Instituto de Biociências

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de Botucatu

PG-BGA

UNIVERSIDADE ESTADUAL PAULISTA

"Júlio de Mesquita Filho"

INSTITUTO DE BIOCIÊNCIAS DE BOTUCATU

OTIMIZAÇÃO DO SEQUENCE SLIDER: UM MÉTODO DE ELUCIDAÇÃO DE ESTRUTURAS CRISTALOGRÁFICAS PROVENIENTES DE FONTES NATURAIS

Aluno: João Paulo Ballerini Bruno

Orientador: Prof. Titular Marcos Roberto de Mattos Fontes

Coorientador: Dr. Rafael Junqueira Borges

Dissertação apresentada ao Instituto de Biociências, Campus de Botucatu, UNESP, para obtenção do título de Mestre no Programa de Pós-Graduação em Biologia Geral e Aplicada, Área de concentração Biomoléculas: estrutura e função (BEF)

BOTUCATU - SP
2022

JOÃO PAULO BALLERINI BRUNO

Otimização do SEQUENCE SLIDER: um método de elucidação de estruturas cristalográficas provenientes de fontes naturais

Dissertação apresentada ao Instituto de Biociências, Campus de Botucatu, UNESP, para obtenção do título de Mestre no Programa de Pós-Graduação em Biologia Geral e Aplicada, Área de concentração Biomoléculas: estrutura e função (BEF)

Orientador: Prof. Titular Marcos Roberto de Mattos Fontes
Coorientador: Dr. Rafael Junqueira Borges

BOTUCATU - SP
2022

FICHA CATALOGRÁFICA ELABORADA PELA SEÇÃO TÉC. AQUIS. TRATAMENTO DA INFORM.
DIVISÃO TÉCNICA DE BIBLIOTECA E DOCUMENTAÇÃO - CÂMPUS DE BOTUCATU - UNESP
BIBLIOTECÁRIA RESPONSÁVEL: ROSEMEIRE APARECIDA VICENTE-CRB 8/5651

Bruno, João Paulo Ballerini.
Otimização do Sequence Slider : um método de elucidação de estruturas cristalográficas provenientes de fontes naturais / João Paulo Ballerini Bruno. - Botucatu, 2022

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Instituto de Biociências, Botucatu
Orientador: Marcos Roberto de Mattos Fontes
Coorientador: Rafael Junqueira Borges
Capes: 20804008

1. Envenenamento - Tratamento. 2. Aprendizado de máquina. 3. Toxicologia. 4. Sequence Slider. 5. Elucidação de estruturas.

Palavras-chave: Aprendizado de máquinas; Elucidação de estruturas; Sequence Slider; Toxinologia; XGBoost.

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Botucatu

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE JOÃO PAULO BALLERINI BRUNO, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM BIOLOGIA GERAL E APLICADA, DO INSTITUTO DE BIOCIÊNCIAS - CÂMPUS DE BOTUCATU.

Aos 21 dias do mês de outubro do ano de 2022, às 14:30 horas, no(a) Anfiteatro Minoru Sakate, realizou- se a defesa de DISSERTAÇÃO DE MESTRADO de JOÃO PAULO BALLERINI BRUNO, intitulada **Otimização do SEQUENCE SLIDER: um método de elucidação de estruturas cristalográficas provenientes de fontes naturais**. A Comissão Examinadora foi constituída pelos seguintes membros: Prof. Tit. MARCOS ROBERTO DE MATTOS FONTES (Orientador(a) - Participação Presencial) do(a) Departamento de Biofísica e Farmacologia / Instituto de Biociências de Botucatu - UNESP, Prof. Assoc. JOEL MESA HORMAZA (Participação Presencial) do(a) Departamento de Biofísica e Farmacologia / Instituto de Biociências de Botucatu - UNESP, Prof. Dr. ANGELO JOSÉ MAGRO (Participação Presencial) do(a) Departamento de Bioprocessos e Biotecnologia / Faculdade de Ciências Agronômicas de Botucatu - UNESP. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.

Prof. Dr. MARCOS ROBERTO DE MATTOS FONTES

AGRADECIMENTOS

- Ao meu pai **Paulo Sérgio Giannelli Bruno** e a minha mãe **Trézia Ieda Ballerini Bruno** por todo o suporte e apoio;

- Ao meu orientador, **Prof. Dr. Marcos Roberto de Mattos Fontes**, pela oportunidade e orientação;

- Ao meu coorientador, **Dr. Rafael Junqueira Borges**, pela atenção, paciência e pelos ensinamentos ministrados.

- Ao programa de Pós-graduação em Biologia Geral Aplicada, por possibilitar a realização deste trabalho.

- O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

RESUMO

A cristalografia desempenha papel essencial na elucidação dos mecanismos de ação de proteínas, por oferecer dados em nível atômico. Para elucidar a estrutura de uma macromolécula é fundamental o conhecimento da exata composição do seu cristal, o que geralmente é o caso de proteínas obtidas de forma recombinante. Porém, em diversas áreas de estudos, como na toxinologia, as amostras são geralmente obtidas através da purificação direta de fontes naturais, como por exemplo veneno de serpentes, onde propriedades físico-químicas semelhantes de isoformas podem dificultar seu isolamento. Na incapacidade de determinar uma única sequência em um cristal e na ausência de dados cristalográficos à resolução atômica, não existem métodos que auxiliem na elucidação destas estruturas *ab initio*. O método SEQUENCE SLIDER foi desenvolvido para avaliar diferentes possibilidades de cadeias laterais em um modelo cristalográfico no âmbito do faseamento no software ARCIMBOLDO e da incerteza da sequência na toxinologia. Nesta última finalidade, SLIDER integra dados de cristalografia, espectrometria de massa e análises filogenéticas. Assim, o objetivo deste trabalho foi otimizar SLIDER através da técnica de aprendizado de máquinas supervisionado *eXtreme Gradient Boosting* (XGBoost) sobre dados de análise de densidade eletrônica e do ambiente físico-químico de cada resíduo para estimar a atribuição do amino ácido correto. Foram utilizadas 41 estruturas cristalográficas de fosfolipases A_2, 15 de receptores de porina e 149 metaloproteases, obtidas de fonte recombinante cuja sequência é conhecida para treinamento e teste da metodologia. Resultados obtidos apresentam acurácia de 94.3% a 98.4% para 16.919 resíduos. É esperado que a aplicação deste método a dados inéditos provenientes de proteínas purificadas a partir de fontes naturais com sequência desconhecida possa melhor caracterizar seus componentes e, consequentemente, auxiliar na compreensão de seus mecanismos de ação e estratégias de inibição. SLIDER ainda poderá auxiliar outros cristalógrafos e biologistas estruturais ao ser disponibilizado à comunidade científica e, utilizado em diferentes sistemas biológicos obtidos de fontes naturais.

Palavras-chave: Aprendizado de Máquinas. SEQUENCE SLIDER. Toxinologia. XGBoost. Elucidação de estruturas.

ABSTRACT

Crystallography plays an essential role for the understanding of the action mechanisms of proteins, as it offers atomic resolution data. In order to elucidate the structure of a macromolecule, it is fundamental to know its exact crystal composition, which is usually the case for recombinant proteins. However, in several areas of study, such as toxinology, samples are usually obtained through direct purification from natural source, such as snake venom, where similar physico-chemical properties of the toxins can cause its isolation to be a challenge. Thus, in case of the inability to determine a single sequence in a crystal and in the absence of crystallographic data at atomic resolution, there are no methods for aiding *ab initio* elucidation of structures. The SEQUENCE SLIDER software was developed to evaluate different side chains possibilities for a crystallographic model in the scope of the ARCIMBOLDO phasing method and the sequences uncertainty in toxinology. In this last aim, SLIDER integrates crystallographic, mass spectrometry and phylogenetic data. Therefore, the goal of this work was to optimize SLIDER through application of the supervised machine learning *eXtreme Gradient Boosting* (XGBoost) with data from electron density and to physico-chemical environment analysis of each residue to estimate the correct amino acid assignment. Train and test data are composed of 41 crystallographic structures of phospholipases A_2, 15 porine receptors and 149 metaloproteases, obtained from recombinant source, whose sequence is known. Obtained results show accuracy ranging from 94.3% to 98.4% for 16.919 residues. It is expected that the application of the method to elucidate novel data from proteins purified from natural source with unknown sequence can better characterize their components and, consequently, aid action mechanisms comprehension and inhibition strategies developments. SLIDER may be able to assist other crystallographers and structural biologists as it will be available to the scientific community and, used for different biological systems whose source are natural.

Keywords: Machine Learning. SEQUENCE SLIDER. Toxinology. XGBoost. Structure Elucidation.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

PREÂMBULO

A cristalografia desempenha papel essencial na elucidação dos mecanismos de ação de proteínas ao revelar os detalhes de estruturas ao nível atômico, fator chave no entendimento de função e estratégias de inibição. Para se resolver a estrutura cristalográfica de uma macromolécula, é necessário conhecer sua composição, o que é usual ao considerar que a maior parte das proteínas são produzidas de maneira recombinante. Na toxinologia, as amostras geralmente são obtidas a partir de origem natural, isto é, do veneno bruto. Nos passos finais da purificação de venenos, amostras podem ser compostas por misturas de isoformas e/ou diferentes proteínas, já que as propriedades físico-químicas semelhantes dessas moléculas dificultam a separação. É possível cristalizar essas amostras, porém determinar a sequência na estrutura cristalográfica é desafiador na ausência de dados a resolução atômica. O desenvolvimento de métodos que auxiliem na elucidação de estruturas de maneira *ab initio* podem auxiliar cientistas da área de toxinologia a compreenderem melhor os mecanismos de ação das proteínas. Idealizado pelo coorientador desta dissertação, Dr. Rafael Junqueira Borges, SEQUENCE SLIDER aplicado a compostos naturais integra dados de cristalografia, espectrometria de massa e análises filogenéticas para avaliar diferentes hipóteses de cadeias laterais para cada posição de resíduo e assignar sequência (BORGES *et al.* 2022).

Esta dissertação almeja melhorar e complementar o método SEQUENCE SLIDER utilizando um algoritmo de aprendizado de máquinas denominado XGBoost, que calculará a probabilidade de assignação correta dentre 20 hipóteses de aminoácidos naturais para cada resíduo de uma proteína de interesse a partir de análise de densidade eletrônica e do entorno físico-químico. Esta implementação facilitará a distinção de diferentes hipóteses de cadeias laterais e conclusão das sequências mais prováveis.

A hipótese deste trabalho é que o algoritmo XGBoost pode conseguir diferenciar possibilidades de aminoácidos naturais corretas de erradas para resíduos de uma proteína de interesse utilizando informações acerca do ambiente físico-químico local e da densidade eletrônica de dados cristalográficos. A construção do modelo utilizando o algoritmo XGBoost foi otimizada utilizando métodos de validação cruzada para evitar gastos desnecessários de poder computacional sem prejudicar o desempenho de predição. Para analisar o desempenho do algoritmo frente a diferentes variáveis da densidade eletrônica e do ambiente físico-químico

local, foram feitas análises utilizando o índice de Youden e que maximizam a métrica f-score para propor diferentes limiares de assignação correta.

A Introdução desta dissertação apresenta as principais motivações teórico-práticas para a criação do método SEQUENCE SLIDER, utilizando como exemplo de possível aplicação, a área da toxinologia. Discorre sobre o método de cristalografia e como esse está relacionado diretamente ao método SLIDER. Exemplifica desafios recorrentes encontradas na área da toxinologia, além de destacar a importância da pesquisa científica nesta área. Explica a metodologia utilizada no cálculo das variáveis de origem físico-química por SLIDER para utilização no desenvolvimento do modelo de aprendizado de máquinas. Posteriormente, a Introdução apresenta a teoria acerca de técnicas de aprendizado de máquinas. Explica e exemplifica métodos supervisionados e não-supervisionados brevemente e o conceito de árvores de decisão, aplicando esses na apresentação de métodos de *ensemble learning*, com enfoque em técnicas de *boosting*. Apresenta XGBoost, o algoritmo utilizado para complementar a saída de SLIDER ao discorrer acerca de seus aspectos particulares, formulações matemáticas e funcionamento, e a metodologia utilizada em sua aplicação, fazendo uso de técnicas de validação cruzada e utilização de métricas na construção de modelos mais robustos iterativamente. Os resultados são analisados utilizando matrizes de confusão para limiares distintos de assignação e curvas ROC (*receiver operating characteristic*) e recall-precisão, com finalidade de avaliar o desempenho do modelo final. Exemplifica, por fim, a saída do algoritmo, discorrendo acerca de possíveis contribuições na elucidação de estruturas.

INTRODUÇÃO

1.1 Cristalografia de proteínas e o método SEQUENCE SLIDER

A cristalografia desempenha papel essencial na elucidação de mecanismos de ação de proteínas, por oferecer dados em nível atômico destas moléculas biológicas na forma nativa ou com mutações e também complexadas com cofatores, inibidores, ou com outras proteínas, com RNA ou DNA. A relevância desta técnica é ilustrada pelos avanços em várias áreas da bioquímica que repercutiram em dezenas de cientistas laureados com prêmios Nobel em virtude do desenvolvimento de pesquisas relacionadas ou baseadas na cristalografia de pequenos compostos inorgânicos ou de origem biológica (JASKOLSKI; DAUTER; WLODAWER, 2014).

O processo de elucidar uma estrutura cristalográfica de uma molécula se inicia em sua cristalização, cujo sucesso está diretamente relacionado à pureza e monodispersidade da amostra. O resultado final é um modelo atômico construído sobre um mapa de densidade eletrônica, sendo este último calculado com base nas intensidades e fases das reflexões provenientes da difração de raios X de cristais homogêneos. Contudo, neste experimento, apenas as intensidades são observadas, de maneira que recuperar as fases é o segundo maior desafio depois da cristalização. Experimentalmente, é possível obter as fases de uma estrutura por meio da introdução de átomos pesados nos cristais (Substituição Isomórfica Múltipla) ou alterando-se o comprimento de onda dos raios X para explorar o espalhamento anômalo de átomos específicos. Ao se obter as fases, que representam apenas parcela da estrutura total, é possível revelar o restante da estrutura utilizando algoritmos de modificação e interpretação de densidade eletrônica.

O conhecimento das estruturas de diversas proteínas permitiu o advento da Substituição Molecular, método mais utilizado para recuperar as fases. Este método consiste em utilizar as fases de um modelo de proteína homóloga que deveria ser semelhante ao da estrutura proteica desconhecida, já que a conservação de estrutura terciária é maior que a primária em proteínas. Nessa técnica, são realizadas duas buscas, uma rotação e outra translação. A amostragem do espaço tridimensional em conjunto com uma função pontuação baseada em máxima parcimônia encontra a solução correta. Com a otimização desta função de pontuação (considerando

resolução e erros dos dados e a fração de espalhamento e divergência esperada do modelo), com melhoria da precisão das medidas de difração de raios X e do poder computacional, surgiu a Substituição Molecular a partir de fragmentos. Nesta, fragmentos são localizados e, por se tratar de porção pequena da estrutura final, a função de pontuação não é suficiente para discriminar verdadeiro de falso positivos. A estratégia é submeter múltiplas possibilidades dos fragmentos aos algoritmos de modificação e interpretação de densidade eletrônica e, caso a subestrutura (fragmentos) esteja acuradamente posicionada, suas fases seriam suficientes para revelar o restante da proteína.

1.1.1 O método ARCIMBOLDO

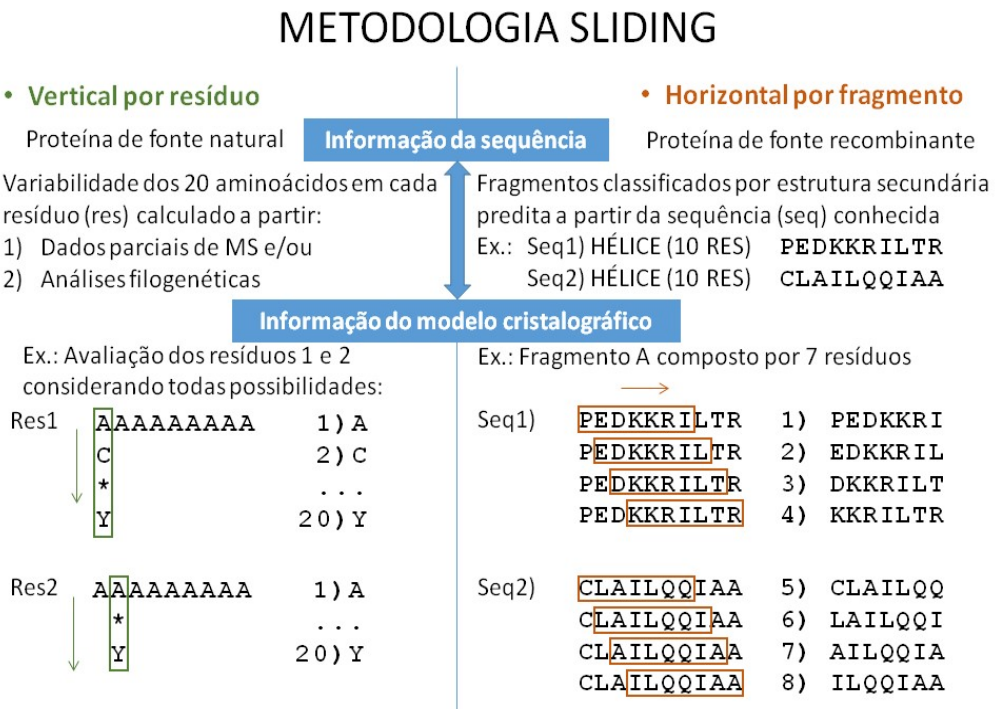
O método ARCIMBOLDO (SAMMITO et al., 2015) foi originalmente concebido como método de faseamento *ab initio*, utilizando a premissa que toda proteína tem em sua composição estrutural fragmentos de estruturas secundárias que se repetem, geralmente hélices α e folhas β constituídas por 8 a 20 resíduos de aminoácidos (MEDINA et al., 2020). A quantidade e as características dos elementos de estruturas secundárias podem ser preditas utilizando ferramentas de bioinformática a partir da sequência. ARCIMBOLDO reúne um conjunto de hipóteses através da localização do(s) fragmento(s) escolhido(s) com o programa de substituição molecular PHASER (MCCOY et al., 2007). Caso o posicionamento dos fragmentos tenha sido correto (desvio quadrático médio (RMSD) abaixo de 0,5), a modificação e interpretação de densidade eletrônica é realizado pelo programa SHELXE (USÓN; SHELDRICK, 2018), construindo-se, então, o restante do modelo.

