre 0,8 e 1,0 bilhão de dólares. Apesar do aumento dos investimentos nas últimas décadas, a produção de novos medicamentos não tem acompanhado esse crescimento devido à baixa eficiência e à alta taxa de falhas no processo. Atualmente, há diversos métodos de simulação que consideram as características tridimensionais das proteínas e compostos. Contudo, esses métodos também exigem muitos recursos e tempo, o que reforça a necessidade de explorar abordagens complementares. Uma dessas estratégias é a descoberta de fármacos baseada em vetores de características proteicas (DTI), que busca otimizar a identificação de interações entre compostos e alvos biológicos.

Nesse contexto, o uso de métodos computacionais para prever interações entre drogas e alvos biológicos (Drug-Target Interaction, ou DTI) tem ganhado destaque. Esses métodos combinam representações vetoriais de características proteicas com técnicas de aprendizado de máquina, oferecendo uma alternativa eficiente e robusta às abordagens experimentais tradicionais. A aplicação de modelos preditivos com vetores de características proteicas não só promete reduzir custos e acelerar o processo de descoberta, mas também oferece uma abordagem sistemática para identificar interações potenciais com alta precisão.

Tradicionalmente, a descoberta de fármacos é realizada por meio de experimentos que testam reações entre compostos e proteínas, um processo caro e demorado. O uso de vetores de características proteicas permite representar de forma eficiente informa-

ções relevantes sobre as proteínas, facilitando a previsão de interações. Essa abordagem, combinada com modelos de inteligência artificial, pode não apenas reduzir custos, mas também aumentar a eficácia dos experimentos, representando uma alternativa viável aos métodos convencionais.

Este estudo propõe investigar o impacto de diferentes representações proteicas e modelos de classificação na previsão de reatividade entre compostos e proteínas. Explorando várias abordagens de aprendizado supervisionado, o trabalho busca responder a questões essenciais sobre a eficácia desses métodos na identificação de compostos bioativos. A pesquisa avalia o desempenho preditivo de modelos como Random Forest, SVM, KNN e XGBoost, aplicados a representações proteicas como APAAC, CTriad e DDE, com o objetivo de identificar combinações ideais para maximizar a precisão e outras métricas de desempenho.

1.2 Objetivos

1.2.1 Objetivo geral

O presente Trabalho tem como objetivo caracterizar técnicas de aprendizagem de máquina aplicando diversos modelos para realizar classificações de reatividade em compostos utilizando vetores de características como entrada, trazendo avaliações para os resultados dos experimentos.

1.2.2 Objetivos específicos

Para alcançar o objetivo geral deste trabalho, definiram-se os seguintes objetivos específicos:

- Verificar possíveis tendências de representações proteicas com maior potencial de predição
- Verificar modelos com maior performance preditiva

Capítulo 2

Fundamentação Teórica

2.1 Representação proteica vetorial

Representações proteicas vetoriais são uma forma de representar a estrutura tridimensional de proteínas de maneira simplificada, utilizando vetores para descrever a posição e a orientação dos átomos ou grupos de átomos. Essas representações são úteis para visualizar e analisar a conformação das proteínas, suas interações e funções biológicas. No contexto atual será utilizada a representação FASTA por se tratar de uma representação econômica de espaço em memória e amplamente utilizada, facilitando a compatibilidade entre os diversos softwares de extração de características que serão detalhados mais a frente.

FASTA A representação proteica vetorial FASTA desenvolvida por Lipman and Pearson (1985) é um formato amplamente utilizado em bioinformática para a representação de sequências de aminoácidos, permitindo a manipulação e análise eficiente de dados biológicos. Neste formato, cada sequência é precedida por uma linha de descrição que começa com o caractere >, seguida pela sequência de aminoácidos, onde cada aminoácido é representado por um código de uma única letra, tendo um tamanho entre 80 a 135 caracteres. A simplicidade do formato FASTA facilita a sua utilização em diversas ferramentas de processamento de texto e linguagens de script, tornando-o um padrão quase universal na área. Originalmente desenvolvido a partir do pacote de software FASTA, o formato permite que sequências sejam organizadas de maneira clara, com a possibilidade de incluir comentários e identificadores únicos, o que é essencial para a rastreabilidade e a integração de dados provenientes de diferentes bancos de dados biológicos. Sendo representação principal no banco de dados *Uniprot* de Consortium (2020) foi a representação proteica utilizada para a criação da base de dados.

2.2 Vetores de características proteicas

Os Vetores de características proteicas têm como objetivo extrair numericamente características de proteínas para aplicações de análises estatísticas e de inteligência artificial com objetivo principal evidenciar similaridades relevantes para identificações. Tem como objetivo também, para o contexto de classificação supervisionada, a constância das dimensões dos dados permitindo a aplicação de modelos de aprendizagem de máquina assim como reduzindo seu custo de armazenamento. Para esse estudo foram selecionados 9 descritores proteicos clássicos, por serem os mais relevantes e citados, para uma caracterização de aprendizagem de máquina, sendo eles: APAAC, CTDD, CTRiad, DDE, Geary, K-SEP PSSM, PFAM, QSO, SPMAP apresentados de forma mais detalhada a seguir.

APAAC O APAAC é um vetor de características que representa a composição de aminoácidos em sequências de proteínas, preservando o efeito da ordem dos resíduos por meio de fatores sequenciais. Esses fatores são derivados de funções de correlação que analisam os índices de hidrofobicidade e hidrofilia dos aminoácidos.

Hidrofobicidade refere-se à tendência de uma substância ou parte de uma molécula em evitar a interação com a água. Aminoácidos hidrofóbicos possuem cadeias laterais que não se ligam bem à água, o que os faz se agrupar em ambientes aquosos, geralmente no interior das proteínas, contribuindo para a estabilidade estrutural. Por outro lado, hidrofilia é a capacidade de uma substância ou parte de uma molécula de interagir favoravelmente com a água. Aminoácidos hidrofílicos possuem cadeias laterais que podem formar ligações de hidrogênio com moléculas de água, tornando-os mais solúveis em ambientes aquosos e frequentemente localizados na superfície das proteínas, onde podem interagir com o meio circundante.

Dessa forma, o APAAC mantém a distribuição de aminoácidos anfifílicos ao longo da cadeia proteica, que são aqueles que apresentam características tanto hidrofóbicas quanto hidrofílicas. Proposta por Chou (2005), essa abordagem tem se mostrado eficaz na previsão de classes de subfamílias de enzimas, gerando uma representação fixa de 80 valores discretos que refletem as propriedades de hidrofobicidade e hidrofilia. Essa representação permite uma análise mais especifica das interações e funções das proteínas em diferentes contextos biológicos por modelos computacionais.

CTDD O vetor de características proteicas CTDD, desenvolvido por Dubchak (1999) para a tarefa de reconhecimento de dobramento de proteínas, fornece padrões de distribuição de aminoácidos com base na classe a que pertencem. Ele utiliza sete tipos de propriedades físico-químicas, incluindo hidrofobicidade, volume, polaridade, polarizabilidade, carga, estruturas secundárias e acessibilidade ao solvente. Cada propriedade é calculada de 20 em 20 aminoácidos e é dividida em três classes: composição, transição e distribuição. Os padrões de distribuição são determinados de

acordo com cinco posições diferentes (resíduos) para a classe correspondente, que são o primeiro resíduo e os resíduos localizados exatamente em 25%, 50%, 75% e 100% da sequência. Essas posições são divididas pelo comprimento da sequência total da proteína para o cálculo das frações de cada classe.

A composição refere-se à porcentagem global de diferentes grupos de aminoácidos na proteína. Para gerá-la os aminoácidos são agrupados nas três categorias de propriedade físico-químicas, como hidrofóbicos, polares e neutros. Posteriormente é contabilizada a porcentagem de aminoácidos em cada uma das categorias gerando um vetor com três valores

A transição refere-se à frequência de mudanças entre os grupos de aminoácidos ao longo da sequência. Para gerá-la calcula-se a frequência com que um aminoácido de um grupo é seguido por um aminoácido de outro grupo, o vetor de transição contém três valores, representando as transições entre os grupos.

A distribuição refere-se ao padrão de localização dos grupos de aminoácidos na sequência. Para gerá-la calcula-se a fração da sequência onde os primeiros aminoácidos de cada grupo aparecem, além das localizações dos percentuais de 25%, 50%, 75% e 100%, o vetor de distribuição contém cinco valores para cada grupo, totalizando 15 valores.

CTriad CTriad, também conhecido como CTF, proposto por Shen J (2007) é baseado na frequência de combinações de três aminoácidos em uma sequência de proteínas, onde os aminoácidos são primeiramente convertidos em um alfabeto reduzido de 7 caracteres mapeado a partir desse conjunto: A,G,V, I,L,F,P, Y,M,T,S, H,N,Q,W, R,K, D,E e C, e depois a sequencia proteica de alfabeto reduzido tem sua frequência de combinação de tríades distintas contadas, sendo um total de 7 x 7 x 7, 343 posições. Como descrito na imagem 2.1.

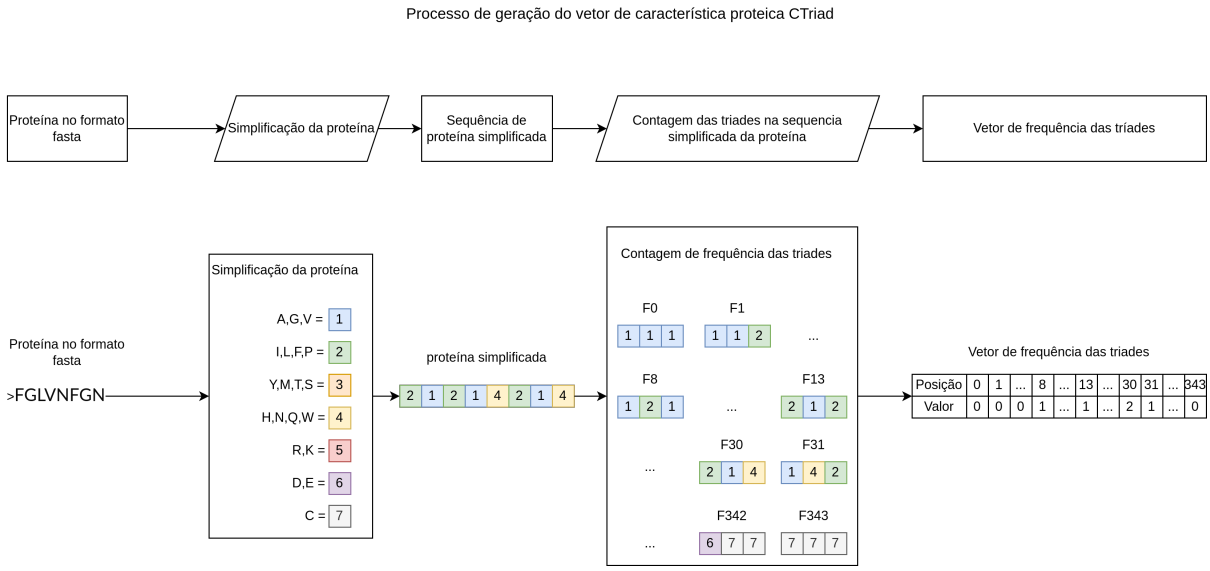

Figura 2.1: Extração do vetor de características CTriad

DDE O vetor de características DDE, proposto por Saravanan and Gautham (2015) para a previsão de epítopos de células B, é um tipo de conjunto de descritores de composição de sequências que se baseia na variação das composições de dipeptídeos em relação às médias esperadas. Para a construção do conjunto de descritores DDE, são calculados três parâmetros: a composição de dipeptídeos (Dc), a média teórica (Tm) e a variância teórica (Tv).

Geary O vetor de características Geary, desenvolvido em Geary (1954) como uma medida de autocorrelação espacial e implementada por Ong et al. (2007) para a previsão de famílias funcionais de proteína, utiliza a distribuição de propriedades estruturais e físico-químicas de aminoácidos ao longo da sequência.

Geary propõe o *Contiguity Ratio* como uma maneira de quantificar como os valores de uma variável mudam espacialmente entre áreas vizinhas. Ele é usado para avaliar a similaridade ou dissimilaridade de valores de uma variável em regiões geograficamente próximas. Essa medida complementa o Índice de Moran, outra medida popular de autocorrelação espacial.

O **Contiguity Ratio** C é definido como:

$$C = \frac{\sum_i \sum_j w_{ij}(x_i - x_j)^2}{2\sum_i (x_i - \bar{x})^2}$$

Onde:

- x_i e x_j: Valores da variável em regiões i e j.

- $\bar{x}$: Média dos valores da variável.
- w_{ij}: Elemento da matriz de pesos espaciais que indica o grau de proximidade ou conexão entre as regiões i e j.

O Geary é implementado hoje pelo servidor *PROFEAT* desenvolvido em Li et al. (2006) onde utiliza do *Contiguity Ratio* para extrair a dissimilaridade entre múltiplos descritores, como de hidrofobicidade e massa para gerar um vetor de 240 posições.

K-Sep_PSSM O K-Sep_PSSM proposto por Saini et al. (2016), é um conjunto de descritores baseado em transformações de colunas que calcula as probabilidades de transição de bigramas entre resíduos, levando em consideração as distâncias posicionais entre eles, correspondendo ao valor de "k". As probabilidades de transição são obtidas a partir de transformações em perfis de matrizes de pontuação específicas de posição de proteínas. Os perfis de proteínas representam a conservação evolutiva dos aminoácidos em uma sequência proteica, derivada de alinhamentos de múltiplas sequências de um conjunto homólogo de sequências de proteínas. Atualmente utiliza-se da ferramenta POSSUM desenvolvida por Wang et al. (2017) que serve para calcular o descritor K-Sep_PSSM.

PFAM O PFAM, utilizado inicialmente por Yamanishi et al. (2011) e Liu et al. (2015), representa perfis de domínios de proteínas, de acordo com as anotações de domínios proteicos no banco de dados PFAM El-Gebali et al. (2019), na forma de vetores de características binárias. Para cada proteína, ele codifica a presença (1) e a ausência (0) de uma lista única de domínios presentes nas proteínas do conjunto de dados correspondente.

QSO O QSO, utilizada pela primeira vez por Chou (2000), reflete o efeito indireto da ordem da sequência de proteínas ao calcular fatores de acoplamento com base nas distâncias entre resíduos contíguos na sequência. Essas distâncias são determinadas utilizando diferentes matrizes de distância de aminoácidos, como a matriz de distância de Schneider and Wrede (1994), que é derivada das propriedades de hidrofobicidade, hidrofilia, polaridade e volume das cadeias laterais dos aminoácidos.

SPMap Proposto por Sarac et al. (2008), para a classificação funcional de proteínas, o SPMap é um método de mapeamento de espaço de características que representa a composição da sequência ao calcular a distribuição de *clusters* de subsequências de proteínas de comprimento fixo (com 5 resíduos na versão padrão) em uma sequência proteica. Os *clusters* de subsequências são gerados utilizando semelhanças baseadas na matriz *BLOSUM62* de todas as possíveis subsequências no conjunto de proteínas fornecido, extraídas por meio da abordagem de janelas deslizantes.

2.3 Métrica de reatividade ChEMBL

O Índice de Reatividade ChEMBL, ou pChEMLBL é definido pelo logaritmo negativo da concentração de resposta para atividades biológicas, como IC50, EC50, XC50, AC50, Ki, Kd e Potência, com o objetivo de padronizar a identificação de eficácia em uma escala simplificada onde valores maiores representam maior reatividade entre os compostos.

- **IC50**: Concentração de um inibidor necessária para reduzir a atividade de um alvo biológico em 50%, é utilizada para medir a eficácia de inibidores e um valor menor representa um inibidor mais potente de forma que uma baixa amostra já possa inibir outro composto. No contexto de logaritmo negativo um valor maior representa maior reatividade por ser inversamente proporcional a quantidade de composto necessária para inibição.

- **EC50**: Concentração de um agonista, que é um composto químico que realiza uma ativação similar a um transmissor ou hormônio, que produz 50% da resposta máxima em um ensaio biológico. um valor baixo indica que o composto agonista é mais potente pois uma concentração baixa realiza o estímulo esperado. No contexto logaritmo negativo que utilizamos no pChEMBL um valor alto significa um composto com maior reatividade, pois é necessária uma concentração baixa do mesmo para realizar um resultado esperado.

- **XC50**: frequentemente usado em contextos de ensaios de citotoxicidade, que significa a capacidade de causar danos ou morte celular, refere-se a concentração que inibe 50% da resposta viabilidade celular de um ensaio, ou seja, a capacidade de sobrevivência celular. Assim como as concentrações anteriores, um valor menor apresenta maior efetividade da substancia por ser necessária uma menor concentração para reduzir a viabilidade celular. No contexto de logaritmo negativo, um valor maior representa um melhor resultado de reatividade.

- **AC50**: Concentração de um composto necessária para produzir 50% da resposta máxima. Essa resposta pode ser qualquer tipo de atividade biológica, como inibição de uma enzima, modulação de um receptor, ou qualquer outro efeito mensurável. Assim como todos as métricas de concentração anteriores, apresenta uma maior eficácia com valores menores, e com logaritmo negativo um valor maior apresenta uma eficácia maior.

- **Ki**: Medida da afinidade de um inibidor por um alvo, representando a concentração necessária para inibir 50% da atividade. Apresenta o mesmo comportamento das métricas anteriores.

- **Kd**: Constante que mede a afinidade de um ligante por um receptor, representando

a concentração na qual metade das moléculas de receptor estão ligadas ao ligante. Apresenta o mesmo comportamento das métricas anteriores.

- **Potência**: Quantidade de um fármaco necessária para produzir um efeito desejado. É frequentemente relacionada ao IC50 ou EC50 e apresenta os mesmos comparativos quanto a valores baixos e valores altos em seu logaritmo negativo.

O pChEMBL foi selecionado por ser a métrica principal na base de dados ChEMBL de onde é catalogado de forma aberta a reatividade entre compostos utilizados na criação da base.

2.4 Modelos de aprendizagem de máquina

Árvore de decisão As árvores de decisão são modelos de aprendizagem de máquina que podem ser utilizadas para classificação e regressão. Uma árvore de decisão é composta por nós que representam testes em atributos, ramos que representam os resultados desses testes e folhas que representam as classes ou valores finais.

Estrutura de uma Árvore de Decisão Uma árvore de decisão é composta por três componentes principais:

- **Nó Raiz:** O nó inicial que representa a totalidade dos dados.
- **Nódulos Internos:** Representam testes em atributos.
- **Folhas:** Representam as classes ou valores de saída.

Construção da árvore A construção de uma árvore de decisão envolve a seleção de um atributo para dividir os dados em subconjuntos. O objetivo é maximizar a pureza dos subconjuntos resultantes. Para isso, utilizamos medidas como a *Entropia* e o *Ganho de Informação*.

Entropia

A entropia é uma medida da incerteza ou impureza de um conjunto de dados. É definida como:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2.1}$$

onde $H(S)$ é a entropia do conjunto S, c é o número de classes e p_i é a proporção de instâncias da classe i no conjunto.

Ganho de informação

O ganho de informação é a redução da entropia após a divisão dos dados em subconjuntos. É calculado como:

$$IG(S, A) = H(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} H(S_v) \tag{2.2}$$

onde $IG(S, A)$ é o ganho de informação ao dividir o conjunto S pelo atributo A, S_v é o subconjunto de S para o valor v do atributo A, e $|S|$ é o número total de instâncias em S.

Critérios de parada

A construção da árvore continua até que um critério de parada seja atingido, que pode incluir:

- A árvore atinge uma profundidade máxima definida pelos parâmetros.
- O número de instâncias em um nó é menor que um limite pré-definido.
- A entropia de um nó é zero (todas as instâncias pertencem à mesma classe).

As árvores de decisão tendem a ser muito eficazes para classificação de grupos pequenos de dados e problemas com overfit, por consequência neste artigo foram utilizadas como modelos filhos do XGBoost e Random forest, modelos mais complexos que usam árvores de decisão como base e as otimizam utilizando técnicas de *begging* ou *boosting*.

Random forest O Random forest é um algoritmo que pode ser usado para classificação e regressão, proposto por Breiman (2001) e pertencente à categoria de modelos de *begging*, que consistem no agrupamento de múltiplos modelos de forma considerar a decisão majoritária do grupo, no Random Forest os modelos agrupados são de árvores de decisão. Foi uma das opções principais para a classificação pois apresentou resultados altos em artigos como Atas Guvenilir and Doğan (2023) e Prasetyo and Anggraeni (2024)

Processo de treinamento

Detalhando o processo de treinamento e estruturação interna do modelo de Random forest, temos a criação de um conjunto de árvores de decisão internas baseadas no hiperparâmetro de quantidade passado, como entrada de dados para as árvores de decisão filhas por padrão utiliza-se a técnica *bootstrap* para selecionar aleatoriamente as características e amostras da base de dados levada em consideração na criação de cada uma das sub árvores de decisão, quando habilitado o *bootstrap* reduz a probabilidade de *overfitting* por produzir uma maior variabilidade de valores considerados pelas árvores filhas do random forest, sendo importante evidenciar que o bootstrap pode também ser habilitado ou não por hiperparâmetros e pode gerar tendencias internas desconsiderando um dado ou considerando multiplas vezes outro, por se tratar de um decisor aleatório. Como hiperparâmetros, também podem

ser passados fatores globalmente utilizados nas árvores filhas, como profundidade máxima, que limita o tamanho de todas as árvores filhas do random forest prevenindo overfitting, e o critério de separação como gini ou impureza, entropia e perda logaritmica.

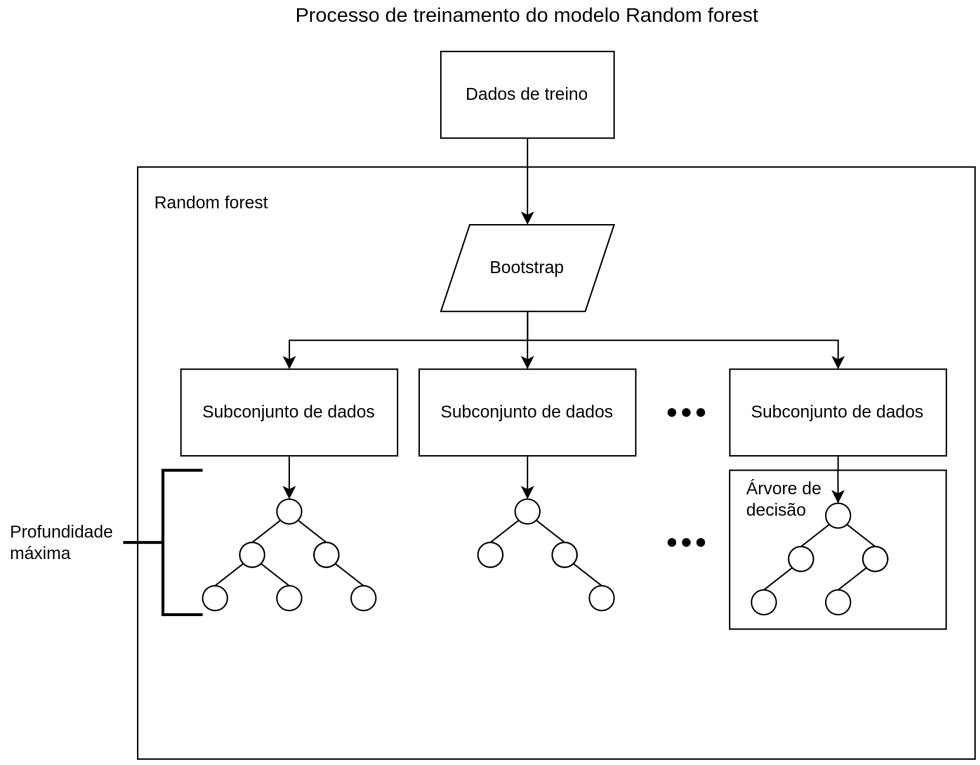

Figura 2.2: Detalhamento do processo de treinamento do modelo *Random forest*

Classificação

O processo de classificação do *Random forest* consiste no tratamento das respostas das árvores filhas, comumente chamado de votação, onde todos sub modelos geram a escolha de uma classe e por fim a classe com mais ocorrências no processo de votação é utilizado como saída da predição do modelo.