O caráter *ab initio* do ARCIMBOLDO destaca sua importância como alternativa às demais técnicas de faseamento que requerem ou dados experimentais adicionais (Espalhamento anômalo ou Substituição Isomórfica Múltipla) ou disponibilidade de um modelo de uma proteína homóloga (Substituição Molecular). Porém, um pré-requisito para ARCIMBOLDO funcionar é que sejam localizados corretamente fragmentos totalizando pelo menos 10% dos átomos proteicos. Entretanto, estender o faseamento a dados de difração com resolução inferior a 2,5 Å e/ou mais de 300 resíduos na unidade assimétrica é desafiador, particularmente na interpretação da densidade.

1.1.2 O método SEQUENCE SLIDER

Incorporar cadeias laterais aos fragmentos inicialmente encontrados permite estender ARCIMBOLDO a resoluções mais baixas, já que os átomos presentes nos modelos estariam aumentando. A inspeção manual da densidade eletrônica não permite distinguir a que região pertence o fragmento da sequência, restando comprovar massivamente as combinações de cadeias laterais que a sequência conhecida permite. Para reduzir o número de combinações, foi proposto o método nomeado SEQUENCE SLIDER. A estratégia é restringir as hipóteses de sequências no fragmento coincidindo sua estrutura secundária com a predita a partir da sequência com o algoritmo PSIPRED (BUCHAN et al., 2013; MCGUFFIN; BRYSON; JONES, 2000). Trechos da sequência são deslizados sobre os fragmentos encontrados por PHASER ou traçados por SHELXE contendo mesma estrutura secundária (quadro direito da Figura 1.1). Com essa estratégia foram capazes de elucidar estruturas desconhecidas com resolução a 2.7 Å contendo 600 resíduos na unidade assimétrica (BORGES et al., 2020).

Figura 1.1 - Metodologia de deslizamento (*sliding*) vertical e horizontal do método SEQUENCE SLIDER



Fonte: Autor - Informações complementares provenientes de espectrometria de massa ou análise filogenética são relacionadas ao modelo cristalográfico para gerar hipóteses das diferentes sequências possíveis. Neste, a janela do *sliding* é vertical e cada resíduo é avaliado por vez. No caso ilustrado, 20 possibilidades são avaliadas para os resíduos 1 (Res1) e 2 (Res) de uma cadeia de polialanina. No caso de faseamento, a janela do *sliding* é horizontal e tem comprimento igual ao tamanho do fragmento avaliado (FragA), hipóteses são avaliadas por fragmento e são

compostas pelas sequências que foram preditas com mesma estrutura secundária do fragmento. No caso ilustrado, as hipóteses permitidas pelas sequências preditas como hélice (Seq 1 e 2) e pelo tamanho do fragmento presente no modelo cristalográfico (FragA) são 8.

1.2 Desafios na elucidação de proteínas purificadas de venenos

Na toxinologia as amostras de toxinas de serpentes são geralmente obtidas através da purificação direta do veneno, uma vez que a extração do mesmo é prática comum pela acessibilidade da glândula e pela grande quantidade armazenada. Caracterizadas por uma das mais rápidas divergências e variabilidades evolutivas em qualquer categoria de proteínas, as toxinas e suas diferentes isoformas compartilham propriedades físico-químicas entre si (CALVETE et al., 2009). Entre as PLA_2s por exemplo, foram identificadas e caracterizadas pelo menos 16 isoformas de β-bungarotoxina no veneno de *Bungarus multicinctus* e 15 isoformas de crotoxina no veneno de *Crotalus durissus terrificus* (DOLEY; KINI, 2009). A amoditoxina de *Vipera ammodytes* tiveram duas isoformas elucidadas por cristalografia e apenas suas duas mutações naturais (F113I e K117E) são suficientes para alterar sua toxicidade e atividade anticoagulante (SAUL et al., 2010).

Em virtude dessa complexidade, as amostras obtidas nos passos finais da purificação podem ser misturas de isoformas e/ou diferentes proteínas. Nesses casos, resultados funcionais e estruturais, incluindo espectrometria de massa, são provenientes de combinação de tais compostos. Na ausência de uma sequência conhecida e de dados cristalográficos a resolução atômica (valores nominais abaixo de 1 Å, que se trata de menos de 1% dos dados disponíveis no PDB (https://www.rcsb.org/stats/distribution_resolution)), poucos são os métodos cristalográficos que auxiliam a elucidação de estruturas com característica *ab initio*, e mesmo estes, apresentam baixa eficácia a resoluções comumente encontradas no PDB.

Modelos cristalográficos com alta confiabilidade são essenciais para servir de suporte à elucidação de mecanismos de ação e desenvolvimento de inibidores específicos que auxiliem na suplementação do soro antiofídico. Geralmente, dados provenientes de espectrometria de massa não alcançam coberturas de 100%, sendo o restante da sequência completado com o observado no banco de dados. Ao desprezar a complexidade destas isoformas, um viés é produzido à informação já disponível no banco de dados evitando introdução de novas sequências. Algumas PLA_2s, cujos dados cristalográficos são provenientes de amostras, não possuem sequência 100% conhecida, o que torna o método SLIDER, que propõe a avaliação de

diferentes possibilidades de cadeias laterais em fragmentos, imediatamente aplicável à incerteza da sequência nos casos de proteínas purificadas de fontes naturais.

1.3 Toxinologia

A toxinologia, o estudo dos venenos, é uma área de notoriedade recente. Descrito como coquetel de moléculas potentes e estáveis de fácil extração a partir de glândulas externas, o veneno apresenta um grande potencial farmacêutico. O exemplo clássico é o famoso Captopril®, medicamento para tratar a hipertensão, que foi sintetizado em laboratório a partir das estruturas de peptídeos purificados do veneno de serpentes *Bothrops jararaca* (E SILVA; BERALDO; ROSENFELD, 1949). Além da utilidade na medicina, a toxinologia auxilia descrição de processos fisiológicos, como a elucidação do mecanismo catalítico de fosfolipases A_2 (PLA_2s) a partir de toxinas de serpente e abelha (SCOTT et al., 1990a, 1990b), grupo de proteínas que também está envolvido na inflamação, apoptose e digestão. A capacidade de analisar amostras complexas quali e quantitativamente abrangeu a toxinologia com a proposição da venômica, conjunto de ferramentas ômicas especializadas em caracterizar venenos (CALVETE; JUÁREZ; SANZ, 2007). Contudo, a área ainda carece do sequenciamento do DNA de animais venenosos (KERKKAMP et al., 2016) e de antídotos capazes de neutralizar os efeitos decorrentes de envenenamento de animais peçonhentos. É desejável, portanto, o desenvolvimento de ferramentas que auxiliem pesquisadores a elucidar novas estruturas de toxinas, permitindo um melhor entendimento de processos fisiológicos nos quais essas participam.

1.3.1 A importância de estudar compostos obtidos de fontes naturais

Os acidentes ofídicos, foram classificados pela Organização Mundial de Saúde como uma doença tropical negligenciada (WORLD HEALTH ORGANIZATION, 2007). É estimado que ocorram aproximadamente 421.000 a 1.800.000 acidentes envolvendo envenenamento por serpentes em todo mundo, em que a taxa de letalidade é cerca de 5% e chances de sequelas físicas permanentes de até 15%. As regiões tropicais e subtropicais, por apresentarem clima ideal para serpentes, registram números de acidentes ofídicos mais elevados, com destaque para o sul e sudeste asiático, África subsaariana e centro e sul do continente americano (KASTURIRATNE et al., 2008).

No Brasil, a taxa de mortalidade é de apenas 0,36% (BRASIL, 2019b) em aproximadamente 28.000 mil casos por ano considerando os dados da última década (BRASIL, 2019a), o que pode ser atribuído à eficiência na produção e distribuição do único tratamento disponível, o soro antiofídico (KASTURIRATNE et al., 2008). Para que os soros antiofídicos tenham um amplo espectro de ação, sua produção é realizada a partir dos venenos das espécies de serpentes mais comuns de uma determinada região (KALIL; FAN, 2016). Fatores como distribuição geográfica, gênero, idade, temporada do ano e variação genética interferem nas características do veneno de serpentes de mesma espécie e, portanto, na complexidade para produção de soros de menor especificidade (KALIL; FAN, 2016).

O gênero *Bothrops* é responsável por cerca de 70% dos acidentes ofídicos no Brasil. Estes eventos apresentam complicações como distúrbios na coagulação, necrose do tecido muscular, hipotensão e disfunções renais, com risco proporcional ao atraso na aplicação do soro específico (BRASIL, 2010). Os danos locais musculares ainda não são neutralizados com eficácia pelo soro antiofídico e podem levar à amputação do membro. O quadro é agravado também pelo fato da grande maioria dos acidentes ofídicos ocorrerem na zona rural durante a execução de trabalhos manuais, fator que dificulta e retarda o acesso aos hospitais (BRASIL, 2010; GUTIÉRREZ; THEAKSTON; WARRELL, 2006).

Dois grupos de toxinas são responsáveis pela mionecrose nos acidentes botrópicos, as metaloproteases e as fosfolipases A_2 (PLA_2s – E.C. 3.1.1.4). A quantidade do último grupo pode variar em até 80% do total proteico do veneno dependendo da espécie (CALVETE et al., 2010; CALVETE; JUÁREZ; SANZ, 2007). As PLA_2s apresentam um amplo espectro de ação. *In vivo*, observa-se miotoxicidade local, edema, liberação de citoquinina, recrutamento de leucócitos, hiperalgesia, analgesia e inibição de crescimento tumoral (LOMONTE et al., 2009). Já nos experimentos *in vitro*, os efeitos tóxicos são vastos, tais como citotoxicidade em diferentes tipos de células humanas, ação bactericida, fungicida e antiparasitária, entre outras (LOMONTE et al., 2009). Os estudos dos efeitos farmacológicos *in vivo* são relevantes no envenenamento ofídico para desenvolvimento de estratégias de complementação do soro antiofídico. Outros estudos na área têm como objetivo auxiliar na compreensão da ação das toxinas e no desenvolvimento de medicamentos, já que os efeitos ocorrem em ambiente controlado diferente do natural (LOMONTE et al., 2009). Essa variada gama de efeitos farmacológicos atraíram a atenção dos pesquisadores devido ao potencial das PLA_2s contra doenças que acometem o ser humano (RODRIGUES et al., 2009).

1.4 Aplicação de SEQUENCE SLIDER à toxinologia

O método SEQUENCE SLIDER foi desenvolvido pelo Dr. Rafael Junqueira Borges, coorientador desta dissertação, com o objetivo de avaliar diferentes hipóteses de sequência a dados cristalográficos no âmbito do faseamento foi adaptado para toxinologia. Novas funções foram criadas para avaliar o ambiente físico-químico local e densidade eletrônica. Essas informações podem ajudar na formulação de hipóteses mais robustas de assignação de sequência, contribuindo na elucidação de uma estrutura de interesse.

1.4.1 Estruturas utilizadas

Foram utilizadas 205 estruturas de proteínas de fonte recombinante, retiradas do banco de dados: *Research Collaboratory for Structural Bioinformatics Protein Data Bank* - RCSB PDB (BERMAN, 2000), divididas em 41 PLA_2s, majoritariamente compostas por α-hélices, 15 receptores porina compostos por folhas-β e 149 metaloproteases, com diferentes estruturas secundárias, totalizando 67.683 resíduos. Os dados apresentam resoluções de 0,85 a 3,0 Å. As estruturas utilizadas estão elencadas no apêndice A.

1.4.2 Variáveis obtidas

Para o método SLIDER aplicado à toxinologia, o algoritmo avaliou as diferentes possibilidades de aminoácidos nas densidades eletrônicas e o entorno físico-químico. A presença de dados de espectrometria de massa, peptídeos sequenciados podem servir para restringir as possibilidades avaliadas por SLIDER (BORGES et al., 2022). Além de dados físico-químicos, análises filogenéticas em conjunto com cálculo da taxa de mutação pontual dos resíduos são outras estratégias possíveis para restringir as múltiplas hipóteses (BORGES et al., 2022).

1.4.2.1 Coeficiente de correlação no espaço real - RSCC

SLIDER avalia as diferentes hipóteses geradas através do cálculo do coeficiente de correlação no espaço real (RSCC – real space correlation coefficient) da densidade observada $\rho(r)_{obs}$ (dados experimentais) com a esperada $\rho(r)_{calc}$ (teórica a partir do modelo) conforme a Equação 1.1. Existem diferentes estratégias para calcular esse RSCC; a atualmente implementada no SLIDER é o POLDER MAP (LIEBSCHNER et al., 2017), que intensifica o

sinal omitindo determinado(s) átomo(s) escolhido(s) pelo usuário no cálculo das fases e evitando que essa região seja alvo da modelagem de solvente.

$$RSCC = \frac{\sum_r \left(\rho(r)_{obs} - \overline{\rho(r)_{obs}}\right) . \left(\rho(r)_{calc} - \overline{\rho(r)_{calc}}\right)}{\left(\sum_r (\rho(r)_{obs} - \overline{(r)_{obs}})^2 . \sum_r (\rho(r)_{calc} - \overline{\rho(r)_{calc}})^2\right)^{1/2}} \quad (1.1)$$

O RSCC é uma variável que fornece informação da densidade eletrônica. Foi calculado para a cadeia principal do aminoácido selecionando os mesmos átomos para diferentes aminoácidos, e para a lateral, que varia de acordo com a hipótese analisada. Se o valor do RSCC é considerado baixo, ou seja, a densidade eletrônica observada e esperada são distintas, as informações obtidas para um determinado resíduo podem ser ruidosas. Uma das razões que ocasionam baixa densidade eletrônica é a flexibilidade de resíduos em regiões da proteína expostas ao solvente.

1.4.2.2 Número de interações de contato

Para cada posição de um resíduo, foi utilizado o *software* PYMOL para gerar as demais moléculas por simetria (SCHRÖDINGER, LLC, 2015), excluindo átomos que estivessem a uma distância maior que 5.5 Å, para reduzir custo computacional e solvente a menos de 2.5 Å para remover impedimento estérico. Foram utilizadas, para cada resíduo, variáveis que representam o número de interações de contato entre átomos. Essas interações podem ocorrer a curtas distâncias; como ligações de hidrogênio (2.7 – 3.3 Å) estabelecendo ligação com átomos hidrofílicos, ou a longas distâncias; como interações hidrofóbicas (3.3 – 4.0 Å). Os números das seguintes interações de contato foram utilizados como variável:

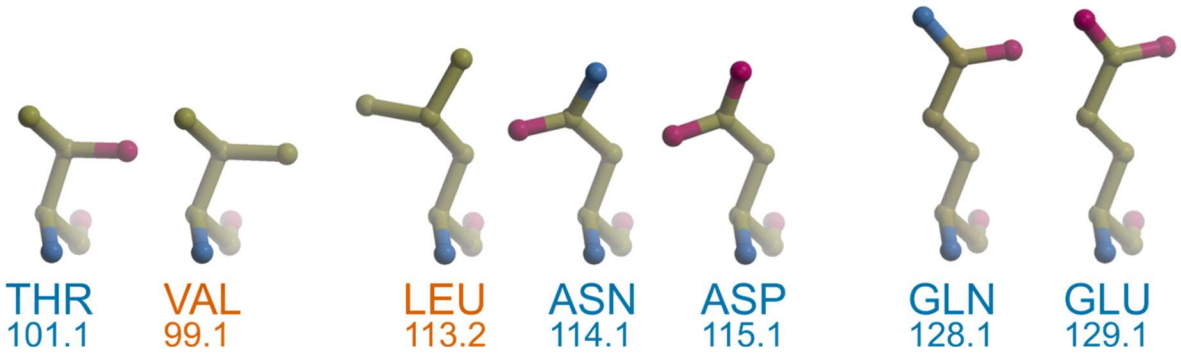
- Ligações de dissulfeto – nSSb – ligação covalente com importante função na formação da estrutura terciária proteica que ocorre entre aminoácidos cisteína.
- Ligações de hidrogênio – nHb – interação polar que ocorre entre átomos de hidrogênio eletropositivos e átomos eletronegativos como O^- e N^- com função de estabilização das estruturas secundárias das proteínas. Esta medida foi realizada com o software *HBPLUS* (MCDONALD; THORNTON, 1994), e apenas as ligações com átomos da cadeia lateral foram consideradas.
- Pontes salinas – nSalt – interação iônica e forte. Ocorre principalmente entre aminoácidos com cadeia lateral ácida (átomo eletronegativo) e básica (grupo amino de cadeia lateral). Foram contadas exclusivamente para interações entre

os últimos átomos de interações entre lisina/arginina e ácido aspártico/ácido glutâmico.

- Interações hidrofóbicas – nHydInt – interações que surgem entre moléculas de água e moléculas hidrofóbicas apolares. Quando moléculas hidrofóbicas estão em meio aquoso, as ligações de hidrogênio das moléculas de água de seu entorno serão quebradas para conceder espaço para as moléculas hidrofóbicas, gerando calor ao sistema. As moléculas de água então distorcidas, vão se rearranjar, formando novas ligações de hidrogênio, em estruturas envelopadas com menor entropia. Foram contados os átomos hidrofóbicos da proteína que interagiriam com os hidrofóbicos da cadeia lateral (HAGEMANS et al., 2015).

Comparar número de interações hidrofílicas e hidrofóbicas pode possibilitar a distinção entre aminoácidos aproximadamente isostéricos com densidade eletrônica semelhante, mais especificamente treonina da valina; e leucina do(a) ácido aspártico/asparagina. A utilização da presença de pontes salinas pode diferenciar a glutamina do ácido glutâmico. A figura 1.2 ilustra aminoácidos com densidade eletrônica semelhante.

Figura 1.2 – Aminoácidos semelhantes.



Fonte: Autor – A figura ilustra três casos de aminoácidos que possuem densidade eletrônica semelhante, possuindo apenas átomos diferentes. Podem ser diferenciados por sua massa atômica (valor indicado) e propriedades físico-químicas distintas da cadeia lateral.

1.4.2.3 Energia de ligação

Define-se energia de ligação como a quantidade de energia necessária para remover uma partícula de um determinado sistema. A energia de ligação de átomos da cadeia lateral com

demais átomos da proteína, ligantes e solventes próximos foi calculada segundo a Equação 1.2, onde nHb, $nSalt$, $nHydInt$ e $nSSb$ são o número de ligações de hidrogênio, número de pontes salinas, número de interações hidrofóbicas e número de ligações de dissulfeto. Os valores de energia por ligação foram extraídos do livro 'Biomolecular Crystallography' do autor Bernhard Rupp (RUPP, 2010, p. 50).

$$Energia = (nHb + nSalt) \times 3 + (nHydInt) \times 0.7 + (nSSb) \times 62 \ kcal/mol \quad (1.2)$$

1.4.2.4 Clashes

Para analisar colisões, foi avaliado apenas os átomos com distância inferior a 5.5 Å da cadeia lateral escolhida incluindo pares simétricos. Foram utilizadas duas variáveis extraídas do software PHENIX (ADAMS et al., 2010) e uma mensurando distâncias. Quanto maior o valor, maior são as colisões.

- Score de colisões – PhClSc – número obtido através de PHENIX utilizando a função phenix.clashscore.
- Superfície de Van der Waals – PhClSum – obtido da somatória da intersecção da superfície de Van der Waals de átomos da cadeia lateral com outros átomos.
- Somatória de scores de distância – SCldist – consiste na somatória da diferença entre 2.5 Å e a distância atômica entre átomos externos, representada pela Equação 1.3.

$$SCldist = \sum_{i=1}^{n} (2.5 - AtomicDist_i) \quad (1.3)$$

1.4.2.5 Rotâmeros

Foi realizada uma análise utilizando o software PHENIX (phenix.rotalyze) que valida os rotâmeros da cadeia lateral de aminoácidos (ADAMS et al., 2010). Os rotâmeros de cada hipótese de aminoácido de um resíduo de interesse foram classificados em favoráveis, permitidos ou não permitidos (*outliers*).

1.4.2.6 Profundidade do resíduo

A profundidade do resíduo é medida através de sua distância às regiões externas da proteína, com contato com solvente. Quanto maior o valor, mais profunda na proteína é a localização do resíduo de interesse, o que pode ajudar a diferenciar aminoácidos hidrofílicos de hidrofóbicos. Para este cálculo, foi utilizada a função *ResidueDepth* da biblioteca PDB de BioPython (COCK et al., 2009).

1.5 Dados obtidos

Os dados obtidos por esta modalidade de SLIDER são em formato LOG. Para cada proteína descrita na seção 1.3.1, cinco arquivos foram obtidos, contendo dados acerca do ambiente físico-químico do resíduo. Para facilitar, as abreviações de letra única dos 20 diferentes aminoácidos estão dispostas na tabela 1.1.

Tabela 1.1 – Os 20 aminoácidos principais

A	Ala	Alanina
C	Cis ou Cys	Cisteína
D	Asp	Aspartato (Ácido aspartico)
E	Glu	Glutamato (Ácido glutâmico)
F	Fen ou Phe	Fenilalanina
G	Gli ou Gly	Glicina
H	His	Histidina
I	Ile	Isoleucina
K	Lis ou Lys	Lisina
L	Leu	Leucina
M	Met	Metionina
N	Asn	Asparagina
P	Pro	Prolina
Q	Gln	Glutamina (Glutamida)
R	Arg	Arginina
S	Ser	Serina
T	Tre ou Thr	Treonina
V	Val	Valina
W	Trp	Triptofano (Triptofana)
Y	Tir ou Tyr	Tirosina

Apenas 20 aminoácidos encontrados em proteínas de fonte natural estão representados.

As tabelas 1.2, 1.3, 1.4, 1.5 e 1.6 ilustram a saída de SLIDER para cada resíduo para as diferentes variáveis que serão analisadas para cada hipótese de um mesmo resíduo.

Tabela 1.2 – Representação do arquivo final_model.log

Chain	ResN	PCCres	PCC	PCCdif%	
A	1	M!	87.7	6.6	No
A	1	Q	81.1	1.4	No
A	1	A	79.7	1.4	No
A	1	E	78.3	0.3	No
A	1	S	78.0	1.2	No
A	1	K	76.8	0.9	No
A	1	L	75.9	0.1	No
A	1	H	75.8	4.2	No
A	1	D	71.6	1.4	No
A	1	I	70.2	0.4	No
A	1	N	69.8	0.5	No
A	1	F	69.3	0.0	No
A	1	T	69.3	0.2	No
A	1	V	69.1	1.9	No
A	1	R	67.2	0.5	No
A	1	G	66.7	3.7	No
A	1	C	63.0	9.5	No
A	1	Y	53.5	6.4	No
A	1	W	47.1	5.4	No
A	1	P	41.7	last	No
A	1		96.2	BaselineMainChain	

Saída de SLIDER para variáveis relacionadas ao coeficiente de correlação de espaço real (RSCC), onde, Chain, ResN, PCCres, PCC e PCCdif% dizem respeito à cadeia, número do resíduo, hipótese de aminoácido, RSCC e diferença de contraste de RSCC, respectivamente. A última coluna da direita indica se a cadeia lateral é ou não excluída. Na última linha é mostrado o RSCC da cadeia principal, igual para todas as hipóteses de aminoácidos analisadas. Os dados estão arranjados de maneira decrescente utilizando o RSCC, onde a hipótese da primeira linha (Metionina - M) possui maior similaridade de densidade eletrônica com o resíduo analisado. A hipótese correta é mostrada pelo símbolo "!", dado o conhecimento da sequência correta dos dados utilizados (fonte recombinante).

Tabela 1.3 – Representação do arquivo clashes.log

Ch	ResN	ResT	SClDist	PhClSum	PhClSc
P	16	A	0.0	0.0	10.5
P	16	C	0.0	0.0	6.9
P	16	D	0.0	0.0	0.0
P	16	E	0.2	0.5	12.9
P	16	F	1.3	0.0	6.2
P	16	G	0.0	0.0	10.9
P	16	H	0.7	2.2	31.9
P	16	I	0.0	0.9	10.6
P	16	K	0.0	1.6	25.8
P	16	L	0.0	0.5	5.3
P	16	M	0.1	0.8	20.0
P	16	N	0.0	0.0	5.4
P	16	P	0.0	0.0	10.1
P	16	Q	0.0	0.6	4.4
P	16	R	7.2	0.0	5.9
P	16	S	0.0	0.0	0.0
P	16	T	0.0	0.0	0.0
P	16	V	0.0	0.0	0.0
P	16	W	1.7	0.0	6.1
P	16	Y	1.9	1.4	24.8

Saída de SLIDER para as variáveis relacionadas às colisões, Ch ResN, ResT, SCldist, PhClSum e PhClSc são abreviações para cadeia, número de resíduo, abreviação de uma letra do aminoácido, somatória de distância em colisão, somatória da intersecção da superfície de Van der Waals e score de colisões, respectivamente.

Tabela 1.4 – Representação do arquivo resdepth.log

Ch	ResN	ResDepth
A	1	3.4
A	2	4.3
A	3	2.0
A	4	2.0
A	5	4.6
A	6	3.3
A	7	2.0
A	8	3.5
A	9	5.2

Saída de SLIDER para a variável de profundidade de cada resíduo. A abreviação Ch, ResN e ResDepth representam a cadeia, o número do resíduo e a profundidade do resíduo em Å, respectivamente.

Tabela 1.5 – Representação do arquivo interactions.log

Ch	ResN	ResT	nSSb	nHb	nSalt	nHydInt	Energy
A	1	A	0	0	0	2	1.4
A	1	C	0	1	0	0	3.0
A	1	D	0	1	0	1	3.7
A	1	E	0	1	0	1	3.7
A	1	F	0	0	0	0	0.0
A	1	G	0	0	0	0	0.0
A	1	H	0	2	0	2	7.4
A	1	I	0	0	0	4	2.8
A	1	K	0	0	0	2	1.4
A	1	L	0	0	0	3	2.1
A	1	M	0	0	0	2	1.4
A	1	N	0	2	0	2	7.4
A	1	P	0	0	0	3	2.1
A	1	Q	0	1	0	2	4.4
A	1	R	0	0	0	2	1.4
A	1	S	0	1	0	0	3.0
A	1	T	0	1	0	0	3.0
A	1	V	0	0	0	3	2.1
A	1	W	0	0	0	0	0.0
A	1	Y	0	0	0	5	3.5

Saída de SLIDER para a variáveis relacionadas às interações, Ch ResN, ResT, nSSb, nHb, nSalt, nHydInt são abreviações para cadeia, número de resíduo, abreviação de uma letra do aminoácido, número de pontes de dissulfeto, número de ligações de hidrogênio, número de pontes salina, número de interações hidrofóbicas, respectivamente.

Tabela 1.6 – Representação do arquivo rotamers.log

Ch	ResN	C	D	E
A	1	Favored	Favored	OUTLIER
A	2	OUTLIER	Allowed	OUTLIER
A	3	Favored	OUTLIER	Allowed
A	4	Favored	Allowed	OUTLIER
A	5	Favored	Favored	OUTLIER
A	6	Favored	Favored	Allowed
A	7	Favored	Favored	Allowed
A	8	OUTLIER	Favored	OUTLIER
A	9	Favored	OUTLIER	OUTLIER

Saída de SLIDER para a análise de rotâmeros. Ch e ResN são abreviações para cadeia e número de resíduo. Apenas três hipóteses representadas pela abreviação de letra única dos vinte aminoácidos foram exemplificadas.

1.6 SLIDER e aprendizado de máquinas

Considerando que um grande volume de dados de estruturas de proteínas depositadas no banco de dados (PDB) possuem sequência conhecida, dado que foram obtidas de fonte recombinante, é direta a possibilidade de predizer tipo do aminoácido para cada resíduo utilizando técnicas de aprendizado de máquinas. Esses algoritmos são capazes de encontrar padrões e correlações em um grande volume de dados para criar um modelo capaz de predizer informações novas. Portanto, obter resultados satisfatórios utilizando um modelo de aprendizado de máquinas em dados de proteínas recombinantes disponíveis no PDB, a partir de dados obtidos por SLIDER, pode auxiliar na elucidação de estruturas provenientes de fontes naturais.

Os algoritmos de aprendizado de máquinas são eficientes na identificação de imagens e voz, funções de material genético não-codificante, de estruturas cristalinas a partir de imagens eletrônicas e padrões de difração sem orientação definida (AGUIAR et al., 2019).

1.7 Introdução ao aprendizado de máquinas

Técnicas de aprendizado de máquinas, que fazem parte do extenso ramo da inteligência artificial, estão cada vez mais sendo utilizadas para solucionar problemas encontrados no meio acadêmico e empresarial que, embora disponham de uma grande quantidade de dados, são dificilmente resolvidos por seres humanos. Alguns exemplos mais comuns de utilização dessas técnicas são no reconhecimento de padrões, na interpretação de textos e na detecção de objetos. (MOHAMMED, et al., 2016)

Modelos preditivos buscam encontrar, por meio de aprendizado utilizando um grande número observacional de dados, uma função f que melhor mapeia variáveis de entrada x em variáveis de saída y, de acordo com a Equação 2.1.

$$y = f(x) + e \tag{2.1}$$

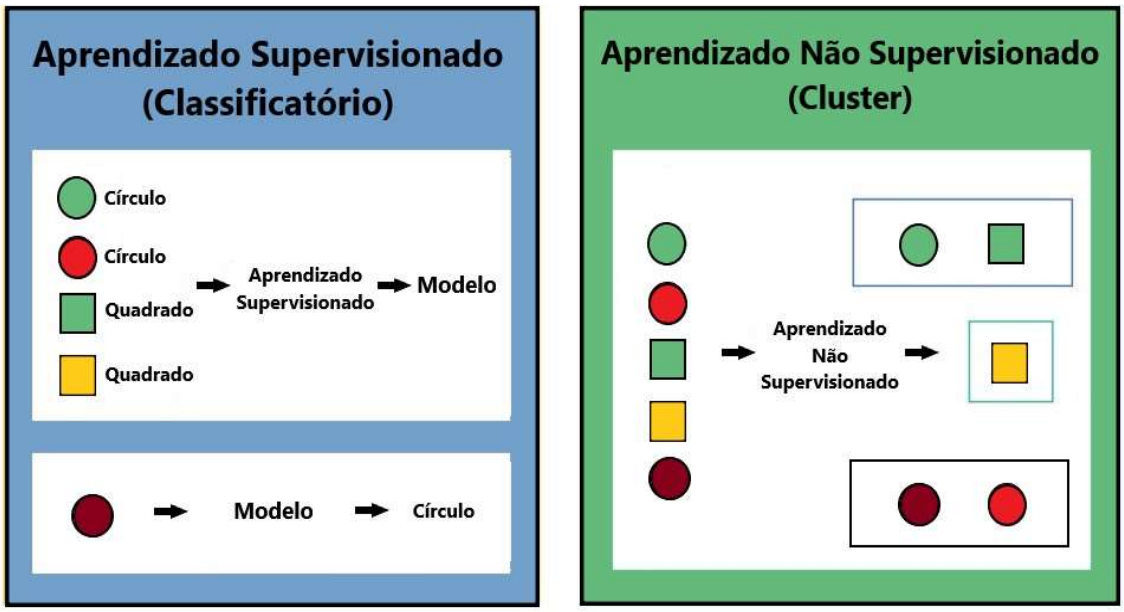
Embora seja desejável que a acurácia do modelo seja perfeita, é provável que exista um erro e que depende da escolha das variáveis de entrada, pois essas podem não ser suficientes para caracterizar um mapeamento perfeito da função f. Logo, grande parte das aplicações de técnicas de aprendizado de máquina consistem na busca de melhores escolhas e refinamento de variáveis de entrada x, com intuito de facilitar a busca pela melhor função de mapeamento

possível, que minimiza o erro e de um determinado modelo preditivo utilizado na resolução do problema de interesse.

1.7.1 Aprendizado supervisionado e não-supervisionado

Existem dois tipos principais de algoritmos de aprendizado de máquina: supervisionados e não-supervisionados. A Figura 1.3 diferencia ambos métodos com um exemplo com formas geométricas.

Figura 1.3 – Exemplos e diferenças ilustrativas entre algoritmos supervisionados e não-supervisionados.



Fonte: Autor - O modelo supervisionado no exemplo tem como variável de saída a forma geométrica; no não supervisionado, a forma geométrica pode ter sido considerada, mas as cores também foram utilizadas no processo de agrupamento dos dados.

Em algoritmos de aprendizado de máquina supervisionados, as variáveis de saída ou, informalmente, respostas procuradas, devem ser bem definidas. Isso possibilita que dados prévios sejam utilizados juntamente com suas respostas referentes já conhecidas para construir um modelo. Para aplicar um método supervisionado, deve-se realizar a separação dos dados do problema entre dois principais grupos: treino e teste. No primeiro, as respostas são evidenciadas juntamente com os dados, e no segundo, são ocultadas, reservando ao algoritmo, com base em

análises realizadas nos dados de treino, predizer a resposta correspondente a cada linha de informação. É importante pontuar que observações devem ser independentes, ou seja, não devem ser repetidas em ambos grupos na mesma análise. Um exemplo de algoritmo supervisionado pode ser uma regressão logística, que realiza previsões sobre uma variável discreta binária.

A principal característica de algoritmos não-supervisionados, ao contrário dos supervisionados, é que a saída não é uma variável definida. O modelo é criado sem a necessidade de dados de treino previamente categorizados. É recomendado quando há necessidade de se evidenciar padrões em um grupo de dados de interesse. Para exemplificar; é possível aplicar um *k-means,* método não-supervisionado, que agrupa os dados conforme suas características em comum.

Ao escolher um algoritmo de aprendizado de máquina, deve-se atentar às suas características, pois essas vão implicar na separação e organização dos dados do problema a ser resolvido.

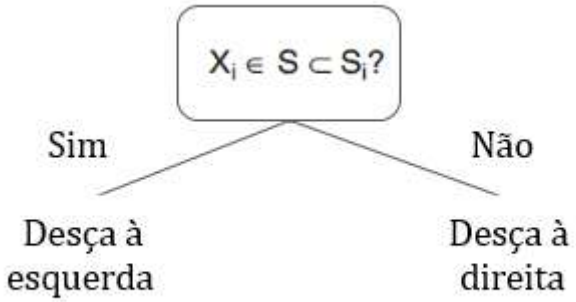
1.7.2 Árvores de decisão

Algoritmos de árvores de decisão são utilizados rotineiramente na construção de modelos de aprendizado de máquina, sendo a base de muitas técnicas poderosas de predição como florestas de decisão aleatória (*random forests),* e aceleração (*boosting)*.

Atualmente, utiliza-se árvores de regressão (CART), termo introduzido por Leo Breiman em 1984 (BREIMAN, et al., 2017) que, considerando métodos supervisionados, fornece relações entre um grupo de atributos (discretos ou contínuos) e uma classe contínua, ao contrário de árvores de decisão de classificação, onde a classe analisada é discreta.

A representação de um modelo construído utilizando uma CART condiz com uma estrutura de dados denominada árvore binária. Os nós representam a variável de entrada, e também pontos de separação com uma determinada condição. Quando não há ramificações adicionais, os nós terminais são chamados de folhas, e contêm variáveis de saída que são utilizadas para predizer algo acerca dos dados analisados. Na figura 2.2, é mostrado um ponto de separação (CUTLER et al., 2012).

Figura 1.4 – Representação da ramificação de uma árvore de decisão



Fonte: Adaptado de (CUTLER et al., 2012) – Considerando uma variável de predição categórica X_i, que possui valores de um conjunto finito de categorias $S_i = \{S_{i,1}, \dots, S_{i,m}\}$, a ramificação ocorre dada a condição $X_i \in S \subset S_i$. Para valores discretos ou categóricos, a árvore é denominada de classificação; para valores contínuos, tipicamente números reais, regressão.

As ramificações em árvores de regressão são definidas utilizando critérios de minimização. Considerando valores resultantes de um nó $y_1, \dots, y_k$, pode se usar como critério o erro quadrático médio residual

$$Q = \frac{1}{k}\sum_{i=1}^{k}(y_i - \bar{y})^2, \tag{2.2}$$

onde $\bar{y}$, é o valor predito no nó

$$\bar{y} = \frac{1}{k}\sum_{i=1}^{k} y_i. \tag{2.3}$$

A escolha é dada minimizando $Q_{sep} = k_E Q_E + k_D Q_D$, onde Q_E e Q_D são os resultados numéricos em um nó resultante da esquerda e direita, respectivamente, e k_E e k_D o tamanho das amostras, utilizando algoritmos com alta taxa de atualização (CUTLER et al., 2012).

1.7.3 Ensemble Learning

Algoritmos de *ensemble learning* (aprendizagem em comitês) utilizam da combinação de múltiplos modelos construídos previamente para melhorar o desempenho geral na predição de dados de um determinado problema de interesse. Combinar a informação presente de

modelos individuais tem como premissa o conceito de '*wisdom of crowds'* (sabedoria das multidões). Existem diversas técnicas utilizadas para agregar informação de múltiplos modelos.