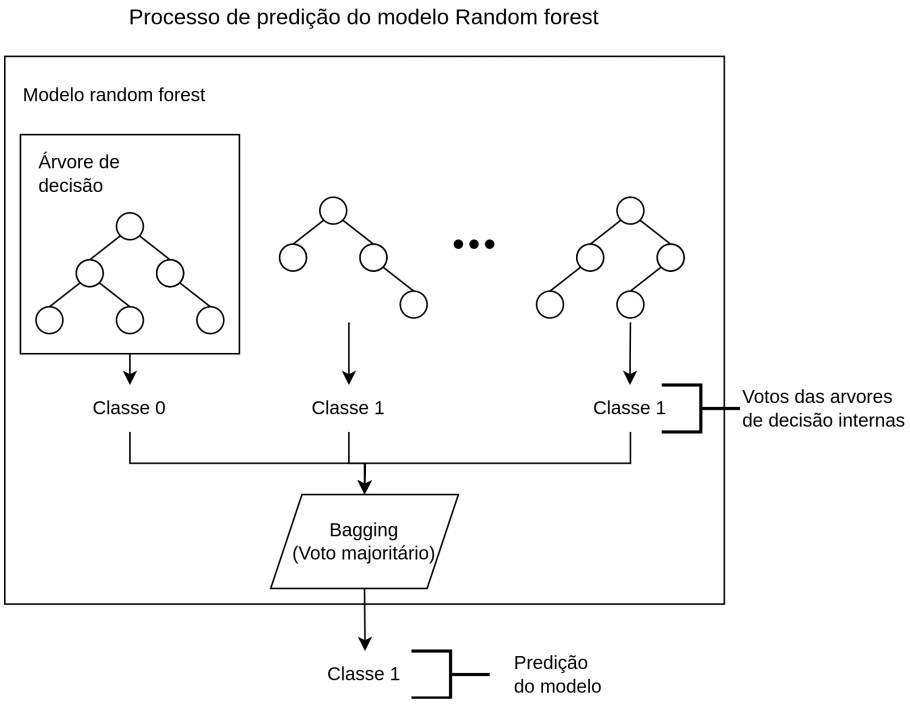

Figura 2.3: Processo de predição do modelo Random forest

Pontos fortes

O random forest tem como pontos fortes a menor suscetibilidade a overfitting quando comparado com árvores de decisão, menor sensibilidade a tunagem de hiperparâmetros e a maior facilidade classificação precisa de classes pequenas, por se tratar de um modelo de regras.

SVM Proposto por Cortes and Vapnik (1995) o funcionamento do modelo *Support Vector Machine* (SVM) para o contexto de classificação consiste na geração e posicionamento de um hiperplano que melhor separa espacialmente as classes. Um hiperplano pode ser entendido como uma linha em duas dimensões, um plano em três dimensões ou um espaço multi dimensional, e sua posição é determinada pela maximização da margem, que é a distância entre o hiperplano e os pontos de dados mais próximos de cada classe, conhecidos como vetores de suporte. Processo de treinamento Durante o treinamento do modelo, o SVM recebe um conjunto de dados rotulados e resolve um problema de otimização. A função objetivo é maximizar a margem entre as classes, ao mesmo tempo em que minimiza o erro de classificação. Os vetores de suporte são os pontos de dados que estão mais próximos do hiperplano e são cruciais para a definição do modelo, pois a posição do hiperplano depende apenas deles. Outros pontos de dados que não estão próximos do hiperplano não influenciam a solução.

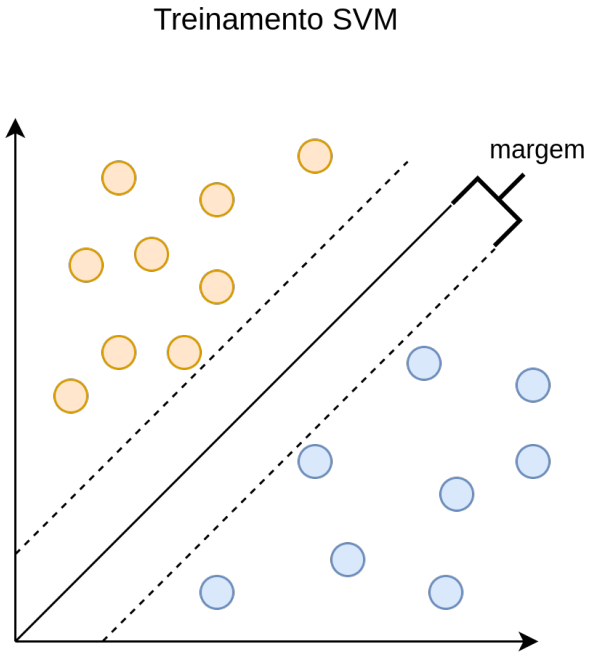

Figura 2.4: Processo de treinamento SVM

Um hiperplano em um espaço de n dimensões pode ser definido pela seguinte equação:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{2.3}$$

onde:

- $\mathbf{w}$ é o vetor de pesos (normal ao hiperplano),
- $\mathbf{x}$ é o vetor de entrada,
- b é o termo de viés (bias).

Margem A margem é a distância entre o hiperplano e os pontos de dados mais próximos de cada classe, conhecidos como vetores de suporte. A margem é calculada como:

$$\text{Margem} = \frac{2}{\|\mathbf{w}\|} \tag{2.4}$$

O objetivo do SVM é maximizar essa margem, o que implica minimizar $\|\mathbf{w}\|$.

Otimização do Hiperplano Para encontrar o hiperplano que maximiza a margem, formulamos o seguinte problema de otimização:

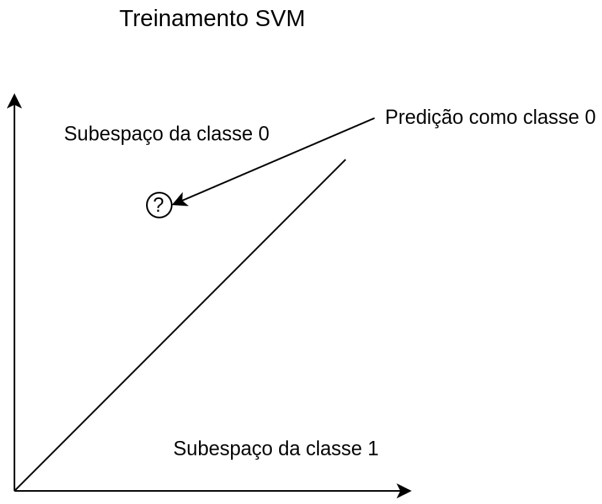

Figura 2.5: Processo de classificação SVM

$$\text{Minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{2.5}$$

$$\text{Subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i \tag{2.6}$$

onde y_i é a classe do ponto de dados $\mathbf{x}_i$ (com valores $+1$ ou -1).

Lagrangeano Para resolver o problema de otimização, utilizamos o método dos multiplicadores de Lagrange. O Lagrangeano é definido como:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \tag{2.7}$$

onde $\alpha_i \geq 0$ são os multiplicadores de Lagrange.

Para o processo de classificação temos o comparativo com a o hiperplano de margens otimizadas, que separa o espaço, calculando a posição do ponto a ser classificado e o atribuindo a classe que representa o subespaço que a reta corta como se pode ver na Figura 2.5.

KNN Proposto por Cover and Hart (1967) o modelo *K-Nearest Neighbours* (KNN) pode ser utilizado para classificação ou regressão. O funcionamento do KNN no contexto de classificação é definido pelo processo de armazenamento dos dados no treino e como é detalhando visualmente na Figura 2.6, onde ha a predição de uma classe para um elemento novo utilizando de k igual a 3, no teste a busca por K elementos próximos do valor a ser predito utilizando uma métrica de distância, como por exemplo: distância Euclidiana, manhattan ou minkowski, e contabilizando a classe com maior frequência no conjunto de K elementos mais próximos a retornando como

seu valor de predição.

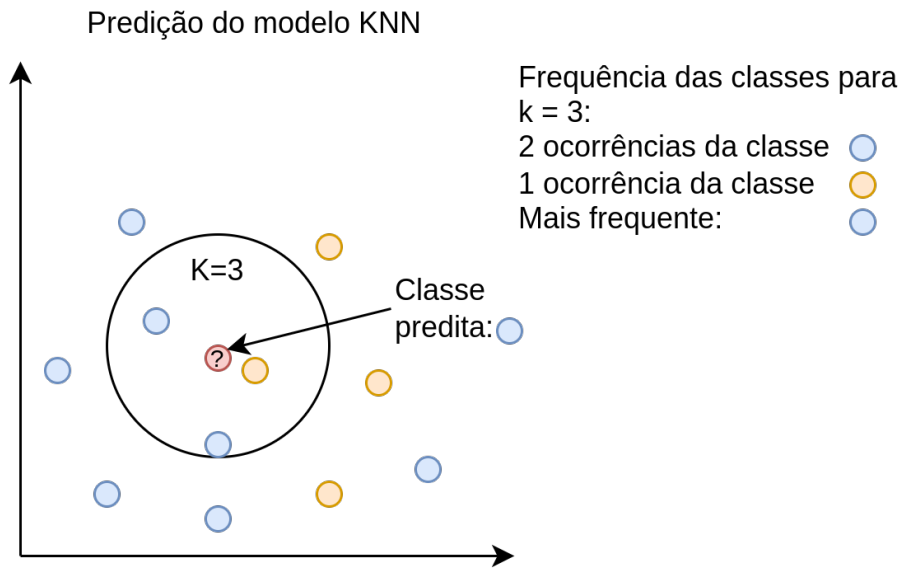

Figura 2.6: Processo de classificação KNN

Como maior detalhamento das distâncias temos:

- Distância Euclidiana:

$$d_E(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- Distância Manhattan:

$$d_M(x, y) = \sum_{i=1}^{n}|x_i - y_i|$$

- Distância de Minkowski:

$$d_{Mink}(x, y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}}$$

  - onde p é um parâmetro que define a ordem da distância (por exemplo, $p = 2$ para a distância Euclidiana e $p = 1$ para a distância Manhattan).

O KNN foi selecionado para a categorização do trabalho atual por sua capacidade de classificação de múltiplas classes espacialmente agrupadas, o que o diferencia da maioria dos modelos utilizados que demandam de uma maior separação espacial.