1.7.3.1 Sabedoria das multidões

Uma tomada de decisão é algo recorrente na vida de um ser humano. Para que esta seja a melhor possível, um indivíduo racional poderia se basear nas opiniões distintas de outras pessoas previamente e optar pela mais observada. Para exemplificar: um indivíduo poderia, antes de realizar uma reserva em um hotel, avaliar os comentários de múltiplos clientes anteriores sobre as possíveis qualidades e defeitos do estabelecimento antes de consumar tal escolha. Essa abordagem, a uma tomada de decisão, com base em múltiplas decisões de nível inferior, é definida como sabedoria das multidões ou '*wisdom of crowds'*, que utiliza da opinião resultante de um agregado de grupos de pessoas ao invés da opinião de um único indivíduo. (SUROWIECKY, 2005)

A aplicação deste conceito é direta a algoritmos de *ensemble learning*. Construir um modelo robusto utilizando da informação agregada de múltiplos modelos menos eficientes, usualmente descritos como *'weak learners'*, é uma estratégia que apresenta melhores resultados em diversas áreas como segurança e detecção de fraudes (VANERIO, CASAS, 2017). As propriedades que fazem esse sistema ser mais eficiente, segundo o livro '*Pattern Classification Using Ensemble Methods*' são elencadas a seguir (ROKACH, 2010):

- Independência: as predições de modelos menores não podem ser influenciadas por outros modelos de mesma categoria, ou seja, as análises devem ser independentes;
- Descentralização: garantir que os modelos possam se especializar e, portanto, resolver um problema de maneira grosseira através de conclusões locais.
- Diversidade de opinião: a interpretação dos dados por um modelo menor, mesmo que crua, deve ser preservada;
- Mecanismos de junção: transformação da informação, anteriormente de caráter privado de um modelo, em uma decisão que pondera a individualidade de todos os modelos menores.

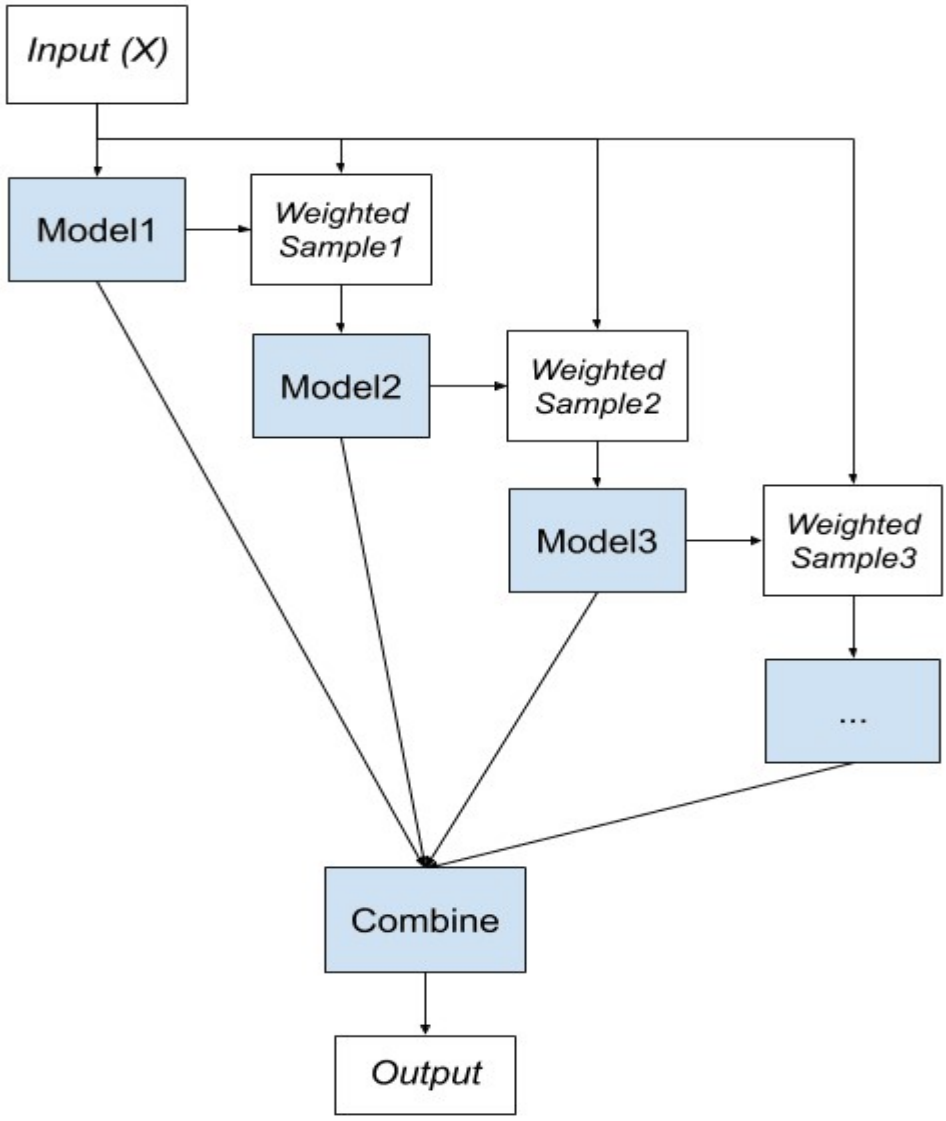
É possível utilizar de variações dessas propriedades para criar abordagens que condizem melhor com o problema a ser solucionado utilizando aprendizado de máquinas.

1.7.3.2 *Boosting*

A criação de novas abordagens na solução de problemas envolvendo métodos de *ensemble learning* é limitada apenas na criatividade do desenvolvedor em definir como os modelos menores vão compor o modelo principal (ZHANG, MA, 2021). Existem, portanto, várias estratégias para aplicar um algoritmo de *ensemble learning* que são utilizadas usualmente, como *bagging* (bolsas), *stacking* (empilhar), *boosting, bucket of models,* entre outras (ZHOU, 2019).

O *boosting,* é uma técnica de *ensemble learning* que utiliza da manipulação analítica do grupo de dados de treino em um modelo supervisionado para focalizar nos erros cometidos por *weak learners* com o propósito de corrigir essas imprecisões. As iterações são feitas sequencialmente, onde há enfoque na correção dos erros cometidos do primeiro modelo ao construir um segundo através de métodos como votação e utilização de um cálculo de média ponderada, com repetição deste processo até a construção de um modelo com melhor acurácia e desempenho. O enfoque pode ser dado utilizando diferentes pesos para observações (linhas de dados) que são mais escassas na predição de um resultado correto e em exemplos que podem ser mais difíceis de se prever. É mostrado, no fluxograma representado na Figura 1.5, a lógica por trás de um algoritmo de *boosting*.

Figura 1.5 – Fluxograma de um *Boosting Ensemble.*



Fonte: Adaptado de: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/

O primeiro algoritmo de *boosting* que se destacou em meio a outras técnicas de *ensemble learning* foi o *Adaptative Boosting,* abreviado em *AdaBoost,* onde os modelos são criados utilizando árvores de decisão (FREUND et al., 1997), demonstrando que a estratégia teórica do '*wisdom of crowds'* é utilizada de maneira eficiente na prática para resolver muitos problemas atuais. Ao utilizar algoritmos de *boosting,* deve-se haver precauções quanto à facilidade de ocorrência de *overfitting* (sobreajuste), que prejudica o desempenho do modelo na predição de novos dados, mesmo que os dados de treino tenham sido muito bem preditos. Métodos mais recentes, porém, possuem ferramentas mais eficientes para evitar esse problema recorrente, as

mais usuais são: utilização de parâmetros como '*learning rate'* (taxa de aprendizagem); controlar diretamente o número de iterações para obter o modelo final; e utilização de métodos de regularização.

Utilizando do conceito inicial de *AdaBoost,* Friedman (2001) generalizou o algoritmo, o denominando *Gradient Boosting Machine*, onde modificou a etapa de junção ao utilizar uma estratégia de estágios ao invés de uma sequencial pura; exemplificando: significa que todos os *weak learners* são preservados, reforçando a ideia de independência dos modelos ao invés de interagirem entre si a cada interação na construção do modelo final. Com a utilização de uma otimização numérica, com objetivo de minimizar a perda de informação desses modelos, por método do gradiente, os algoritmos de *boosting* mais modernos podem ser utilizados em problemas de classificação de classes múltiplas, com possibilidade de suporte a regressão, entre outras vantagens, em relação a seus predecessores (TANHA, 2020).

Os conceitos principais do método de *gradient boosting* são, portanto: a criação de modelos do tipo *weak learners*, construídos a partir de árvores de decisão; uma função de perda genérica e, por fim; a construção de um modelo de junção, que possibilita a adição de árvores com independência, garantindo a individualidade da informação dos modelos, possibilitada pela aplicação do método do gradiente. Para garantir que um determinado modelo minimize o erro a cada interação, este é parametrizado e modificado utilizando parâmetros usuais de árvores de decisão, sempre em direção ao mínimo local.

1.7.4 *eXtreme Gradient Boosting* – XGBoost

O método *eXtreme Gradient Boosting* (XGBoost) (CHEN, GUESTRIN, 2016) é um algoritmo *open source* (código aberto, disponível em https://github.com/dmlc/xgboost/) de aprendizado de máquina construído com base em árvores de decisão, utilizando técnicas de aumento de gradiente – *gradient boosting.* Desenvolvido para ser altamente flexível e eficiente, possibilita resolução de diversos problemas encontrados atualmente na ciência e no meio privado, apresentando resultados de última geração em competições atuais de resolução de problemas por métodos de aprendizado de máquina (NIELSEN, 2016).

1.7.4.1 Aspectos teóricos

O algoritmo XGBoost (CHEN, GUESTRIN, 2016), é considerado um *gradient boosting* regularizado por utilizar dos conceitos de otimização por funções de custo, além de diversas

técnicas de regularização. A função objetiva do algoritmo, que se deseja minimizar, em uma interação k, é dada por

$$\mathcal{L}^{(k)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i{}^{(k-1)} + f_k(\chi_i)\right) + \Omega(f_k) \tag{2.4}$$

onde l é uma função de árvores de regressão, $\hat{y}_i{}^{(k)}$ é a predição de uma instância i-nésima e $\Omega(f_k)$ é um termo de regularização. Ao constatar que a função objetiva de XGBoost é uma função com parâmetros de funções, deve-se destacar a impossibilidade de utilização de métodos tradicionais de otimização no espaço euclidiano de maneira imediata. Uma aproximação por séries de Taylor de segunda ordem para transformar a função objetiva original foi utilizada para possibilitar uso de técnicas de otimização;

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 \tag{2.5}$$

$$\mathcal{L}^{(k)} \cong \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}^{(k-1)}\right) + g_i f_k(\chi_i) + \frac{1}{2}h_i f_k^2(\chi_i)\right] + \Omega(f_k) \tag{2.6}$$

$$\tilde{\mathcal{L}}^{(k)} = \sum_{i=1}^{n}\left[g_i f_k(\chi_i) + \frac{1}{2}h_i f_k^2(\chi_i)\right] + \Omega(f_k) \tag{2.7}$$

onde (2.7) é a função objetiva em uma interação k e g_i e h_i são estatísticas do gradiente de primeira e segunda ordem da função de custo. É possível expandir $\Omega(f_k)$ ao definir uma coleção ordenada de instâncias $I_j = \{i|q(\chi_i) = j\}$ de uma folha j;

$$\Omega(f_k) = \gamma K + \frac{1}{2}\lambda\sum_{j=1}^{K} w_j^2 \tag{2.8}$$

$$\tilde{\mathcal{L}}^{(k)} = \sum_{j=1}^{K}\left[\left(\sum_{i\,\in\, I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\,\in\, I_j} h_i + \lambda\right)w_j^2\right] + \gamma K \tag{2.9}$$

Para uma estrutura fixa de uma árvore $q(\chi)$, pode-se computar o peso ideal de uma folha j: w_j^*;

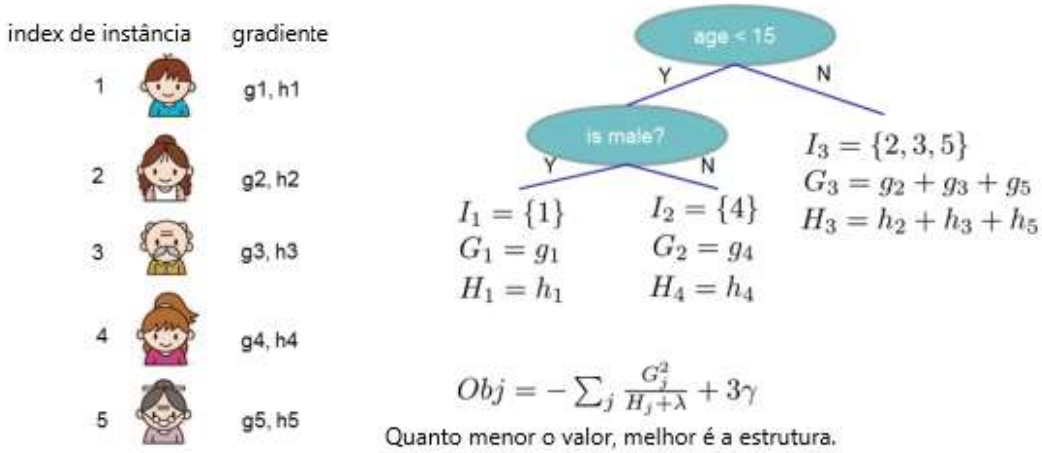
$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \ , \tag{2.10}$$

utilizado para calcular o valor ideal correspondente pela função objetiva;

$$\tilde{\mathcal{L}}^{(k)}(q) = -\frac{1}{2}\sum_{j=1}^{K} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma K \tag{2.11}$$

que por sua vez é utilizada como avaliadora da qualidade de uma árvore q; avaliação exemplificada na Figura 1.6.

Figura 1.6 – Cálculo da qualidade de uma árvore de decisão



Fonte: Adaptado/Traduzido de (CHEN, GUESTRIN, 2016) - Utilizar o valor da somatória das estatísticas dos gradientes de primeira e segunda ordem em cada folha na fórmula de avaliação representada resulta em um avaliador de qualidade da estrutura.

É utilizado um algoritmo ambicioso exato, que itera sobre todos as possíveis condições de ramificação, escolhendo, de maneira eficiente, melhores escolhas em múltiplos estágios menores para se obter um melhor resultado global. O algoritmo utiliza, dado que I_E e I_D são

instâncias dos nós esquerdo e direito após a ramificação, respectivamente, e considerando $I = I_E \cup I_D$, a fórmula;

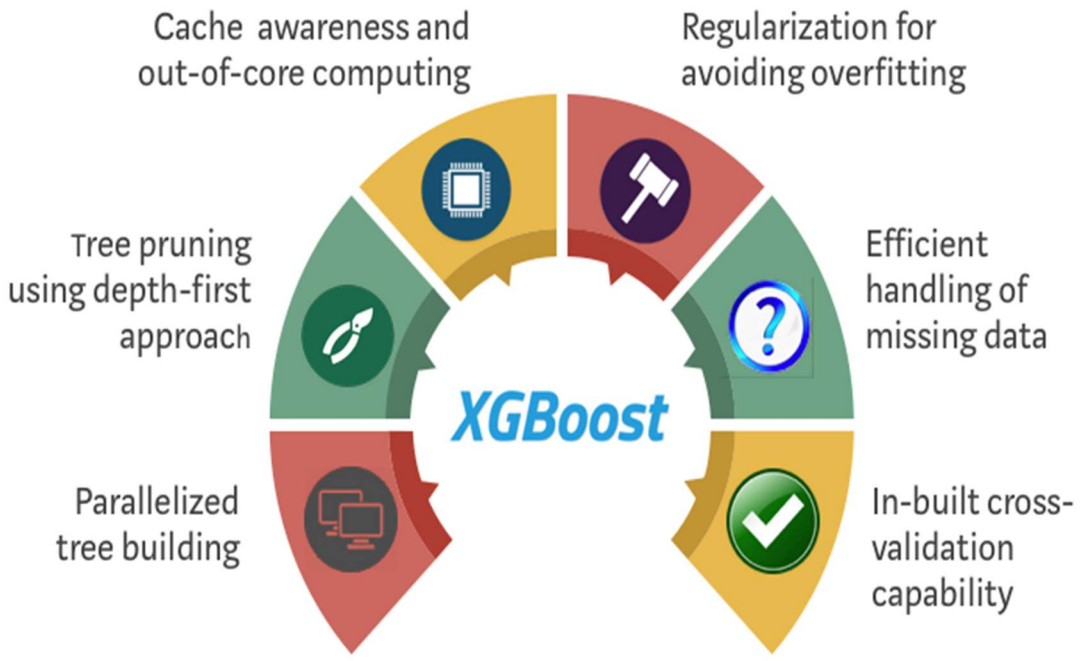
$$\mathcal{L}_{ram} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_E} g_i\right)^2}{\sum_{i \in I_E} h_i + \lambda} + \frac{\left(\sum_{i \in I_D} g_i\right)^2}{\sum_{i \in I_D} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \qquad (2.12)$$

para encontrar a melhor ramificação de maneira ambiciosa.

1.7.4.2 Recursos de XGBoost

Nos últimos anos, diversas bibliotecas foram construídas para várias linguagens de programação como Python, R, Julia, entre outras. XGBoost apresenta inúmeros recursos desenvolvidos para auxiliar nas dificuldades rotineiramente encontradas na aplicação de métodos de aprendizado de máquina. Alguns deles são exemplificados na figura 1.7 e sintetizados a seguir:

Figura 1.7 – Recursos e vantagens de XGBoost.



Fonte: http://dataanalyticsedge.com/2019/11/23/xgboost-using-python/ - acesso mais recente em 12/11/2021

1. Construção de árvores de maneira paralelizada ou aprendizagem paralelizada: Uso de todos os núcleos da CPU durante a fase de treinamento dos dados em virtude da

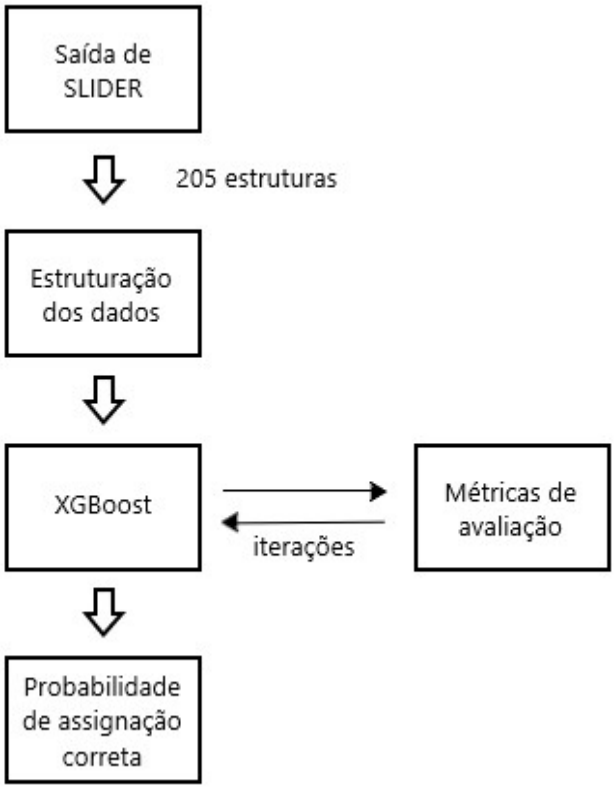
utilização de uma estrutura em forma de blocos organizados com base nas diferentes variáveis do problema em questão.

2. Poda da árvore por profundidade: Há opção de construir as árvores de decisão com profundidade máxima definida com base na escolha do hiperparâmetro *'max_depth',* o que concede velocidade na fase de treinamento. Após a construção das árvores de decisão, o algoritmo realiza exclusão de nós com parâmetros de regularização não-críticos ou redundantes para a classificação correta. A abordagem de poda de XGBoost consiste em começar pelo nó de maior profundidade, dependendo de hiperparâmetros definidos pelo desenvolvedor, que podem deixar o algoritmo mais ou menos conservador ao realizar podas.
3. Cache-consciente e cálculo *out-of-core*: Implementação de um algoritmo para lidar com problemas relacionados à alocação de memória de maneira indireta resultante da estrutura de blocos característica do algoritmo. Essa estrutura permite um sistema de alocação de uso do disco ao invés de memória temporariamente, minimizando utilização do segundo.
4. Controle de *overfitting*: Observado na Equação 2.4, o termo adicional de regularização auxilia a penalizar a complexidade de modelos (*weak learners*), ponderando melhor o peso de dados na construção de um modelo definitivo. Além disso, vários hiperparâmetros de controle são disponibilizados para uso que contribuem para controlar o *overfitting*, preocupação recorrente em técnicas de *boosting*.
5. Alocação consciente de dados faltantes: O algoritmo é capaz de interpretar facilmente agrupamentos de dados nulos nos dados de maneira otimizada.
6. Validação cruzada embutida: Funções disponíveis que concedem suporte à validação cruzada como xgboost.cv().

MATERIAIS E MÉTODOS

Para que os dados de saída do método SEQUENCE SLIDER, que consistem em informações do ambiente físico-químico, sejam utilizados na aplicação de um método de aprendizado de máquinas, foi realizada uma reestruturação dos dados utilizando uma série de ferramentas descritas nos itens a seguir, juntando todas as variáveis em um único arquivo no formato CSV (valores separados por vírgulas) e aplicando transformações (*one-hot-encoding,* estratificação dos grupos de treino e teste) que vão auxiliar o desempenho do aprendizado de máquina. Utilizou-se XGBoost para construir um modelo a partir dos dados estruturados e transformados. Métricas foram selecionadas para avaliar o desempenho geral do algoritmo, e comparar diferentes iterações e modelos. A figura 2.1, a seguir, ilustra a metodologia proposta.

Figura 2.1 – Metodologia proposta.



Fonte: Autor.

2.1 Ferramentas Utilizadas

Utilizou-se a linguagem de programação *Python* na construção dos algoritmos de reestruturação e preparação dos dados e aplicação do modelo de aprendizado de máquina. É uma linguagem de programação interpretada, orientada a objetos e de alto nível. Além de fácil compreensão e utilização, *Python* possui uma vasta quantidade de módulos e pacotes disponibilizados sem custo e de distribuição livre (ROSSUM, DRAKE, 2009).

2.1.1 *Numpy*

Fez-se uso da biblioteca *Numpy* para a realização de algumas operações de maneira direta, com a obtenção do índice de maior valor em um determinado eixo através de *argmax* (HARRIS, et al., 2020).

2.1.2 *Pandas*

Foi utilizada a biblioteca *Pandas* (MCKINNEY, 2010) para a conversão e realização de manipulação dos dados em formato estruturado. A formatação dos dados em *dataframes* para estruturar os dados com padrão bem definido contribui para o desempenho de alguns algoritmos de aprendizado de máquina, e é diretamente convertido em arquivos do formato CSV. Além do mais, *Pandas* possui muitos recursos disponíveis para manipular dados de maneira rápida e eficiente; os principais recursos utilizados foram:

- Capacidade de juntar diferentes arquivos CSV, transformados em *dataframes* em um único arquivo utilizando .merge();
- Localizar informações relevantes nos arquivos de saída de SLIDER que podem ser utilizados como variáveis independentes que podem contribuir no desempenho do algoritmo de aprendizado de máquina. Foi utilizado o atributo .loc para filtrar e reestruturar algumas variáveis;
- Ajustes em organização e nos nomes das colunas do *dataframe*, facilitando entendimento e padronizando as variáveis na saída do algoritmo.

2.1.3 *Matplotlib* e *Seaborn*

Para construir curvas de interesse e matrizes de confusão para avaliar o desempenho do algoritmo foi utilizado duas bibliotecas: *Matplotlib* e *Seaborn* que se especializam na criação e visualização de dados (HUNTER, 2007). Ambas foram criadas para a linguagem de programação *Python*. *Seaborn* é uma extensão a *Matplotlib* com intuito de melhorar e expandir a qualidade da interface gráfica e recursos disponíveis (WASKOM, 2021).

2.1.4 *Scikit-Learn*

Scikit-learn é uma biblioteca, de código aberto, para a linguagem de programação *Python*, que oferece suporte à métodos de aprendizado de máquina (PEDREGOSA, 2011). Destaca-se o suporte à classificação, regressão, agrupamento (clusters), redução de dimensão, seleção de modelos finais e pré-processamento de dados. Foi utilizada, em conjunto com a biblioteca de XGBoost, na construção de modelos diversos neste trabalho, através de múltiplas iterações que combinam funcionalidades de ambas bibliotecas.

2.2 Estruturação dos dados

Foram utilizados como dados a informação de 205 proteínas de origem recombinante com sequência conhecida, extraídos do PDB e analisados por SEQUENCE SLIDER para se obter variáveis do entorno físico-químico, com detalhes descritos no capítulo anterior.

Antes de se iniciar a aplicação do aprendizado de máquinas, os dados de diversas análises de SLIDER foram agregados em um único arquivo. Utilizou-se o arquivo de formato LOG de saída de SLIDER denominado final_model.log (RSCC; tabela 1.1) para realizar as padronizações necessárias antes de agregar as informações. O perfil de saída de SLIDER está exemplificado na seção 1.5 do capítulo anterior.

Primeiramente, o arquivo foi aberto no formato CSV utilizando pandas.read_csv(path) e depois da criação de novos índices, foi salvo utilizando pandas.DataFrame.to_csv(path). As designações das variáveis foram padronizadas, adicionando para a coluna localizada na extrema direita o nome da variável em questão; cadeia lateral excluída (ESCh). Criou-se duas outras colunas: uma com dados referentes ao RSCC (PCC) da cadeia principal do resíduo, informação disponível na última linha da tabela, que é comum entre todas as hipóteses de aminoácidos para um mesmo resíduo; outra com a informação da assignação correta do resíduo em questão,

representada na tabela pelo símbolo ‘!’ na coluna dos aminoácidos (hipóteses). Essa última informação é crucial na etapa de treino do algoritmo de aprendizado de máquinas, sendo ocultada nos testes, para ser, justamente, predita.

Todos os outros arquivos que são saída de SLIDER com dados acerca de colisões, interações, rotâmeros e profundidade de resíduo foram agrupados ao arquivo com as modificações acima utilizando pandas.DataFrame.merge(), fazendo uso de parâmetros condizentes para cada caso. O objetivo foi, portanto, agrupar todos os dados em um único arquivo de maneira estruturada, onde para cada resíduo, há informação de todas as variáveis de interesse para cada provável hipótese de aminoácido.

Foi construído um script simples para iterar sobre as 205 estruturas, com total de 67.683 resíduos, para se obter um arquivo único de análise de 109MB; a entrada crua, mas estruturada, que será utilizada na aplicação do algoritmo XGBoost.

2.3 Aplicação de XGBoost

A aplicação de XGBoost foi feita utilizando sua API em Python. Antes da construção do modelo, foram necessárias transformações no grupo de dados. As duas principais utilizadas foram: a utilização de *one-hot-encoding* e separação dos dados entre treino e teste.

A transformação denominada *one-hot-encoding* auxilia na melhor alocação de memória, possibilitando uma variável categórica ser representada de forma binária ao criar colunas adicionais com valores ‘0’ ou ‘1’. As variáveis ‘dumificadas’ anteriormente à aplicação de XGBoost, recorrentemente denominadas *‘dummies’,* foram: as diferentes categorias de rotâmeros; favoráveis, permitidos ou *outliers,* e as vinte hipóteses de diferentes aminoácidos de origem natural.

XGBoost é um método de aprendizado de máquinas supervisionado. Para ser aplicado, é necessário separar os dados em grupos de treino e teste. A proporção utilizada foi de 3:1. Os dados foram separados utilizando a função *GroupShuffleSplit()* da biblioteca *scikit-learn*. Utilizando uma variável de mapeamento, os dados foram estratificados, separando as informações disponíveis por resíduos e não por hipóteses de aminoácidos individuais. Essa estratégia permite que se mantenha a mesma proporção de verdadeiros negativos e positivos nos grupos de treino e teste; o que condiz com o perfil de entrada dos dados de proteínas individuais, sejam elas de sequência conhecida com intuito de validação ou desconhecida para elucidação.

O agrupamento dos dados e a construção do modelo foram feitos utilizando um desktop com Intel Core i7-6700K CPU@4.00GHz, NVIDIA GeForce GTX 1080Ti e 32GB de memória RAM. Os parâmetros *'n_estimators'* e *'learning_rate'* iniciais definidos foram 2000 e 0.01 respectivamente. XGBoost oferece suporte à construção das árvores de decisão utilizando a placa gráfica através do parâmetro '*gpu_hist'*. Hiperparâmetros são atributos utilizados para controlar o treinamento de um modelo de aprendizado de máquinas, ajudando na fidelidade do mesmo para com o problema real. Foram escolhidos para análise utilizando validação cruzada: *'max_depth'*, *'gamma'*, *'reg_lambda, 'learning_rate',* resultantes de um procedimento de validação cruzada. O hiperparâmetro *'scale_pos_weight = 19'* foi definido seguindo recomendação disponível na documentação de XGBoost, utilizando a divisão da soma das instâncias negativas pela soma das positivas (Disponível em: https://xgboost.readthedocs.io/en/stable/). Os demais hiperparâmetros não citados podem ser encontrados na documentação e foram definidos de acordo com o padrão recomendado.

2.4 Métricas de Avaliação

Métricas de avaliação são fatores decisivos na construção de qualquer modelo de aprendizado de máquinas, pois através delas, é possível obter informações que possibilitam a construção de iterações com melhores desempenhos.

Utilizou-se as seguintes métricas para avaliar o desempenho do algoritmo iterativamente:

- Validação cruzada pela função *cv*() da biblioteca de XGBoost em conjunto com curvas de aprendizagem para investigar o comportamento do modelo, como por exemplo averiguar tendência de *overfitting/underfitting*;
- Validação cruzada das variáveis utilizando uma função da biblioteca de XGBoost, *plot_importance*(), com o propósito de verificar qual influência exercem, na acurácia do modelo, as variáveis utilizadas. Eliminar informação que agrega pouco ou até mesmo prejudica o modelo contribui para uma melhor otimização com aumento de desempenho;
- Hiperparâmetros de XGBoost foram obtidos por meio de validação cruzada, com auxílio da biblioteca *Hyperopt* (BERGSTRA, et al., 2013);
- Com o auxílio da curva ROC (*Receiver Operating Characteristic*) e da curva de recall-precisão, foram utilizados diferentes limiares de assignação para analisar o comportamento do modelo. Um valor de maior desempenho foi obtido

utilizando o índice de Youden, a partir da curva ROC (FLUSS, et al., 2005). Um segundo, obtido ao maximizar a métrica f-score na curva de recall-precisão;

- Matrizes de confusão foram construídas utilizando a biblioteca *Matplotlib* e *Seaborn* para o caso geral para ambos limiares. Uma matriz para representar o caso de classe múltipla foi construída para analisar as assignações de hipóteses dos 20 aminoácidos possíveis, também para ambos limiares.

RESULTADOS E DISCUSSÃO

É ilustrado a seguir os resultados das principais métricas de avaliação do modelo final construído a partir da informação de 205 estruturas que totalizam 67.683 resíduos. Após validação cruzada dos hiperparâmetros e das variáveis, foram utilizadas curvas ROC e recall-precisão para se obter dois limiares diferentes de assignação. Nosso objetivo com os dois limiares é identificar dois cenários: i) resíduos onde a atribuição é definitiva; ii) resíduos ambíguos. Matrizes de confusão foram construídas para esses limiares para avaliar a predição das hipóteses binárias. Utilizando-se de adaptações do conceito de matriz de confusão, construiu-se matrizes para visualização simulando uma classificação múltipla dos 20 possíveis aminoácidos, analisando múltiplas assignações de hipóteses para um mesmo resíduo. Para o resultado do modelo final desenvolvido, construiu-se a saída do algoritmo que consiste em uma porcentagem de provável assignação correta de cada aminoácido para um determinado resíduo de uma proteína de interesse.

3.1 Validação Cruzada

Para maximizar o desempenho do modelo final, foram realizados 3 principais procedimentos de validação cruzada com resultados mostrados a seguir: verificação de tendência de *overfitting*; validação dos hiperparâmetros de XGBoost e das variáveis (*features*) que são utilizadas para predizer hipóteses corretas de aminoácidos.

A tabela 3.1 mostra o resultado obtido após validação cruzada com parâmetros semelhantes ao modelo inicial. Os dados foram divididos em 4 *folds,* considerando os parâmetros iniciais descritos na metodologia.

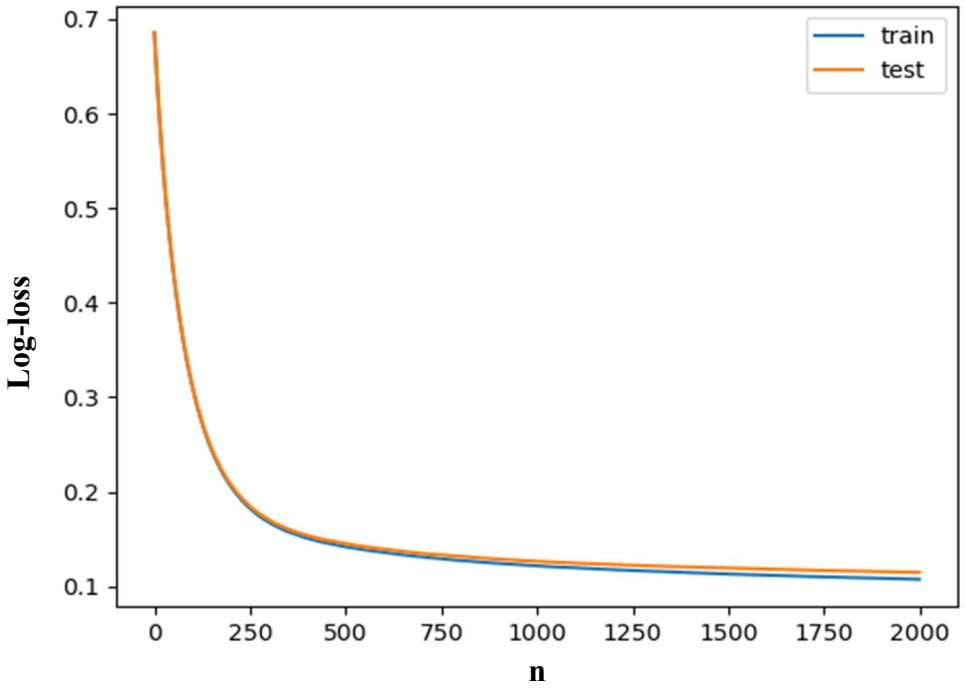
Tabela 3.1 – Resultado do teste de validação cruzada

	train-aucpr-mean	train-aucpr-std	test-aucpr-mean	test-aucpr-std
0	0,806423	0,003256	0,798872	0,004444
51	0,841498	0,001477	0,832281	0,003821
637	0,907810	0,000609	0,889635	0,002456
1132	0,922625	0,000637	0,898876	0,002004
1483	0,928224	0,000838	0,900820	0,001917
1941	0,933787	0,000678	0,902147	0,001962
1999	0,934442	0,000712	0,902232	0,001982

É ilustrada a média e o desvio padrão da métrica AUC-PR para os dados de treino e teste para 7 iterações (modelos) das 2000 realizadas, a primeira, a última e 5 de maneira aleatória em setores estratificados.

Os dados resultantes da validação cruzada apontam pequena tendência de *underfitting* nos dados, em virtude dos valores de média da métrica AUC-PR continuarem a crescer mesmo após 2000 iterações realizadas e pelo comportamento apresentado na figura 3.1, onde é plotada a curva de aprendizagem do modelo rudimentar.

Figura 3.1 – Curva de aprendizagem do modelo.