MLP A MLP é composta por múltiplas camadas de neurônios artificiais, do tipo perceptron propostos por McCulloch and Pitts (1943), incluindo uma camada de entrada,

uma ou mais camadas ocultas e uma camada de saída. Cada neurônio em uma camada está conectado a todos os neurônios da camada seguinte, formando uma estrutura densa que permite a modelagem de relações complexas entre os dados.

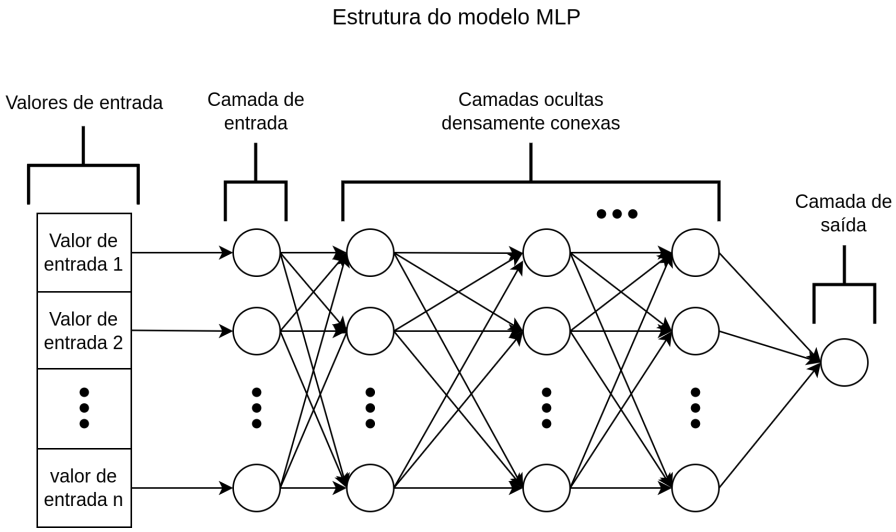

Figura 2.7: Estrutura de uma rede MLP

O funcionamento da MLP se baseia em um processo de aprendizado supervisionado. Durante a fase de treinamento, a rede recebe um conjunto de dados rotulados e ajusta os pesos das conexões entre os neurônios para minimizar a diferença entre as previsões da rede e os rótulos reais. Esse ajuste é realizado através de um algoritmo chamado retropropagação (backpropagation), que calcula o gradiente do erro em relação aos pesos e os atualiza usando um método de otimização, como o gradiente descendente. O modelo MLP utiliza funções de ativação, como a função ReLU (Rectified Linear Unit) ou tanh, para introduzir não linearidades no modelo.

De forma mais detalhada a saída de um neurônio é calculada como uma combinação linear das entradas, seguida por uma função de ativação. A equação para a saída de um neurônio j na camada l pode ser expressa como:

$$z_j^{(l)} = \sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}$$

onde:

- $z_j^{(l)}$ é a entrada do neurônio j na camada l,
- $w_{ij}^{(l)}$ é o peso da conexão entre o neurônio i da camada anterior e o neurônio j da camada l,
- $a_i^{(l-1)}$ é a saída do neurônio i da camada anterior,
- $b_j^{(l)}$ é o viés do neurônio j na camada l.

A saída do neurônio é então calculada aplicando uma função de ativação f:

$$a_j^{(l)} = f(z_j^{(l)})$$

Sendo essa função de ativação definida por hiperparâmetros, tendo opções como Tanh e Relu.

A função tangente hiperbólica (tanh) é uma função de ativação que mapeia a entrada para um intervalo entre -1 e 1. Sua fórmula é dada por:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

A derivada da função tanh é:

$$f'(x) = 1 - \tanh^2(x)$$

A função de ativação ReLU (Rectified Linear Unit) é definida como:

$$f(x) = \max(0, x)$$

A derivada da função ReLU é:

$$f'(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases}$$

De forma detalhada o backpropagation é utilizado para treinar redes neurais, ajustando os pesos e viéses com base no erro da saída. O processo envolve duas etapas: a propagação da entrada para frente e a retropropagação do erro.

Cálculo do Erro

Seja y a saída desejada e $\hat{y}$ a saída da rede, o erro pode ser calculado como:

$$E = \frac{1}{2}(y - \hat{y})^2$$

Atualização dos Pesos

A atualização dos pesos é feita utilizando a regra da cadeia. A derivada do erro em relação ao peso $w_{ij}^{(l)}$ é dada por:

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \delta_j^{(l)} a_i^{(l-1)}$$

onde $\delta_j^{(l)}$ é o erro do neurônio j na camada l, que pode ser calculado como:

$$\delta_j^{(l)} = (a_j^{(l)} - y) f'(z_j^{(l)})$$

Os pesos são atualizados da seguinte forma:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial E}{\partial w_{ij}^{(l)}}$$

Onde:

- η: É a taxa de aprendizado definida pelos parâmetros do modelo.
- $\frac{\partial E}{\partial w_{ij}^{(l)}}$: É a derivada do erro em relação ao peso $w_{ij}^{(l)}$. Essa derivada indica como o erro muda em resposta a uma pequena mudança no peso.
- $\delta_j^{(l)}$: É o termo de erro do neurônio j na camada l. Ele é calculado com base no erro da saída e na derivada

XGBoost O XGBoost (Extreme Gradient Boosting) é um algoritmo de aprendizado de máquina que pode ser utilizado para classificação e regressão, ele é uma implementação otimizada do método de boosting, que combina múltiplos modelos, geralmente árvores de decisão, para criar um modelo mais robusto corrigindo erros de predição do modelo composto anterior.

O funcionamento do XGBoost baseia-se na ideia de que, em vez de treinar um único modelo complexo, é mais eficaz treinar uma série de modelos simples que se corrigem mutuamente. O algoritmo começa com um modelo inicial o treina e em seguida, ele itera, ajustando novos modelos para prever os erros dos modelos anteriores. Cada nova árvore é treinada para minimizar a função de perda, que mede a diferença entre as previsões do modelo e os valores reais. O XGBoost utiliza uma técnica chamada regularização, que ajuda a evitar o *overfitting*, controlando a complexidade do modelo.

De forma mais detalhada temos: A função objetivo do XGBoost é composta por duas partes: a função de perda e a regularização. A função objetivo pode ser expressa como:

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

onde:

- $L(\theta)$: É a função objetivo total que queremos minimizar.

- n: É o número de exemplos de treinamento.
- $l(y_i, \hat{y}_i)$: É a função de perda que mede a diferença entre a saída real y_i e a saída prevista $\hat{y}_i$ para o exemplo i.
- K: É o número total de árvores no modelo.
- $\Omega(f_k)$: É a função de regularização que penaliza a complexidade da árvore f_k.

A função de perda l pode ser escolhida de acordo com o tipo de problema (regressão, classificação, etc.). Para problemas de classificação, uma função de perda comum é a log-loss:

$$l(y_i, \hat{y}_i) = -[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

A regularização é uma parte importante do XGBoost, pois ajuda a evitar o *overfitting*. A função de regularização $\Omega(f_k)$ é geralmente definida como:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

onde:

- T: É o número de folhas na árvore f_k.
- γ: É um parâmetro que controla a complexidade da árvore (número mínimo de folhas).
- w_j: É o peso da folha j da árvore.

O XGBoost utiliza uma abordagem de otimização chamada *Gradient Boosting*, onde os pesos das árvores são atualizados com base no gradiente da função de perda. A atualização dos pesos pode ser expressa como:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$$

onde:

- $\hat{y}_i^{(t)}$: É a previsão atualizada para o exemplo i na iteração t.
- $\hat{y}_i^{(t-1)}$: É a previsão anterior para o exemplo i.
- η: É a taxa de aprendizado (ou shrinkage), que controla a contribuição de cada árvore.
- $f_t(x_i)$: É a saída da árvore t para o exemplo x_i.

2.5 DTI

A modelagem de interações entre drogas e alvos (DTI), popularizado por Chen et al. (2016), envolve a criação de uma base de dados especializada em um composto para prever a reatividade de uma substancia alvo, por classificação ou regressão. Para essa abordagem é necessário catalogar os coeficientes de reações de compostos relacionados ao alvo, assim como seus devidos vetores de características como pode ser visto na Figura 2.8.

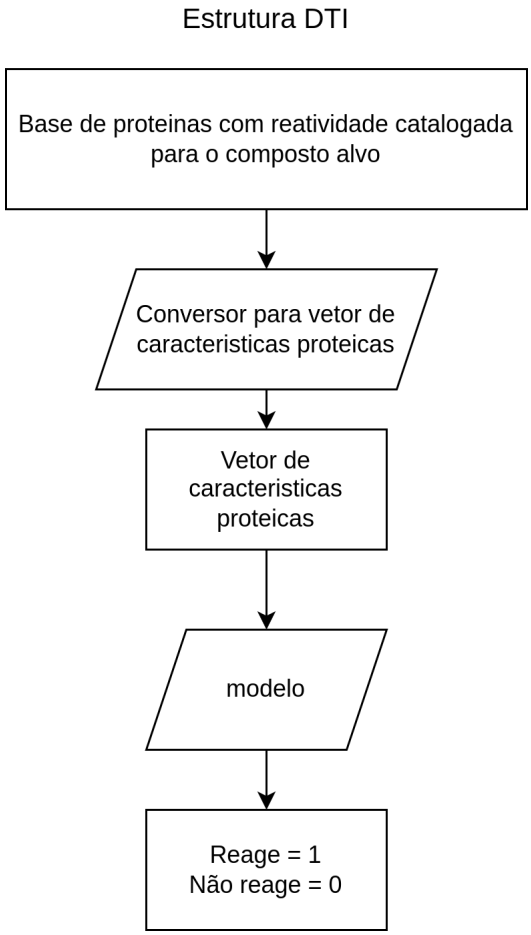

Figura 2.8: Representação gráfica do funcionamento de *DTI*

2.6 Métricas de avaliação

As métricas de avaliação são utilizadas para avaliar a performance e investigar tendencias nos resultados dos modelos em relação aos casos de teste, que de forma indutiva é utilizada para estimar a performance real do modelo. Foram selecionadas as métricas acurácia, precisão, recall e f1 score por suas características específicas de demonstrar falhas nos resultados dos modelos, de forma mais detalhada temos:

Acurácia A Acurácia é a proporção de previsões corretas em relação ao total de previsões realizadas, foi uma das métricas selecionadas por ser a mais simples e amplamente

utilizada para avaliar o resultado de modelos de aprendizagem de máquina.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}$$

onde:

- TP (True Positives) - Verdadeiros Positivos
- TN (True Negatives) - Verdadeiros Negativos
- FP (False Positives) - Falsos Positivos
- FN (False Negatives) - Falsos Negativos

Precisão A Precisão mede a proporção de verdadeiros positivos em relação ao total de positivos previstos. É uma métrica importante quando o custo de falsos positivos é alto e portanto foi utilizada para evidenciar tendencias nos modelos em casos de classificação de reação falsa que é algo que precisamos evitar no processo de experimentação laboratorial.

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{2.9}$$

Recall O Recall, também conhecido como Sensibilidade ou Revocação, mede a proporção de verdadeiros positivos em relação ao total de positivos reais. É crucial em situações onde é importante capturar todos os casos positivos sendo importante para o descobrimento de novos fármacos.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.10}$$

F1 Score O F1 Score é a média harmônica entre Precisão e Recall, oferecendo uma única métrica que considera tanto falsos positivos quanto falsos negativos. É especialmente útil em conjuntos de dados desbalanceados sendo uma métrica essencial para o caso atual onde a base de dados pode apresentar desbalanceamento pelo tamanho reduzido.

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{2.11}$$

Capítulo 3

Metodologia

A previsão binária de reatividade entre compostos usando DTI pode ser categorizada como um problema de aprendizado supervisionado que depende da segmentação de indices de reatividade catalogados em grupos reagentes e não reagentes. Para cada composto químico alvo é criada uma base de dados contendo as informações proteicas das proteínas com reatividade catalogada para composto, depois a cadeia proteica que define essa proteína passa por um processo de extração de características para cada um dos vetores de características utilizado no artigo, que serão descritos nominalmente na seção referente a geração do banco de dados, e para cada composto alvo e tipo de vetor de característica proteica é rodada uma instância para cada modelo classificador utilizado na análise do presente artigo, todos eles utilizado de uma otimização de parâmetros do tipo *Grid search*.

Esse capítulo tem como objetivo detalhar o desenvolvimento do trabalho e os parâmetros utilizados. Serão apresentados parâmetros de modelos, vetores de características proteicas e compostos-alvo no contexto da classificação DTI (Interação entre Droga e Alvo).

3.1 Base de dados

No contexto de DTI foram utilizadas as bases geradas no artigo Atas Guvenilir and Doğan (2023), de compostos centrais ChEMBL44, ChEMBL50, ChEMBL83, ChEMBL91, ChEMBL104, ChEMBL808, ChEMBL116438 e ChEMBL295698, onde para cada composto é extraída da base *ChEmbl*, apresentada em Gaulton et al. (2016), proteínas com índice de reatividade *ChEMBL* para seus compostos centrais alvos descritos anteriormente, para cada proteína é adquirida sua representação proteica em *FASTA* presente na base *Uniprot*, apresentada por Consortium (2020), e é gerada a extração das representações utilizando o software *IFeature*, desenvolvido por Chen et al. (2018), para as seguintes representações: APAAC, CTDD, CTRiad, DDE, Geary, K-SEP PSSM, PFAM, QSO, SPMAP. Para *baseline* de representações temos o random200 desenvolvido por Atas Guvenilir and Doğan (2023) consistindo em uma lista de 200 linhas com valores aleatórios

no conjunto dos reais entre 0 e 1.

Para a classificação binária dos modelos os valores de reação *ChEMBL* são separados em não reagentes e reagentes, para valores não reagentes foram separadas as proteínas com coeficiente *ChEMBL* de reação menor ou igual a 5 e para as reagentes, com valores maiores que 5, sendo convertidos para 0 e 1 respectivamente.

O tamanho das bases para cada composto varia entre 76 a 294 proteínas catalogadas, de forma detalhada temos os compostos com suas quantidades de amostras de proteínas com informações de reatividade catalogadas na Tabela 3.1.

Composto	**Quantidade de Amostras**
ChEMBL44	204
ChEMBL50	184
ChEMBL83	294
ChEMBL91	98
ChEMBL104	76
ChEMBL808	108
ChEMBL116438	188
ChEMBL295698	158

Tabela 3.1: Quantidade de Amostras de Reatividade por Composto

A quantidade de colunas nos vetores de características proteicas variam de 80 a 1298 colunas, se mantendo constantes para cada representação, mais detalhes na Tabela 3.2

Vetor de características proteicas	**Quantidade de Colunas**
APAAC	80
CTDD	195
CTRIAD	343
DDE	400
Geary	240
K-SEP PSSM	400
PFAM	1298
QSO	100
SPMAP	542
Random200	200

Tabela 3.2: Tabela da quantidade de colunas por vetor de características proteicas

3.2 Modelos

Para cada um dos compostos alvo foram aplicados os modelos: *Random forest* (*RandomForestClassifier*), KNN(KNeighborsClassifier), SVM(SVC), MLP (MLPClassifier), XGBoost e *Dummy* (DummyClassifier), sendo o *XGBoost* de uma biblioteca XGboost de

Chen and Guestrin (2016) na versão 2.1.2 e os outros classificadores da biblioteca *scikit-learn* de Pedregosa et al. (2011) na versão 1.3.2. Em exceção ao *Dummy* foi utilizado nos modelos *Grid Search* para otimização de hiperparâmetros. Para todos os modelos foi utilizada a separação *Random split* com razão 80% para a base de treino e 20% para a base de teste. Como métrica de avaliação foram utilizadas as métricas*Accuracy*(Acurácia), *F1 Score*, Precisão e *recall* (Revocação ou Sensibilidade). Como parâmetro de seleção do melhor modelo pelo *GridSearch* foi utilizada a métrica de acurácia.

3.2.1 Otimização de hiperparâmetros

Para otimização de hiperparâmetros foi utilizado o algoritmo *Grid Search*, utilizado em LaValle et al. (2004), que consiste na utilização de uma tabela de parâmetros definidos pelo usuário e a partir dos valores definidos para a esta tabela de hiperparâmetros são geradas todas as combinações possíveis e executadas para a base de dados utilizada, avaliando o resultado de cada execução e retornando o melhor modelo e subconjunto de parâmetros utilizados que resultaram na maximização da métrica em relação às outras combinações geradas. Para cada um dos modelos foi gerado um conjunto de valores para o grid search apresentados a seguir:

MLP

- tamanhos das camadas ocultas: Define a estrutura da rede neural, especificando o número de neurônios em cada camada oculta. Valores utilizados: uma camada oculta com 50 perceptrons, uma camada oculta com 100 perceptrons, duas camadas ocultas com 50 perceptrons cada.
- função de ativação: Determina a função utilizada para ativar os neurônios. Valores utilizados: *tanh*, *relu*.
- coeficiente de regularização: Controla a regularização L2, ajudando a prevenir *overfitting*. Valores utilizados: "0.0001", "0.001", "0.01".

Random Forest

- número de estimadores: Indica quantas árvores de decisão serão construídas no modelo. Valores: 50, 100, 200.
- profundidade máxima: Limita a profundidade das árvores, ajudando a controlar o *overfitting*. Valores utilizados: *None*, 10, 20, 30.
- mínimo de amostras para divisão: Define o número mínimo de amostras necessárias para dividir um nó. v utilizados: 2, 5, 10.

SVC

- parâmetro de regularização: Controla a penalização de erros de classificação. Valores utilizados: "0.1", 1, 10.
- função do kernel: Especifica a função utilizada para transformar os dados. Valores utilizados: *linear*, *rbf*.
- coeficiente gamma: Define a influência de um único exemplo de treinamento. Valores utilizados: *scale*, *auto*.

KNN

- número de vizinhos: Indica quantos vizinhos mais próximos serão considerados para a classificação. Valores utilizados: 3, 5, 7, 10.

XGBoost

- número de estimadores: Indica quantas árvores serão construídas no modelo. Valores utilizados: 50, 100.
- profundidade máxima: Limita a profundidade das árvores, ajudando a controlar o *overfitting*. Valores utilizados: 3, 5, 7.
- taxa de aprendizado: Controla a contribuição de cada árvore para a previsão final. Valores utilizados: "0.01", "0.1", "0.2".

Dummy Classifier

estratégia: Define a estratégia utilizada para fazer previsões. Valor: mais frequente.

Capítulo 4

Resultados e Discussões

O objetivo desta seção é avaliar o desempenho dos diferentes modelos de predição de reatividade entre compostos e proteínas, a fim de fornecer recomendações sobre quais abordagens utilizar. Para isso, foram gerados gráficos que ilustram os resultados específicos para cada composto-alvo, além da classificação de cada modelo em relação às representações proteicas utilizadas como entrada.

Os resultados das execuções dos modelos foram armazenados em arquivos CSV, e a manipulação e geração dos gráficos foram realizadas utilizando a linguagem Python (versão 3.8.19) e as bibliotecas Pandas pandas development team (2020) (versão 2.0.3), Seaborn Waskom (2021) (versão 0.13.2), Matplotlib Hunter (2007) (versão 3.4.3) e NumPy Harris et al. (2020) (versão 1.24.3).

A análise das acurácias médias de classificação de todos os modelos, com hiperparâmetros otimizados, em relação aos vetores de representação proteica não revelou um modelo com desempenho consistentemente superior. Como ilustrado na Figura 4.1, o eixo Y representa a acurácia média das predições dos modelos em relação a todos os compostos-alvo da base ChEMBL, enquanto o eixo X apresenta os diferentes vetores de características utilizados como entrada. As curvas no gráfico correspondem aos modelos, e os pontos em cada representação indicam o resultado médio da classificação para cada uma das bases de proteínas catalogadas, considerando os oito compostos-alvo.

Na Figura 4.1, observa-se que todos os modelos apresentaram, em pelo menos um dos vetores de características proteicas, uma performance média variando entre 67% e 73%. Notou-se uma tendência de picos de desempenho entre pares de modelo-característica, destacando-se os seguintes resultados: CTriad com XGBoost (72,78%), DDE com Random Forest (71,08%), K-SEP PSSM com KNN (70,31%) e SVM Classifier com Geary (69,51%). Esses resultados sugerem a necessidade de uma investigação mais aprofundada sobre as interações entre os modelos e as características.

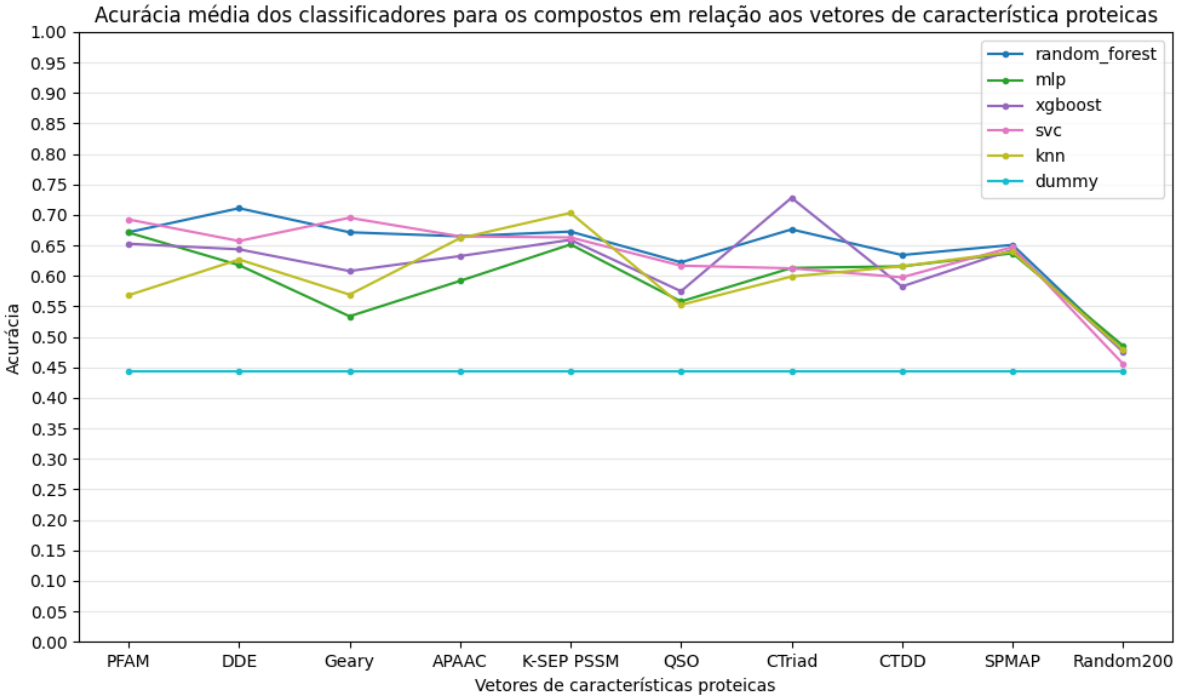

Figura 4.1: Resultados da média da acurácia da predição dos modelos para todos os compostos em relação aos vetores de características proteicas

É importante ressaltar que o modelo MLP apresentou desempenho abaixo da média, conforme evidenciado na Figura 4.1. Essa limitação pode ser atribuída à estruturação da base de dados, que contém uma quantidade relativamente baixa de compostos, variando de 76 a 294, conforme mostrado na Tabela 3.1. Essa escassez de dados dificulta a geração de bases que favoreçam a utilização de modelos mais complexos, como o MLP, que requerem um volume maior de dados para alcançar uma generalização eficiente e resultados consistentes. Em contrapartida, essa característica da base de dados valoriza modelos baseados em regras, como o XGBoost e o Random Forest, que se mostraram mais eficazes para o conjunto de dados atual.