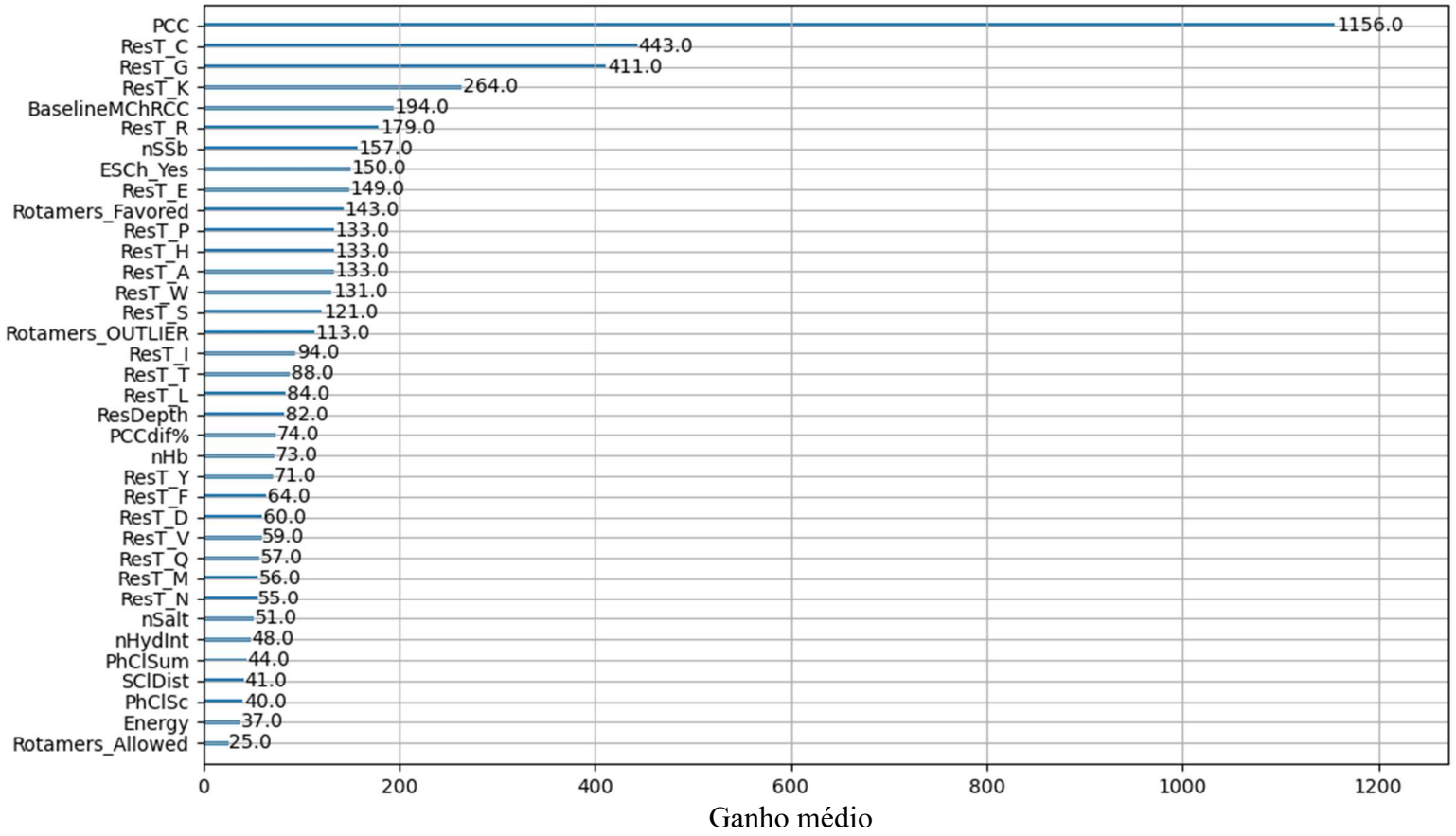


Fonte: Autor – É mostrado a métrica *log-loss* de ambos modelos; treino e teste, para *n* árvores construídas (iterações).

Para a validação cruzada dos hiperparâmetros, os resultados foram obtidos utilizando a biblioteca *Hyperotp* para os seguintes hiperparâmetros: *'gamma'*, *'reg_lambda'*, *'max_depth'* e *'learning_rate'* ao colocar parâmetros constantes de *'n_estimators'* e *'scale_pos_weight'* iguais a 2000 e 19, respectivamente, além dos já padronizados pela biblioteca, em no máximo 100 iterações (modelos construídos). A escolha dos parâmetros foi feita considerando a curva de aprendizagem ilustrada na figura 3.1, onde estes podem ser utilizados para diminuir o *underfitting* dos dados. Os parâmetros *'gamma'* e *'reg_lambda'*, ao final das 100 iterações, foram os valores 1,0 e 0, respectivamente. Estão relacionados com conservação do modelo a cada iteração, onde quanto maior valor nominal, mais conservativo o modelo se torna, o que ajuda a controlar *overfitting* segundo informação disponível na documentação de XGBoost e por isso, tiveram pouco impacto no modelo. Para '*learning_rate*', foi estabelecido o valor 0,025; corroborando com a figura 3.1, o aumento desse parâmetro, anteriormente com valor 0,01, faz com que o algoritmo aprenda mais rápido, de forma que foram necessárias apenas 1180 árvores para obter um modelo de desempenho superior ao de 2000 iterações anterior. O parâmetro *'max_depth'* divergiu do valor inicial, aumentando consideravelmente a cada rodada de validação, mas o aumento deste não contribuiu diretamente no ganho de desempenho. Este parâmetro aumenta a profundidade máxima permitida das árvores de decisão. Foi escolhido o valor 8 para o parâmetro *'max_depth',* pois valores acima deste não contribuíam para um aumento significativo nas métricas de avaliação de desempenho, não justificando o gasto de memória que é exponencialmente maior a cada aumento na profundidade máxima das árvores. A combinação do resultado do processo de validação cruzada dos hiperparâmetros de XGBoost observado indica que o modelo apresenta alta complexidade, pois múltiplas ramificações são necessárias para melhor caracterizar se determinada hipótese de aminoácido é correta.

Para a validação das variáveis do modelo, foi utilizada a função da biblioteca de XGBoost *plot_importance*() para organizar as variáveis por suas respectivas contribuições ao modelo. Analisou-se o atributo 'ganho', definido como a contribuição relativa de uma variável ao ser utilizada em uma ramificação, para o aumento de acurácia do modelo final. A figura 3.2 mostra os diferentes valores de ganho para cada uma das variáveis que compõe o modelo.

Figura 3.2 – Ganho médio das variáveis do modelo.



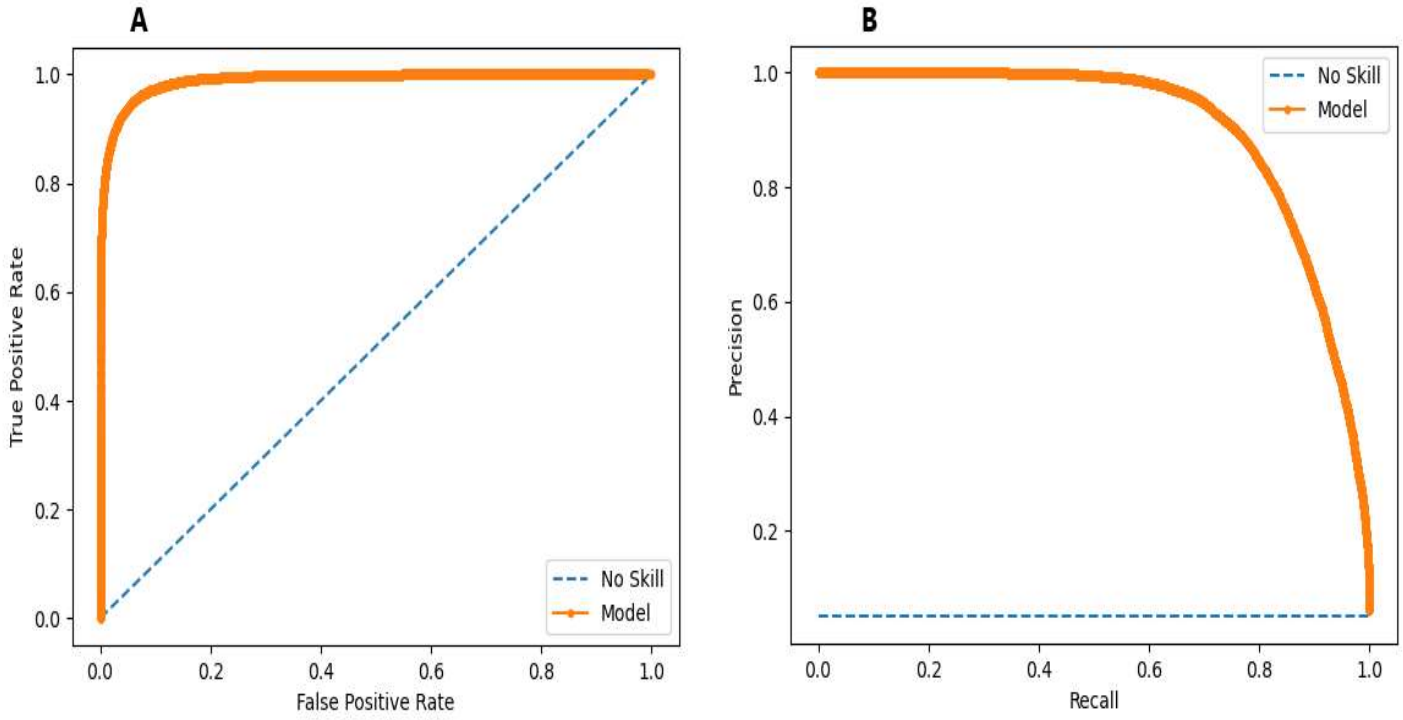
Fonte: Autor – Quanto maior o ganho médio atribuído a uma variável, maior é sua contribuição no desempenho do modelo.

Com base na figura 3.2 e em testes feitos utilizando modelos com variáveis faltantes (excluídas do treinamento do modelo), nenhuma delas foi retirada na construção do modelo final, pois todas ajudam, em alguma medida, no desempenho do mesmo. O RSCC é a variável que oferece maior discriminação no modelo de predição, de forma que os dados de difração de raios X são os mais relevantes. Embora as variáveis ResT sejam apenas classificadoras simples de uma hipótese de aminoácido, estas foram responsáveis por ganhos significativos. Essas variáveis podem ter auxiliado na separação de caraterísticas físico-químicas características desses resíduos por meio das árvores de decisão; uma correlação possível da figura 3.2 é acerca das variáveis ResT_C (linhas de informação relacionadas a hipótese do aminoácido cisteína) e nSSb (pontes de dissulfeto); ao considerar o fato observável de que o número de pontes de dissulfeto está relacionado diretamente com a existência de resíduos cisteína, é esperado valores de ganho para essas variáveis como observados na figura 3.2.

3.2 Curvas ROC e recall-precisão

Construiu-se duas curvas de avaliação com finalidade de aferir o desempenho do modelo final. A curva ROC (AUC = 0,989) e recall-precisão (AUC = 0,905) são apresentadas na figura 3.3.

Figura 3.3: Curvas de avaliação de desempenho.



Fonte: Autor – **A**: Curva ROC do modelo; AUC = 0,989. **B**: Curva recall-precisão do modelo; AUC = 0,905.

A utilização da curva ROC, obtida através da taxa de falsos positivos pela taxa de verdadeiros positivos (recall), é uma técnica que possibilita estimar a sensibilidade e especificidade de um modelo ao ilustrar possíveis escolhas de limiares de assignação, sendo um dos principais objetos de avaliação do desempenho de um modelo. A área embaixo da curva (AUC), representa a medição agregada de desempenho de todos os possíveis limiares de classificação. Valores próximos a 1 representam um modelo preditivo eficiente e próximos a 0.5 uma predição aleatória, sem valor. O valor de AUC obtido para o modelo final (0,989) indica que o modelo separa com excelência hipóteses de aminoácidos corretas com saída binária ‘1’ de erradas ‘0’.

Também é ilustrada na figura 3.3 a curva de recall-precisão do modelo (AUC = 0,905), obtida através da relação entre a precisão e o recall. Foi utilizada juntamente com a curva ROC,

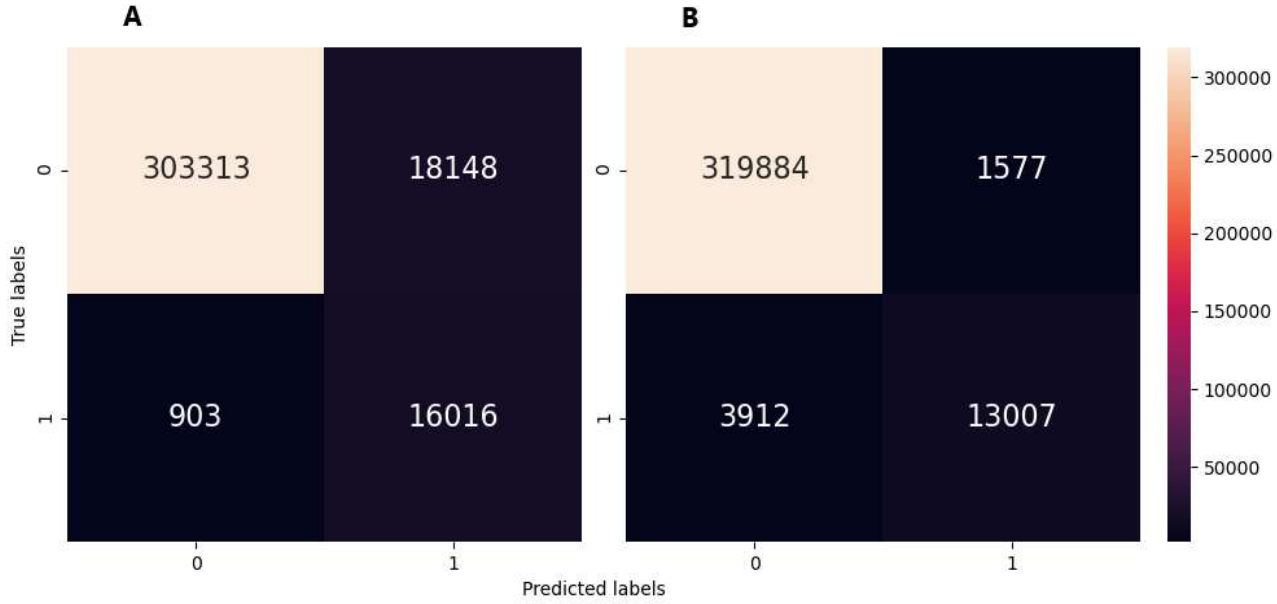
que é métrica frequentemente usada na avaliação de desempenho, pois há na literatura sugestões que a curva de recall-precisão é mais indicada na avaliação de modelos que possuem dados não balanceados (SAITO, REHMSMEIER, 2015). Considerando que há uma hipótese correta dentro de vinte possibilidades de aminoácidos naturais para o problema em questão, e que dados que não sejam balanceados podem representar desafios maiores para algoritmos de aprendizado de máquinas (FERNÁNDEZ et al., 2018), a comparação entre diferentes limiares escolhidos utilizando as características das curvas é relevante para analisar a qualidade da saída do algoritmo.

Foram escolhidos dois limiares de assignação. O primeiro valor de limiar escolhido foi de 0,3987 utilizando a curva ROC, definido a partir do índice de Youden, métrica avaliada quando se deseja escolher o limiar considerando o melhor balanço entre sensibilidade e especificidade (BEWICK, et al., 2004). O segundo valor escolhido foi de 0,9240 utilizando a maximização da métrica f-score na curva de recall-precisão. Maximizar a métrica f-score, que consiste na média harmônica da precisão e do recall, minimiza a quantidade de falsos positivos e negativos simultaneamente.

3.3 Matrizes amostrais e de confusão

Para ambos limiares, matrizes de confusão foram construídas para avaliar o comportamento das predições do algoritmo, com finalidade de compreender as limitações do modelo. A figura 3.4 mostra duas matrizes de confusão para diferentes limiares de assignação.

Figura 3.4: Matrizes de confusão do modelo para dois diferentes limiares de assignação.



Fonte: Autor – **A**: Matriz de confusão do modelo utilizando limiar de assignação de 0,3987 obtido pelo índice de Youden. **B**: Analogamente a A, mas com limiar de 0,9240, obtido utilizando maximização do f-score.
É possível verificar verdadeiros e falsos positivos e negativos utilizando uma matriz de confusão, mostrando hipóteses preditas e hipóteses verdadeiras.
Matrizes construídas utilizando a biblioteca Seaborn e Matplotlib,

Quanto menor o valor do limiar de assignação, mais hipóteses de aminoácidos são consideradas corretas, ou analogamente, atribuídas com o valor ’1’. É importante considerar que alguns resíduos podem não ter nenhuma hipótese, das 20 possíveis de aminoácidos naturais, considerada correta dependendo do limiar escolhido.

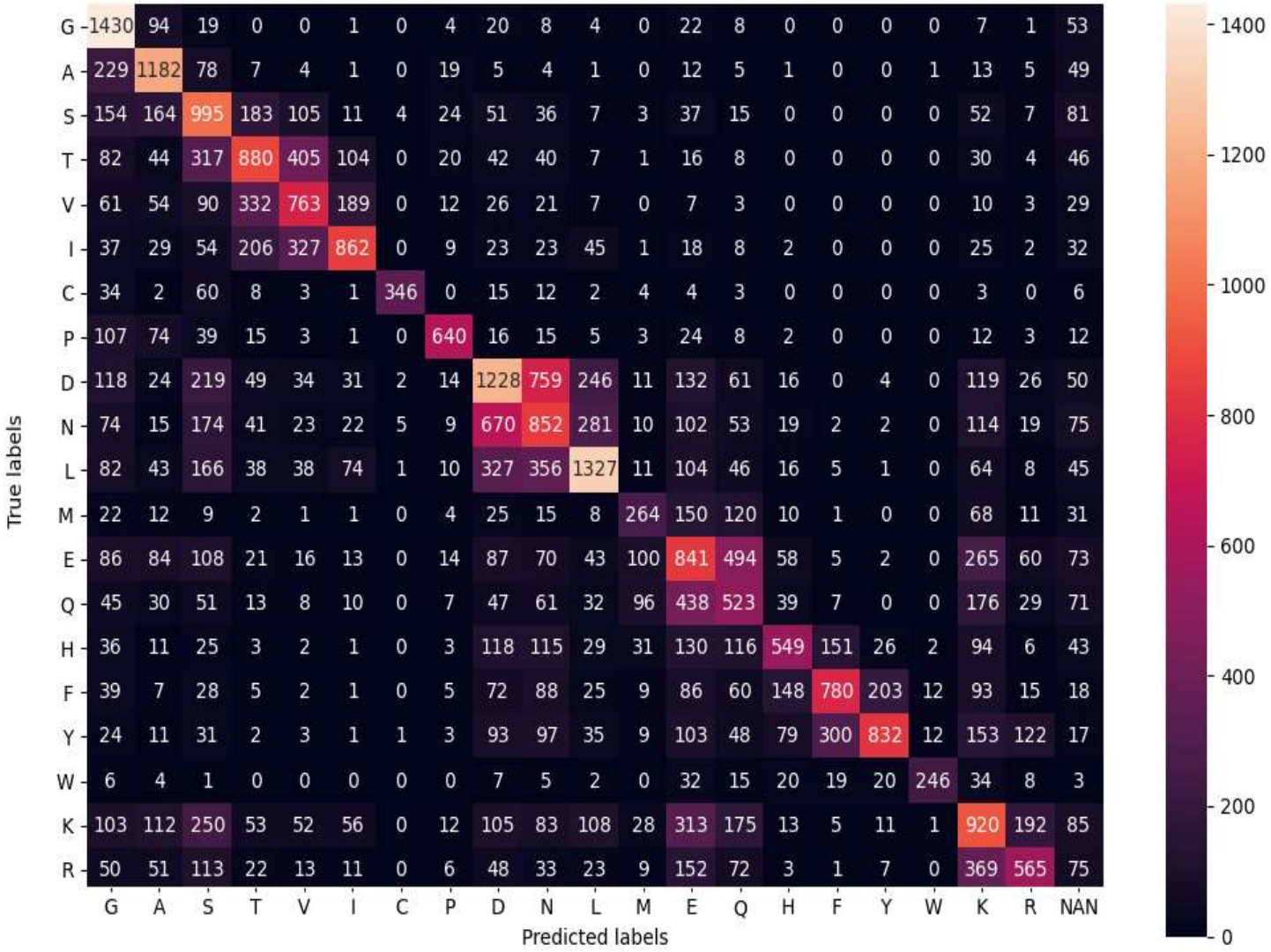
Para a matriz de confusão ‘A’ na figura 3.4, o limiar escolhido, através da estatística de Youden, busca minimizar os valores de sensibilidade e especificidade. Dos 16.919 resíduos utilizados para o grupo de teste, cada um com 20 possibilidades, analisados de 205 estruturas de proteínas de origem recombinante, 16.016 hipóteses foram assignadas conforme a informação previamente conhecida do banco de dados das proteínas utilizadas. A acurácia do

modelo para com limiar de 0,3987 foi de 94,3%, reduzida pela quantidade de falsos positivos, indicando uma baixa precisão, de 46,9%. Em contrapartida, o modelo para esse limiar apresenta elevada sensibilidade, de 94,7%.