4.1 XGboost, Random forest e CTriad

O modelo XGBoost, utilizando o vetor de características CTriad como entrada, apresentou uma média elevada de acurácia, conforme mostrado na Figura 4.1. Devido a esse desempenho, investigamos mais detalhadamente os resultados médios, analisando as métricas extraídas da classificação realizada pelos modelos. Para isso, foram gerados gráficos de caixa, apresentados nas Figuras 4.2 e 4.3.

Na Figura 4.2, o eixo x representa as diferentes representações proteicas utilizadas para cada base de dados de compostos ChEMBL, enquanto o eixo y mostra a acurácia de cada

execução para os 8 compostos analisados. As caixas apresentam os valores obtidos pelo modelo XGBoost. Já na Figura 4.3, o eixo x contém as representações proteicas utilizadas nas proteínas que interagem com o composto-alvo como entrada para o modelo, e o eixo y exibe os valores da métrica F1-score para cada execução entre todos os 8 compostos.

Os resultados indicam que, para a representação CTriad, o terceiro quartil, a mediana e o limite superior da caixa se situam em uma faixa de 72,4% a 88,2% de acurácia, acompanhados por valores elevados de F1-score também com terceiro e quarto quartis na faixa de 72% a 88% de performance. Esses números sugerem que os dados são altamente separáveis, o que favorece o desempenho de classificadores baseados em regras, como o XGBoost, pode ser evidenciado tabém boas performances isoladas nas representações DDE e K-SEP PSSM comm limites superiores das caixas na faixa de 80% ou mais, podendo ser boas representações para estudos e aprofundamentos futuros na execução de uma maior quantidade de instâncias. Além disso, é importante destacar que o modelo Random Forest também apresentou um bom desempenho médio com a representação CTriad, quando comparado aos demais modelos no gráfico da Figura 4.1.

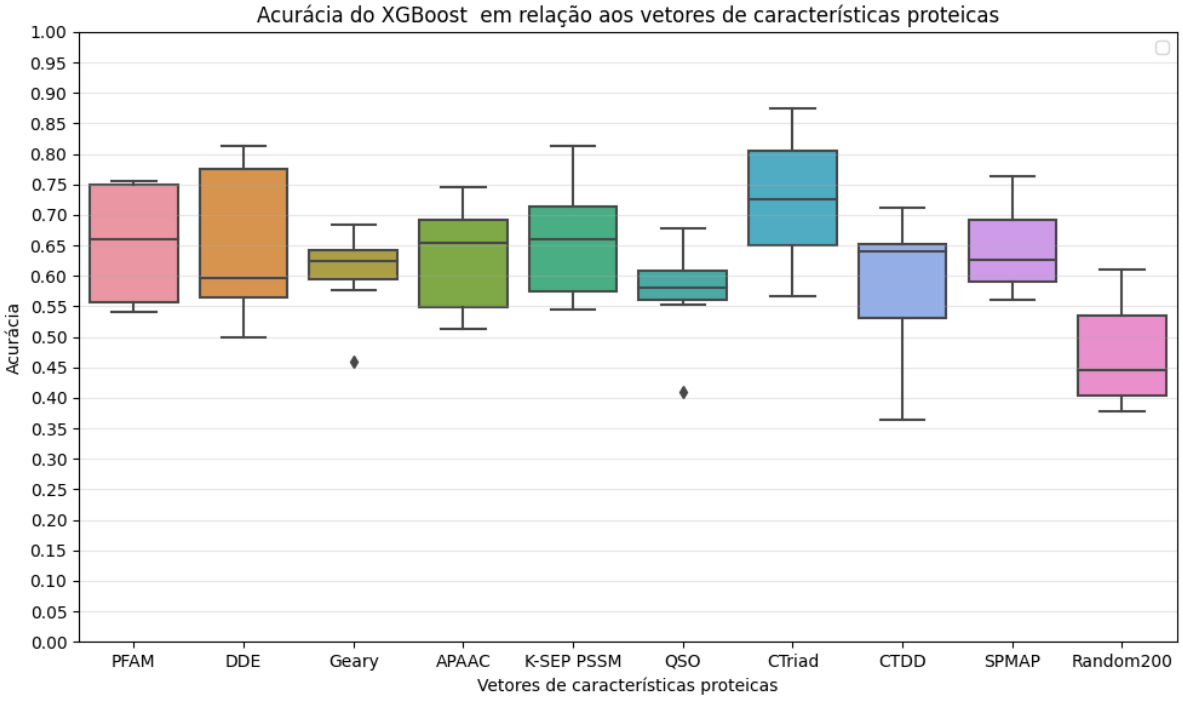

Figura 4.2: Acurácia do XGboost em relação aos vetores de características proteicas

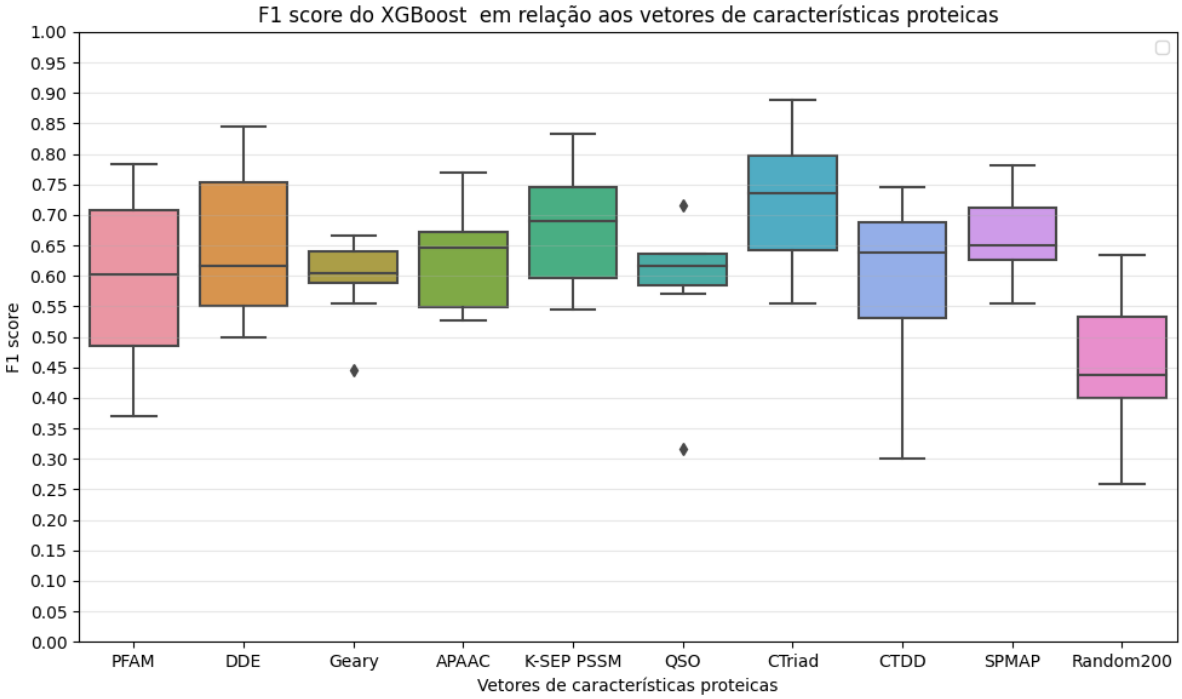

Figura 4.3: F1 score do XGboost em relação aos vetores de características proteicas

4.2 Random forest e DDE

Os modelos DDE e Random Forest também se destacaram no gráfico de médias de classificação. Ao analisarmos as Figuras 4.4 e 4.5, observamos que para a representação dde temos uma consistência alta nos dados onde a dispersão da caixa é bem baixa e majoritariamente entre 65% e 78%, apresentando uma boa característica na classificação, é possível ver também que para as representações Geary e CTDD houve uma grande dispersão dos dados, inclusive sendo apresentado como *outlier* na representação geary que atingiu aproximadamente 88% de acurácia, podendo ser um indicativo uma baixa qualidade de pareamento entre o modelo e as representações necessitando de avaliações posteriores com uma maior gama de dados e execuções.A instância correspondente ao *outlier* é ilustrado na Figura 4.6 onde temos o resultado da métrica acurácia na classificação de reatividade de proteínas em relação ao composto ChEMBL104.

Na Figura 4.6, o eixo x representa os vetores de características proteicas utilizados como entrada para o modelo, contendo os dados das proteínas que interagem com o composto ChEMBL104. Já o eixo y exibe a acurácia obtida pelos modelos na classificação dessa base. Vale destacar que o ChEMBL104 possui o menor número de amostras entre os compostos analisados, com apenas 76 proteínas catalogadas, conforme indicado na Tabela 3.1.

Esse cenário sugere que o desempenho otimista observado pode estar relacionado à

natureza dos modelos baseados em regras, como o Random Forest. Esse comportamento também é evidenciado para o modelo XGBoost na mesma figura, reforçando a capacidade desses algoritmos em lidar com conjuntos de dados de menor dimensão, apresentando resultados elevados mesmo em cenários com poucas amostras.

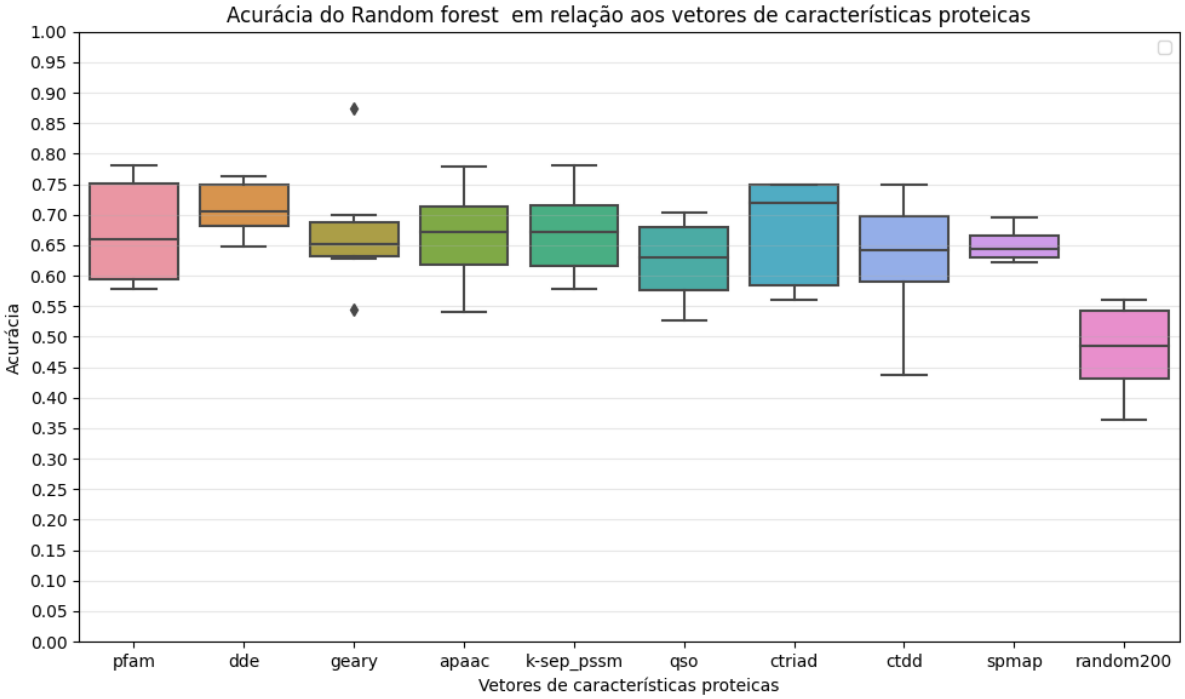

Figura 4.4: Acurácia do Random forest em relação aos vetores de características proteicas

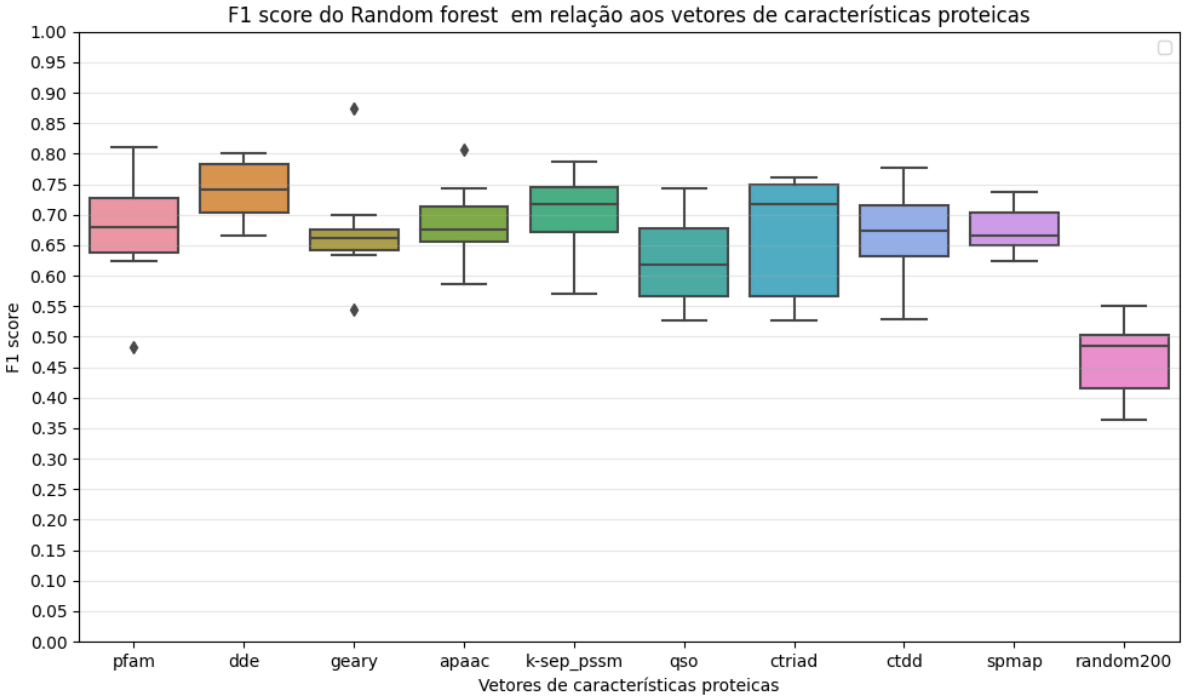

Figura 4.5: F1 score do Random forest em relação aos vetores de características proteicas

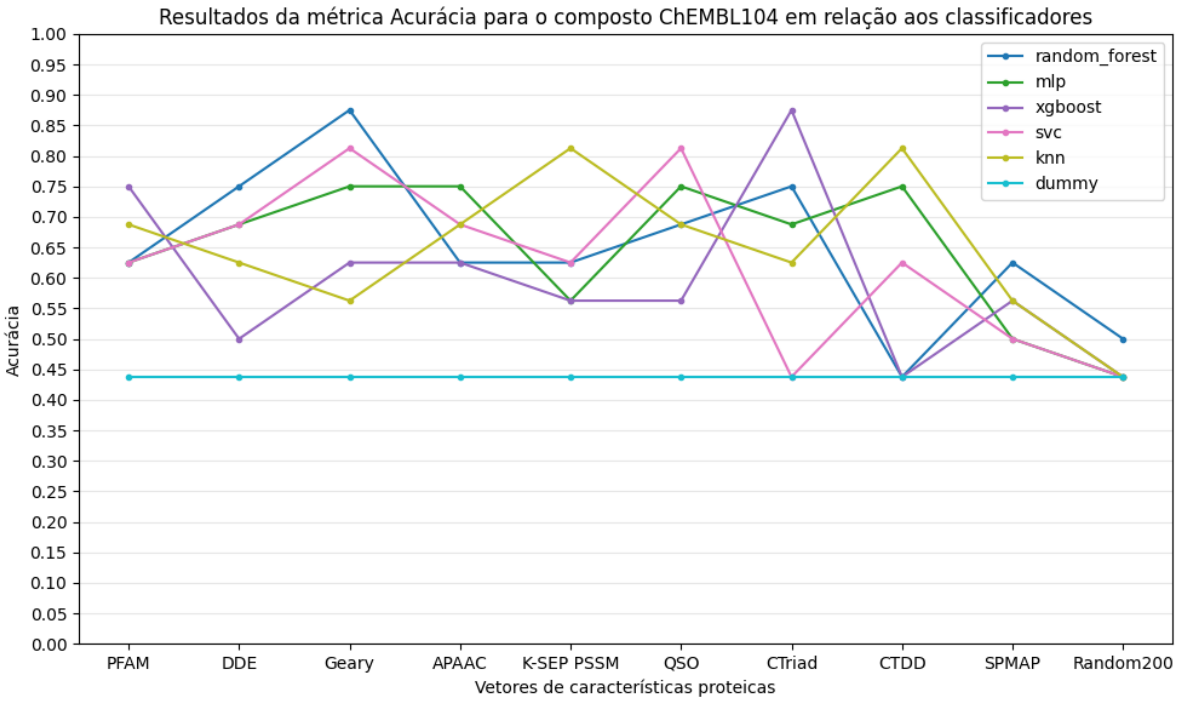

Figura 4.6: Resultado da métrica acurácia para o composto ChEMBL 104 para todos os modelos

4.3 KNN e K-SEP PSSM

KNN e K-SEP PSSM compartilham um padrão que considera o agrupamento e a separação espacial como fatores de similaridade, apresentando possivelmente um agrupamento maior permitindo a maior performance apresentada no KNN. Dado que esses aspectos são fundamentais em suas definições, é compreensível que seu pareamento e desempenho se destaquem, apresentando resultados acima da média onde pode ser visto o detalhamento da sua acurácia na Figura 4.7, onde no eixo x temos os vetores de características proteicas utilizadas como entrada para o modelo e no eixo y a acurácia resultante de cada uma das 8 execuções.

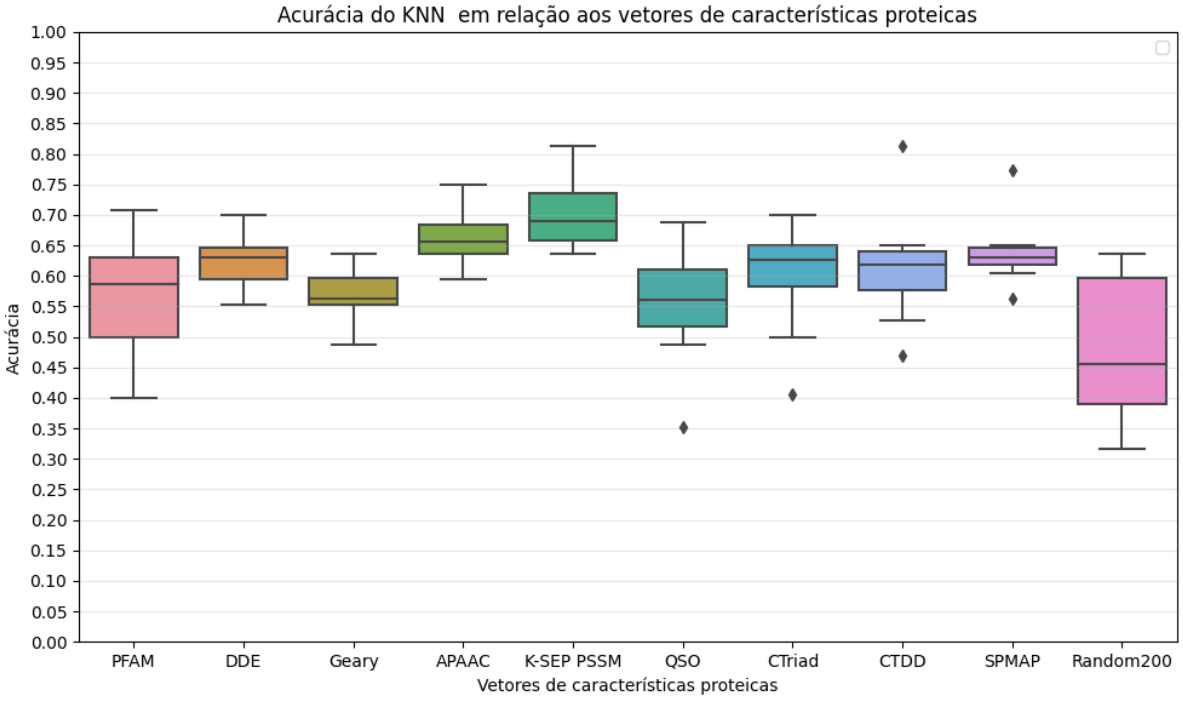

Figura 4.7: Acurácia do KNN em relação aos vetores de características proteicas

4.4 SVM e Geary

O SVM classifier apresentaram na grande maioria dos compostos uma das melhores performances quando comparado com outros modelos utilizando o vetor de características proteicas Geary, como pode ser visto nas Figuras 4.8, e 4.9, ambos gráficos contendo os vetores de características proteicas de entrada no eixo x e o resultado das métricas no eixo y.