Para a matriz de confusão 'B' da figura 3.4, o limiar escolhido, através da maximização da métrica f-score, busca minimizar falsos negativos e positivos simultaneamente, aumentando valores de precisão e sensibilidade de maneira conjunta. Dos 16.919 resíduos, 13.007 hipóteses foram assignadas conforme informação prévia. A acurácia do modelo para o limiar de 0,9240 foi de 98,4%, valor acima do observado no modelo de 'A', por diminuir o valor nominal da soma de falso positivos e negativos. Em contrapartida, o modelo para esse limiar apresenta baixa sensibilidade, de 76,9%, posto que o limiar de assignação tem valor nominal maior que 'A' (0,9240 > 0,3987).

Para visualizar o desempenho dos modelos na predição de aminoácidos, foi construída uma matriz adaptada de classe múltipla, utilizando o tipo de resíduo e sua predição binária discutida anteriormente. As figuras 3.5 e 3.6 mostram matrizes amostrais com separação por resíduos, onde a primeira foi construída utilizando o limiar 0,3987 e a segunda, 0,9240. Deve-se atentar para a não utilização de técnicas de avaliação métrica de matrizes de confusão às figuras 3.5 e 3.6, pois as informações estão dispostas de maneira diferente. Para exemplificar, o número total de glicinas (G) nos testes deve ser obtido através da soma dos valores $a_{G,G}$ e $a_{G,NAN}$ ao invés de computar todos os falsos positivos, pois estes podem aparecer mais de uma vez para um mesmo resíduo; uma alanina (A) pode ser assignada como correta erroneamente em um resíduo de glicina (G) e este estar computado em $a_{G,G}$.

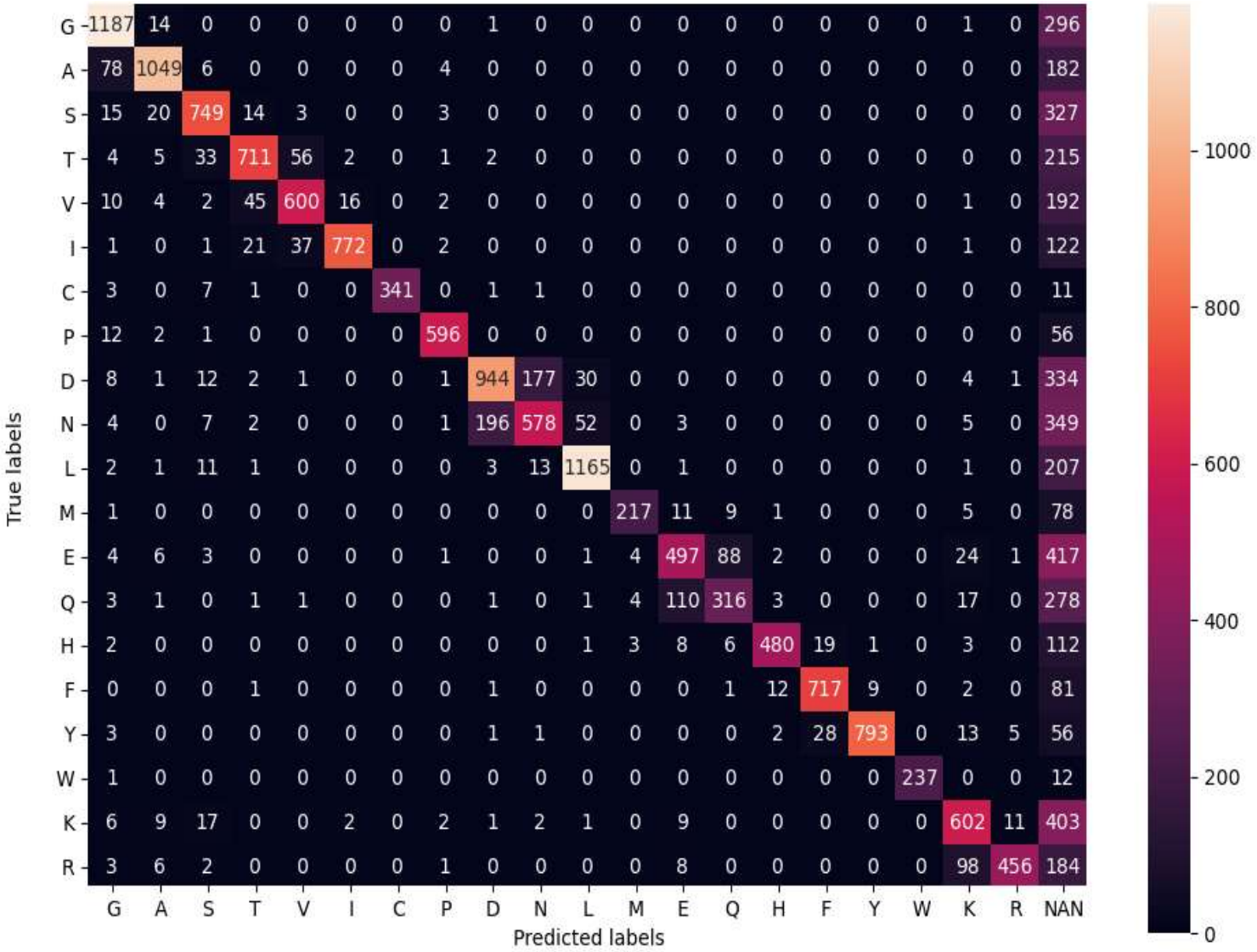
Figura 3.5: Representação por resíduo para limiar igual a 0,3987.



Fonte: Autor – A matriz mostra quais hipóteses de resíduos foram preditas e quais eram os verdadeiros aminoácidos de acordo com o banco de dados. A última coluna à direita representa predições iguais a '0' pelo algoritmo, e portanto, não predizem a hipótese correta.
Matriz construída utilizando a biblioteca Seaborn e Matplotlib,

A disposição dos rótulos dos aminoácidos nas matrizes por resíduos foi escolhida considerando suas respectivas densidades eletrônicas, colocando aminoácidos semelhantes neste quesito lado a lado (um único elétron de diferença em alguns casos). É possível, utilizando esse recurso, verificar a presença de agrupamentos bem definidos (*clusters*) de indecisão para o limiar de 0,3987. Estes falsos positivos representados, porém, dificilmente passam de duas a três hipóteses de resíduos na maioria dos casos, o que sugere que o modelo apresenta potencial em excluir grande parte de hipóteses de aminoácidos erradas.

Figura 3.6: Representação por resíduo para limiar igual a 0,9240.



Fonte: Autor – Análoga à figura 3.5.

Matriz construída utilizando a biblioteca Seaborn e Matplotlib,

Para o limiar de 0,9240; há drástica redução no número de falsos positivos, com aumento de falsos negativos quando comparado com o obtido pelo método de Youden. A especificidade elevada do modelo contribui para a diminuição de dúvidas nos clusters observados no limiar definido anteriormente (0,3987). Porém, embora a acurácia (98,4%) e o f-score (0,826) do modelo sejam maiores, alguns tipos de aminoácidos podem não ser tão bem classificados, como o ácido glutâmico (E).

Considerando o objetivo proposto de complementar SLIDER utilizando um algoritmo de aprendizado de máquinas, em uma primeira análise, o limiar mais restritivo (‘A’) permite atribuir aminoácidos com alto grau de confiabilidade. Para os casos onde não há clara

discriminação, é possível aplicar o limiar mais permissivo ('B'), onde mais opções de aminoácidos são possíveis, cabendo ao usuário decidir entre discriminar entre as possibilidades utilizando técnicas complementares ou modelar mais de um aminoácido para posição com ocupação parcial ou modelar uma Glicina ou Alanina com atribuição UNK (*Unknown*, desconhecido).

3.4 Saída do algoritmo

A saída do algoritmo consiste em uma probabilidade de assignação correta para cada uma das 20 hipóteses de aminoácidos considerando o resíduo específico analisado. A probabilidade de hipótese de um aminoácido p_a foi calculada a partir da equação (2.13):

$$p_a = 100 * \frac{x_a}{\sum_{i=1}^{20} x_i}, \tag{2.13}$$

onde x_a é a probabilidade de saída do algoritmo de aprendizado de máquinas para uma hipótese de aminoácido e $\sum_{i=1}^{20} x_i$, a somatória de todas as 20 probabilidades de um único resíduo.

É dada em um arquivo de formato CSV, que possibilita análise em forma de tabelas para facilitar a visualização e interpretação. As tabelas 3.2, 3.3 e 3.4 mostram diferentes perfis de saída do algoritmo para resíduos específicos da proteína PLA_2 pancreática de humano (código do PDB 3ELO) do grupo de teste escolhida.

Tabela 3.2 – 3ELO-A-6

ResT	Aa%	ResT	Aa%
P	90,3	L	0,1
A	4,6	M	0,1
G	2,0	N	0,1
I	0,8	K	0,1
V	0,4	C	0,0
D	0,3	H	0,0
E	0,3	R	0,0
Q	0,3	F	0,0
S	0,2	Y	0,0
T	0,2	W	0,0

Tabela 3.3 – 3ELO-A-31

ResT	Aa%	ResT	Aa%
N	28,3	Y	1,1
D	28,0	I	0,9
E	15,8	F	0,8
T	6,0	K	0,3
S	5,3	M	0,2
A	4,1	C	0,0
H	3,8	L	0,0
Q	2,2	P	0,0
G	1,9	W	0,0
V	1,2	R	0,0

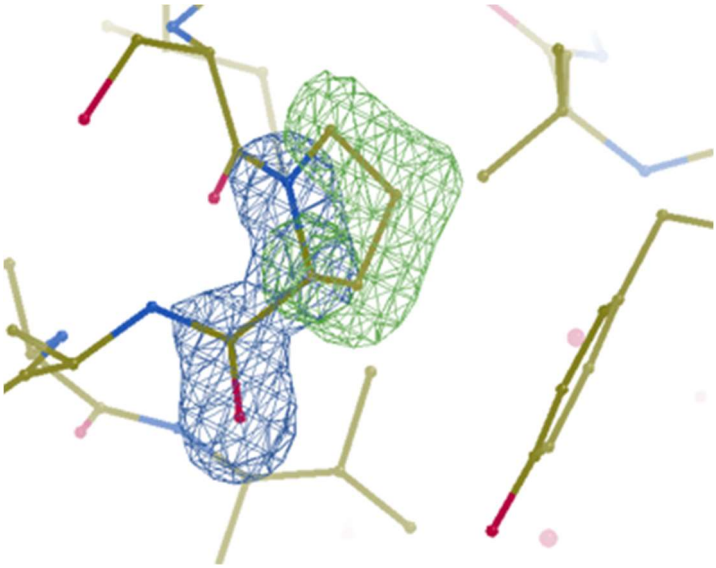
Tabela 3.4 – 3ELO-A-1

ResT	Aa%	ResT	Aa%
T	12,0	A	3,2
S	11,4	I	3,1
G	10,8	R	2,4
V	10,1	L	1,7
Q	8,3	M	1,6
N	8,2	Y	1,3
P	6,6	H	1,1
E	6,0	F	0,7
K	5,7	W	0,2
D	5,5	C	0,0

Fonte: Autor – Tabelas da porcentagem de assignação para cada hipótese de aminoácido. O quadrado vermelho indica qual o aminoácido correto de acordo com o banco de dados.

A tabela 3.2 é referente ao sexto resíduo da cadeia 'A' da proteína PLA_2 pancreática de humano (código do PDB 3ELO), cuja numeração é -1. Neste caso o algoritmo previu corretamente ao pontuar 90,3% de chance que o resíduo analisado é uma prolina, hipótese essa que condiz com a estrutura depositada. A prolina é um aminoácido com densidade eletrônica muito característica e por isso é facilmente resconhecível pelo modelo na maioria dos casos. A figura 3.7 mostra os dados de densidade eletrônica para esse resíduo.

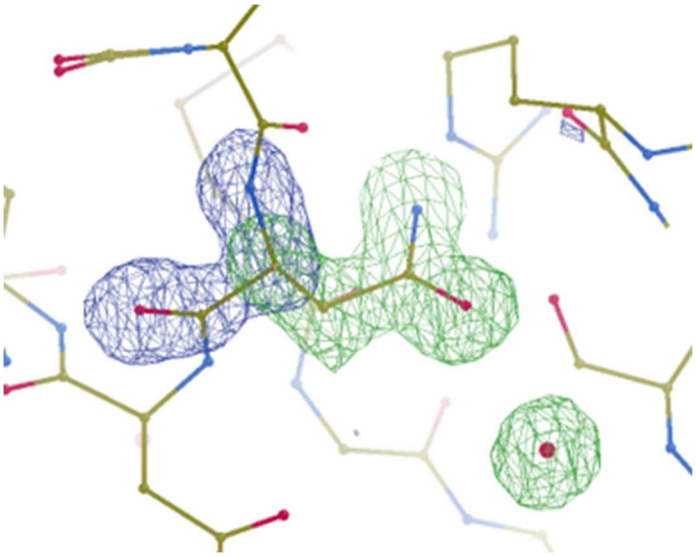
Figura 3.7: Representação da densidade eletrônica para o resíduo 3ELO-6



Fonte: Autor – Em azul é ilustrada a densidade eletrônica da cadeia principal, em verde a cadeia lateral.

A tabela 3.3 é referente ao trigésimo-primeiro resíduo da cadeia ‘A’ da proteína PLA_2 pancreática de humano (código do PDB 3ELO), cuja numeração é 24. O algoritmo apresenta dúvida entre 3 principais aminoácidos: asparagina (N) com 28,3%, ácido aspártico (D) com 28,0% e ácido glutâmico (E) com 15,8%. Para os dois aminoácidos mais prováveis, a diferença de densidade eletrônica é de apenas um elétron, e por isso é esperado que este seja um resíduo de elucidação desafiadora para o modelo considerando que o RSCC é a variável com maior ganho. A figura 3.8 mostra os dados de densidade eletrônica para esse resíduo.

Figura 3.8: Representação da densidade eletrônica para o resíduo 3ELO-31

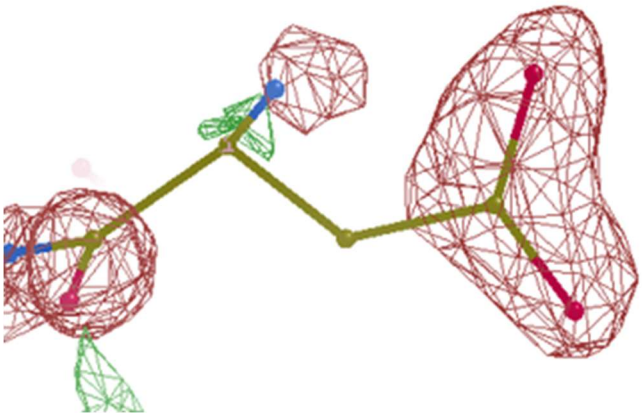


Fonte: Autor – Em azul é ilustrada a densidade eletrônica da cadeia principal, em verde a cadeia lateral.

É possível observar na tabela 3.3 que o restante das probabilidades estão uma ou duas ordens de grandeza abaixo das primeiras, possibilitando o descarte de algumas hipóteses. Ao considerar uma proteína purificada a partir de fontes naturais, cuja sequência é desconhecida, essa exclusão de hipóteses pode auxiliar o pesquisador, que possui melhor conhecimento de uma proteína da qual estuda, atribuir o aminoácido; mesmo para as três primeiras probabilidades, caso este, hipoteticamente, tenha a informação de que naquele resíduo há um aminoácido de carácter básico, a assignação poderia ser direta, ao excluir as hipóteses de aminoácidos ácidos que possuem densidade eletrônica semelhante.

A tabela 3.4 é referente ao primeiro resíduo da cadeia ‘A’ da PLA_2 pancreática de humano (código do PDB 3ELO), cuja numeração é -6. O algoritmo não conseguiu prever corretamente a hipótese correta, que está marcada com vermelho, considerando que as probabilidades estão com valores próximos. A figura 3.9 mostra os dados de densidade eletrônica para esse resíduo.

Figura 3.9: Representação da densidade eletrônica para o resíduo 3ELO-1



Fonte: Autor – Em vermelho é mostrada zonas de densidade eletrônica negativa, os dados apresentados apresentam ruído.

Este cenário ocorreu por conta deste resíduo estar localizado no início da proteína em uma região exposta ao solvente sem densidade definida, o que é observado com valores nominais de RSCC abaixo de 80, incluindo o RSCC da cadeia principal de valor 45.5. Ao investigar o *B-factor*, parâmetro de deslocamento atômico que traduz quanto um átomo necessita deslocar-se para ocupar a densidade eletrônica observada, é possível constatar que este resíduo apresenta um valor de 18.36, referente a 3,4σ acima da média de todos os átomos da proteína (média de 7,34; desvio padrão 3,23) Estas observações são indicativos de alta flexibilidade, e, portanto, podem apresentar informação ruidosa. Dos 67.683 resíduos escolhidos, 2.968 (4,4%) apresentaram valores de RSCC abaixo de 80 para a cadeia lateral.

Considerando os exemplos apresentados de saída do algoritmo das tabelas 3.2, 3.3 e 3.4, a aplicação do aprendizado de máquinas ao método SLIDER fornecerá uma complementação métrica complementar à análise empírica da densidade eletrônica e do entorno físico-químico. Adicionalmente, é possível excluir hipóteses com maior segurança utilizando tanto as novas probabilidades obtidas através da aplicação do modelo, quanto a análise empírica. Ainda,

porém, a separação de hipóteses de aminoácidos com densidades eletrônicas semelhantes é desafiadora, mesmo considerando que o modelo, como verificado na análise de importância de variáveis (figura 2.8), utiliza dados físico-químicos para construir ramificações que impactam no ganho de acurácia.