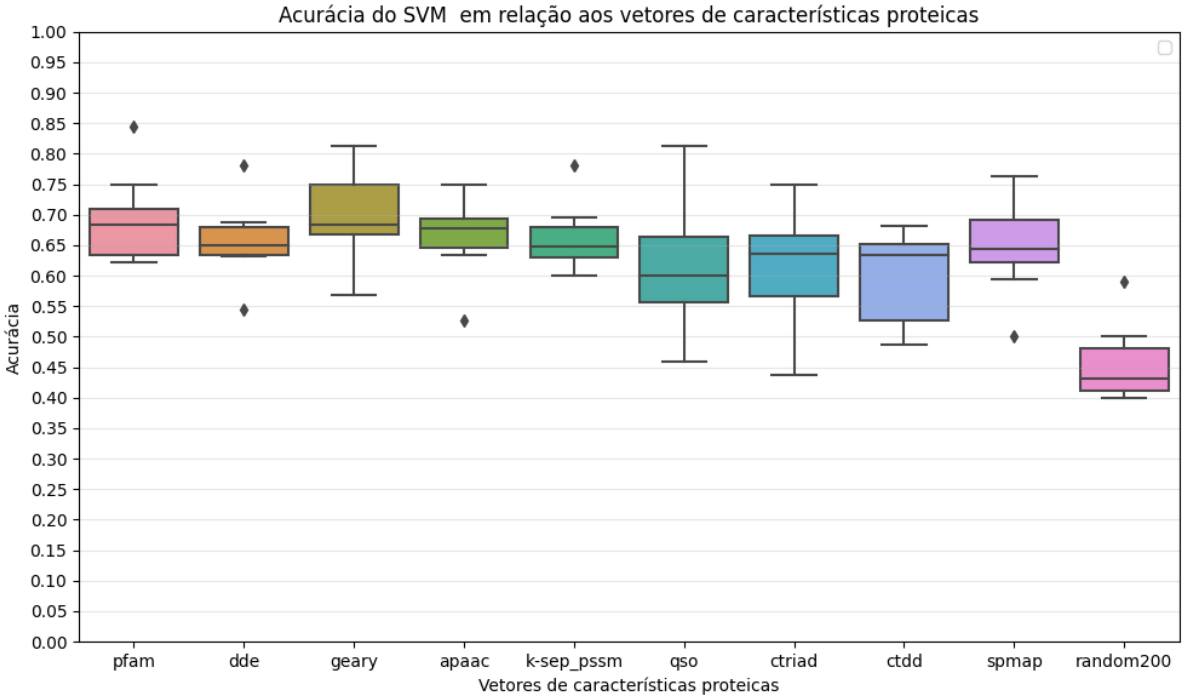

Figura 4.8: Acurácia do SVC em relação aos vetores de características proteicas

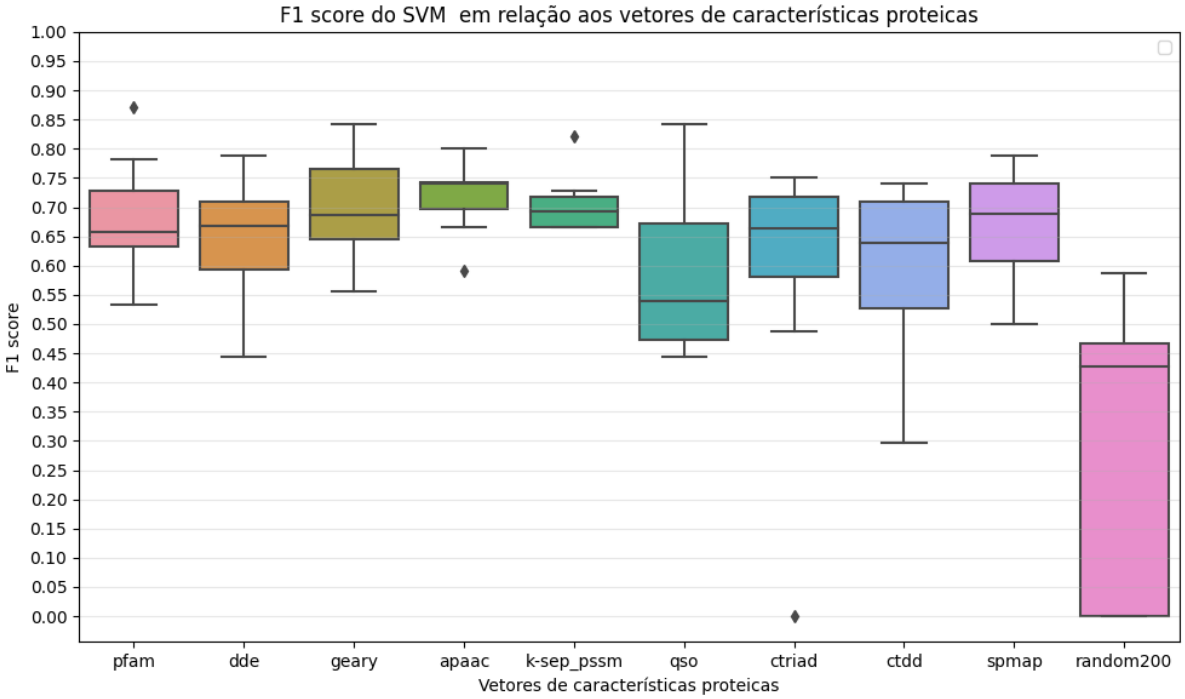

Figura 4.9: F1 score do SVC em relação aos vetores de características proteicas

Capítulo 5

Conclusão

A descoberta de medicamentos é um processo intrinsecamente complexo e de alto custo, com prazos médios de 14 anos e investimentos financeiros que podem ultrapassar 1 bilhão de dólares. A pesquisa conduzida neste trabalho propôs uma abordagem para otimizar este processo, utilizando vetores de características proteicas em conjunto com algoritmos de aprendizado de máquina. Este método tem o potencial de reduzir o tempo e os custos associados à identificação de proteínas em relação a reações com compostos alvo.

Após analisar os resultados obtidos pode ser observada uma tendencia de pareamento de vetores de representações proteicas e modelos de classificação, assim como afinidade entre composto alvo e seus reagentes com alguns vetores de características específicos podendo esses conterem de forma mais isolada um identificador da possibilidade de reação da proteína com o composto alvo.

Principais Resultados Os experimentos realizados demonstraram que diferentes combinações de modelos e vetores de características produzem resultados variados em termos de acurácia, precisão, F1 Score e recall. Em particular, os seguintes pares modelo-característica se destacaram:

XGBoost com CTriad: Alcançou uma acurácia média de 72,78%, com valores de F1 Score que variaram de 72,4% a 88,2%. Esses resultados indicam um desempenho robusto e consistente, evidenciando que esta combinação é eficaz na separação de dados reativos.

Random Forest com DDE: Embora tenha apresentado um desempenho médio de 71,08%, houve instâncias específicas, como o composto ChEMBL104, onde a acurácia ultrapassou 88%. Esse comportamento pode ser atribuído à capacidade do modelo de explorar padrões em conjuntos de dados menores.

KNN com K-SEP PSSM e SVM com Geary: Ambos apresentaram resultados sólidos, com acurácias de 70,31% e 69,51%, respectivamente. Esses modelos exploram bem características espaciais e padrões intrínsecos nos dados, como clusterização no caso do KNN e separação espacial como no caso do SVM 4.1.

5.1 Considerações Finais

A pesquisa demonstrou que a utilização de vetores de características proteicas, quando combinada com técnicas de aprendizado de máquina, pode prover uma boa eficiência do processo de descoberta de fármacos. Modelos como XGBoost e Random Forest mostraram-se particularmente eficazes, destacando-se em cenários com bases de dados menores e características fortemente separáveis sendo as principais recomendações para o contexto atual. Além disso, o desempenho variado entre diferentes modelos e representações indica que a escolha do par modelo-característica é crucial e deve ser orientada pela natureza específica do problema e dos dados.

5.2 Trabalhos Futuros

Os resultados dos experimentos apresentados abre a possibilidade de aprofundamento em trabalhos futuros em vertentes como:

- Aplicação de métricas de teoria da informação para extração de pares de modelo-representação com tendencias à uma melhora de performance por dispersão de compostos alvo.
- Obtenção de bases de dados com maior registro de reatividade entre compostos possibilitando a utilização de uma gama maior de modelos em sua performance basal mais consistente.
- Aplicar redução de dimensionalidade e extração de índices de correlação para compostos alvo, para constatar correlações maiores em pares específicos de compostos alvo e representações

Referências Bibliográficas

Atas Guvenilir, H. and Doğan, T. (2023). How to approach machine learning-based prediction of drug/compound–target interactions. *Journal of Cheminformatics*, 15(1):16.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Chen, X., Yan, C. C., Zhang, X., Zhang, X., Dai, F., Yin, J., and Zhang, Y. (2016). Drug–target interaction prediction: databases, web servers and computational models. *Briefings in bioinformatics*, 17(4):696–712.

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., Webb, G. I., Smith, A. I., Daly, R. J., Chou, K.-C., et al. (2018). ifeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, 34(14):2499–2502.

Chou, K.-C. (2000). Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochemical and biophysical research communications*, 278(2):477–483.

Chou, K.-C. (2005). Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics:2005.21.*

Consortium, T. U. (2020). Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489.

Cortes, C. and Vapnik, V. (1995). Support vector machine. *Machine learning*, 20(3):273–297.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Dubchak, I. (1999). Recognition of a protein fold in the context of the scop classification. *Proteins Struct Funct Genetics 35:401–407.*

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A., et al. (2019). The pfam protein families database in 2019. *Nucleic acids research*, 47(D1):D427–D432.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magariños, M. P., Overington, J. P., Papadatos, G., Smit, I., and Leach, A. R. (2016). The chembl database in 2017. *Nucleic Acids Research*, 45(D1):D945–D954.

Geary, R. C. (1954). The contiguity ratio and statistical mapping. *The incorporated statistician*, 5(3):115–146.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

LaValle, S. M., Branicky, M. S., and Lindemann, S. R. (2004). On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):673–692.

Li, Z.-R., Lin, H. H., Han, L., Jiang, L., Chen, X., and Chen, Y. Z. (2006). Profeat: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic acids research*, 34(suppl_2):W32–W37.

Lipman, D. J. and Pearson, W. R. (1985). Rapid and sensitive protein similarity searches. *Science*, 227(4693):1435–1441.

Liu, H., Sun, J., Guan, J., Zheng, J., and Zhou, S. (2015). Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics*, 31(12):i221–i229.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133.

Ong, S. A., Lin, H. H., Chen, Y. Z., Li, Z. R., and Cao, Z. (2007). Efficacy of different protein descriptors in predicting protein functional families. *Bmc Bioinformatics*, 8:1–14.

Ou-Yang, S.-s., Lu, J.-y., Kong, X.-q., Liang, Z.-j., Luo, C., and Jiang, H. (2012). Computational drug discovery. *Acta Pharmacologica Sinica*, 33(9):1131–1140.

pandas development team, T. (2020). pandas-dev/pandas: Pandas.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Prasetyo, V. P. and Anggraeni, W. (2024). Drug-target interactions prediction using stacking ensemble learning approach. In *2024 International Electronics Symposium (IES)*, pages 681–686. IEEE.

Saini, H., Raicar, G., Lal, S. P., Dehzangi, A., Imoto, S., and Sharma, A. (2016). Protein fold recognition using genetic algorithm optimized voting scheme and profile bigram. *Journal of Software*, 11(8):756–767.

Sarac, O. S., Gürsoy-Yüzügüllü, Ö., Cetin-Atalay, R., and Atalay, V. (2008). Subsequence-based feature map for protein function classification. *Computational biology and chemistry*, 32(2):122–130.

Saravanan, V. and Gautham, N. (2015). Harnessing computational biology for exact linear b-cell epitope prediction: A novel amino acid composition-based feature descriptor. *OMICS: A Journal of Integrative Biology*.

Schneider, G. and Wrede, P. (1994). The rational design of amino acid sequences by artificial neural networks and simulated molecular evolution: de novo design of an idealized leader peptidase cleavage site. *Biophysical Journal*, 66(2):335–344.

Shen J, Zhang J, L. X. (2007). Predicting protein–protein interactions based only on sequences information. *Proc Natl Acad Sci USA 104:4337–4341*.

Wang, J., Yang, B., Revote, J., Leier, A., Marquez-Lago, T. T., Webb, G., Song, J., Chou, K.-C., and Lithgow, T. (2017). Possum: a bioinformatics toolkit for generating numerical sequence feature descriptors based on pssm profiles. *Bioinformatics*, 33(17):2756–2758.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Yamanishi, Y., Pauwels, E., Saigo, H., and Stoven, V. (2011). Extracting sets of chemical substructures and protein domains governing drug-target interactions. *Journal of chemical information and modeling*, 51(5):1183–1194.