CONCLUSÃO

Proteínas purificadas a partir de fontes naturais, como hemoglobinas, lisozimas ou toxinas, podem apresentar incerteza de sequência e sua purificação pode ser desafiador pela coexistência de isoformas. O método SEQUENCE SLIDER aplicado a toxinologia propõe assignar sequência destas proteínas integrando com dados disponíveis, sendo utilizado para a elucidação de uma metaloproteases (SALVADOR et al., 2020) e uma fosfolipase A_2 homóloga de *Bothrops moojeni* (SALVADOR et al., 2021) e duas isoformas de uma PLA_2 de *Bothrops jararacussu* (BORGES et al., 2021). Nestes artigos, dados de cristalografia, espectrometria de massa e análise filogenética foram integrados (BORGES et al., 2022). A análise das densidades eletrônicas e do entorno físico-químico foi realizada manualmente, desta maneira a implementação de aprendizado de máquinas para atribuir probabilidades automatiza SLIDER e auxiliará a elucidação de estruturas futuras.

A implementação do algoritmo de aprendizado de máquinas XGBoost aos dados cristalográficos do SLIDER cria uma nova métrica que integra caracterização de densidade eletrônica e do entorno físico-químico: a probabilidade de assignação de hipóteses de aminoácidos para cada resíduo. Esta implementação foi possível utilizando diversos dados cristalográficos de proteínas com sequência conhecida, já que sua produção foi de forma recombinante. Com os resultados do modelo, é proposto dois limiares de assignação, o primeiro com alta especificidade onde um aminoácido se discrimina dos demais; e o segundo com alta sensibilidade considera mais de um aminoácido como possível. Neste sentido, a metodologia agora otimizada, com uma etapa adicional automática envolvendo aprenzidado de máquinas, se torna aplicável a elucidar novas estruturas cristalográficas de proteínas purificadas a partir de fontes naturais e podem ser utilizados por outros biologistas estruturais ao ser disponibilizado à comunidade. Os resultados gerados podem melhor caracterizar os componentes de venenos de serpente, e auxiliar no desenvolvimento de estratégias mais eficientes no tratamento contra envenenamento ofídico.

REFERÊNCIAS

ADAMS, P. D. ET AL. PHENIX: A COMPREHENSIVE PYTHON-BASED SYSTEM FOR MACROMOLECULAR STRUCTURE SOLUTION. **Acta Crystallographica Section D Biological Crystallography**, v. 66, n. 2, p. 213–221, 22 jan. 2010.

AGUIAR, J. A. ET AL. DECODING CRYSTALLOGRAPHY FROM HIGH-RESOLUTION ELECTRON IMAGING AND DIFFRACTION DATASETS WITH DEEP LEARNING. **Science Advances**, v. 5, n. 10, p. eaaw1949, out. 2019.

BERGSTRA, J., YAMINS, D., COX, D. D. MAKING A SCIENCE OF MODEL SEARCH: HYPERPARAMETER OPTIMIZATION IN HUNDREDS OF DIMENSIONS FOR VISION ARCHITECTURES. **30th International Conference on Machine Learning**, 2013.

BERMAN, H. M. THE PROTEIN DATA BANK. **Nucleic Acids Research**, v. 28, n. 1, p. 235–242, 1 jan. 2000.

BORGES, R. J. ET AL. SEQUENCE SLIDER : EXPANDING POLYALANINE FRAGMENTS FOR PHASING WITH MULTIPLE SIDE-CHAIN HYPOTHESES. **Acta Crystallographica Section D Structural Biology**, v. 76, n. 3, 1 mar. 2020.

BORGES, R. J. ET AL. BTHTX-II FROM BOTHROPS JARARACUSSU VENOM HAS VARIANTS WITH DIFFERENT OLIGOMERIC ASSEMBLIES: AN EXAMPLE OF SNAKE VENOM PHOSPHOLIPASES A2 VERSATILITY. **International Journal of Biological Macromolecules**, v. 191, p. 255–266, nov. 2021.

BORGES, R. J. ET AL. SEQUENCE SLIDER: INTEGRATION OF STRUCTURAL AND GENETIC DATA TO CHARACTERIZE ISOFORMS FROM NATURAL SOURCES. **Nucleic Acids Research**, gkac029, 2022.

BRASIL. MINISTÉRIO DA SAÚDE. SECRETARIA DE VIGILÂNCIA EM SAÚDE. DEPARTAMENTO DE VIGILÂNCIA EPIDEMIOLÓGICA. In: **Doenças Infecciosas e Parasitárias: Guia de Bolso**. 7. ed. Brasília, DF: Ministério da Saúde, 2010. p. 431–437.

BRASIL. MINISTÉRIO DA SAÚDE. SECRETARIA DE VIGILÂNCIA EM SAÚDE. DEPARTAMENTO DE VIGILÂNCIA EPIDEMIOLÓGICA. Casos de acidentes por serpentes. Brasil, Grandes Regiões e Unidades Federadas 2000 a 2018. [s.l: s.n.].

BRASIL. MINISTÉRIO DA SAÚDE. SECRETARIA DE VIGILÂNCIA EM SAÚDE. DEPARTAMENTO DE VIGILÂNCIA EPIDEMIOLÓGICA. Óbitos por serpentes. Brasil, Grandes Regiões e Unidades Federadas 2000 a 2018. [s.l: s.n.].

BREIMAN, LEO ET AL. **Classification and regression trees**. Routledge, 2017.

BUCHAN, D. W. A. ET AL. SCALABLE WEB SERVICES FOR THE PSIPRED PROTEIN ANALYSIS WORKBENCH. **Nucleic Acids Research**, v. 41, n. W1, p. W349–W357, 1 jul. 2013.

CALVETE, J. J.; JUÁREZ, P.; SANZ, L. SNAKE VENOMICS. STRATEGY AND APPLICATIONS. **Journal of mass spectrometry: JMS**, v. 42, n. 11, p. 1405–1414, nov. 2007.

CALVETE, J. J. et al. SNAKE VENOMICS OF THE CENTRAL AMERICAN RATTLESNAKE CROTALUS SIMUS AND THE SOUTH AMERICAN CROTALUS DURISSUS COMPLEX POINTS TO NEUROTOXICITY AS AN ADAPTIVE PAEDOMORPHIC TREND ALONG CROTALUS DISPERSAL IN SOUTH AMERICA. **Journal of proteome research**, v. 9, n. 1, p. 528–544, jan. 2010.

CHAP T. LE. A SOLUTION FOR THE MOST BASIC OPTIMIZATION PROBLEM ASSOCIATED WITH AN ROC CURVE. **Statistical Methods in Medical Research**. 15: 571-584, 2006.

CHEN, Tianqi; GUESTRIN, Carlos. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. 2016. p. 785-794.

COCK P.A., ET AL., BIOPYTHON: FREELY AVAILABLE PYTHON TOOLS FOR COMPUTATIONAL MOLECULAR BIOLOGY AND BIOINFORMATICS. **Bioinformatics**, 25, 1422-1423

CUTLER, ADELE; CUTLER, D. RICHARD; STEVENS, JOHN R. RANDOM FORESTS. In: **Ensemble machine learning**. Springer, Boston, MA, 2012. p. 157-175.

DOLEY, R.; KINI, R. M. PROTEIN COMPLEXES IN SNAKE VENOM. **Cellular and molecular life sciences: CMLS**, v. 66, n. 17, p. 2851–2871, set. 2009.

FERNÁNDEZ, A. ET AL. **Learning from Imbalanced Data Sets**. Cham: Springer International Publishing, 2018.

FLUSS, RONEN; FARAGGI, DAVID; REISER, BENJAMIN. ESTIMATION OF THE YOUDEN INDEX AND ITS ASSOCIATED CUTOFF POINT. **Biometrical Journal: Journal of Mathematical Methods in Biosciences**, v. 47, n. 4, p. 458-472, 2005.

FREUND, Yoav; SCHAPIRE, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, v. 55, n. 1, p. 119-139, 1997.

FRIEDMAN, Jerome H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, p. 1189-1232, 2001.

GUTIÉRREZ, J. M.; THEAKSTON, R. D. G.; WARRELL, D. A. CONFRONTING THE NEGLECTED PROBLEM OF SNAKE BITE ENVENOMING: THE NEED FOR A GLOBAL PARTNERSHIP. **PLoS Medicine**, v. 3, n. 6, jun. 2006.

HAGEMANS, D. ET AL., A SCRIPT TO HIGHLIGHT HYDROPHOBICITY AND CHARGE ON PROTEIN SURFACES. **Frontiers in Molecular Biosciences**, v. 2, 13 out. 2015.

HARRIS, C.R., MILLMAN, K.J., VAN DER WALT, S.J. ET AL. ARRAY PROGRAMMING WITH NUMPY. **Nature 585**, 357–362 (2020).

HUNTER, J. D. MATPLOTLIB: A 2D GRAPHICS ENVIRONMENT. **Computing in Science & Engineering**, v. 9, n. 3, p. 90–95, 2007.

JASKOLSKI, M.; DAUTER, Z.; WLODAWER, A. A BRIEF HISTORY OF MACROMOLECULAR CRYSTALLOGRAPHY, ILLUSTRATED BY A FAMILY TREE AND ITS NOBEL FRUITS. **The FEBS journal**, v. 281, n. 18, p. 3985–4009, set. 2014.

KALIL, J.; FAN, H. W. PRODUCTION AND UTILIZATION OF SNAKE ANTIVENOMS IN SOUTH AMERICA. In: GOPALAKRISHNAKONE, P. (Ed.). **Toxins and Drug Discovery**. Dordrecht: Springer Netherlands, 2016. p. 1–22.

KASTURIRATNE, A. ET AL. THE GLOBAL BURDEN OF SNAKEBITE: A LITERATURE ANALYSIS AND MODELLING BASED ON REGIONAL ESTIMATES OF ENVENOMING AND DEATHS. **PLoS medicine**, v. 5, n. 11, p. e218, 4 nov. 2008.

LIEBSCHNER, D. ET AL. POLDER MAPS: IMPROVING OMIT MAPS BY EXCLUDING BULK SOLVENT. **Acta Crystallographica Section D Structural Biology**, v. 73, n. 2, p. 148–157, 1 fev. 2017.

LOMONTE, B. ET AL. THE PHOSPHOLIPASE A2 HOMOLOGUES OF SNAKE VENOMS: BIOLOGICAL ACTIVITIES AND THEIR POSSIBLE ADAPTIVE ROLES. **Protein and peptide letters**, v. 16, n. 8, p. 860– 876, 2009.

MCCOY, A. J. ET AL. PHASER CRYSTALLOGRAPHIC SOFTWARE. **Journal of Applied Crystallography**, v. 40, n. 4, p. 658–674, 13 jul. 2007.

MCDONALD, I. K.; THORNTON, J. M. SATISFYING HYDROGEN BONDING POTENTIAL IN PROTEINS. **Journal of Molecular Biology**, v. 238, n. 5, p. 777–793, 19 mai. 1994.

MCGUFFIN, L. J.; BRYSON, K.; JONES, D. T. THE PSIPRED PROTEIN STRUCTURE PREDICTION SERVER. **Bioinformatics (Oxford, England)**, v. 16, n. 4, p. 404–405, abr. 2000.

MCKINNEY, DATA STRUCTURES FOR STATISTICAL COMPUTING IN PYTHON. **Proceedings of the 9th Python in Science Conference**, Volume 445, p. 56-61, 2010.

MEDINA, A. ET AL. ALEPH: A NETWORK-ORIENTED APPROACH FOR THE GENERATION OF FRAGMENT-BASED LIBRARIES AND FOR STRUCTURE INTERPRETATION. **Acta Crystallographica Section D Structural Biology**, v. 76, n. 3, 1 mar. 2020.

MOHAMMED, M.; KHAN, M. B.; BASHIER, E. B.. **Machine learning: algorithms and applications**. Crc Press, 2016.

NIELSEN, Didrik. **Tree boosting with xgboost-why does xgboost win" every" machine learning competition?**. 2016. Dissertação de Mestrado. NTNU.

PEDREGOSA, F. ET AL. SCIKIT-LEARN: MACHINE LEARNING IN PYTHON. **The Journal of Machine Learning Research**, v. 12, n. null, p. 2825–2830, 1 nov. 2011.

RODRIGUES, R. S. ET AL. SNAKE VENOM PHOSPHOLIPASES A2: A NEW CLASS OF ANTITUMOR AGENTS. **Protein and Peptide Letters**, v. 16, n. 8, p. 894–898, 2009.

RUPP, B. **Biomolecular crystallography: principles, practice, and application to structural biology**. New York: Garland Science, 2010.
SAITO, TAKAYA; REHMSMEIER, MARC. THE PRECISION-RECALL PLOT IS MORE INFORMATIVE THAN THE ROC PLOT WHEN EVALUATING BINARY CLASSIFIERS ON IMBALANCED DATASETS. **PloS one**, v. 10, n. 3, p. e0118432, 2015.

SALVADOR, G. H. M. ET AL. BIOCHEMICAL, PHARMACOLOGICAL AND STRUCTURAL CHARACTERIZATION OF BMOOMP-I, A NEW P–I METALLOPROTEINASE FROM BOTHROPS MOOJENI VENOM. **Biochimie**, v. 179, p. 54–64, dez. 2020.

SALVADOR, G. H. M. ET AL. THE SYNTHETIC VARESPLADIB MOLECULE IS A MULTI-FUNCTIONAL INHIBITOR FOR PLA2 AND PLA2-LIKE OPHIDIC TOXINS. **Biochimica et Biophysica Acta (BBA) - General Subjects**, v. 1865, n. 7, p. 129913, jul. 2021.
SAMMITO, M. ET AL. ARCIMBOLDO_LITE: SINGLE-WORKSTATION IMPLEMENTATION AND USE. **Acta Crystallographica. Section D, Biological Crystallography**, v. 71, n. Pt 9, p. 1921–1930, 1 set. 2015.

SAUL, F. A. ET AL. COMPARATIVE STRUCTURAL STUDIES OF TWO NATURAL ISOFORMS OF AMMODYTOXIN, PHOSPHOLIPASES A2 FROM VIPERA AMMODYTES AMMODYTES WHICH DIFFER IN NEUROTOXICITY AND ANTICOAGULANT ACTIVITY. **Journal of Structural Biology**, v. 169, n. 3, p. 360–369, mar. 2010.

SCHRÖDINGER, LLC. **The PyMOL Molecular Graphics System, Version 1.8**. nov. 2015.

SUROWIECKI, JAMES. **The wisdom of crowds**. Anchor, 2005.
ROKACH, LIOR. **Pattern classification using ensemble methods**. World Scientific, 2010.

TANHA, Jafar et al. Boosting methods for multi-class imbalanced data classification: an experimental review. **Journal of Big Data**, v. 7, n. 1, p. 1-47, 2020.

USÓN, I.; SHELDRICK, G. M. AN INTRODUCTION TO EXPERIMENTAL PHASING OF MACROMOLECULES ILLUSTRATED BY SHELX ; NEW AUTOTRACING FEATURES. **Acta Crystallographica Section D Structural Biology**, v. 74, n. 2, p. 106–116, 1 fev. 2018.

VANERIO, Juan; CASAS, Pedro. Ensemble-learning approaches for network security and anomaly detection. In: **Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks**. 2017. p. 1-6.

VAN ROSSUM, G., & DRAKE, F. L. **Python 3 Reference Manual.** Scotts Valley, CA: CreateSpace, 2009.

WASKOM, M. SEABORN: STATISTICAL DATA VISUALIZATION. **Journal of Open Source Software**, v. 6, n. 60, p. 3021, 6 abr. 2021.

WORLD HEALTH ORGANIZATION. RABIES AND ENVENOMINGS: A NEGLECTED PUBLIC HEALTH ISSUE: REPORT OF A CONSULTATIVE MEETING, World Health Organization, Geneva, 10 January 2007. p. 32, 2007.

ZHANG, CHA; MA, YUNQIAN (ED.). **Ensemble machine learning: methods and applications**. Springer Science & Business Media, 2012.

ZHOU, ZHI-HUA. **Ensemble methods: foundations and algorithms**. Chapman and Hall/CRC, 2019.

APÊNDICE A – ESTRUTURAS UTILIZADAS

As 41 PLA_2s escolhidas foram: 1FX9, 1HN4, 1LE6, 1LE7, 1O2E, 1O3W, 2AZZ, 2AZY, 2B00, 2B01, 2B03, 2B04, 2B96, 2BAX, 2BCH, 2BD1, 2WG7, 2WG8, 2WG9, 2ZP3, 2ZP4, 2ZP5, 3ELO, 3FVJ, 3G8F, 3G8G, 3U8B, 3U8H, 4UY1, 5G3M, 5OW8, 5OWC, 5VFM, 5WZM, 5WZO, 5WZS, 5WZT, 5WZU, 5WZV, 5WZW e 5Y5E.

Os 15 receptores de porina escolhidos foram: 1BH3, 2ZFG, 3SY7, 3SYB, 3SZV, 3T0S, 3T24, 3VZT, 4GCP, 4LSI, 5DL7, 5O77, 5PRN, 6PRN e 7PRN.

As 149 metaloproteases escolhidas foram: 2YB9, 1UTT, 3NX7, 5CUH, 3L0V, 2J83, 5D1T, 3TS4, 4PKS, 3LEA, 4ON1, 4IJO, 3EHX, 3LIL, 4GR8, 3LJZ, 1I1I, 3KEC, 3Q2G, 4R3V, 2V4B, 3G42, 2O3E, 4PKQ, 3HDA, 4XM6, 2D1N, 1OS9, 1JIW, 4PKU, 4XM7, 3LJG, 4CTH, 3LQB, 3HB2, 3KMC, 2OY2, 4GF1, 2HU6, 4JQG, 3L0T, 3F19, 2X3A, 4I03, 3LJT, 4FXQ, 2OW9, 4GR0, 4JIJ, 4WK7, 4H30, 5D2B, 3WV1, 3EHY, 4JP4, 3LIR, 6GID, 4PKR, 1ROS, 3F18, 4EFS, 4L6T, 1UTZ, 3HBU, 4AXQ, 5B5O, 5BOY, 1R55, 4PKW, 4H3X, 3DPE, 3ELM, 4WF6, 2OXW, 5D1S, 4L63, 1JIZ, 3KEJ, 3ZXH, 3DPF, 5D3C, 3KHI, 3KME, 4ILW, 3LGP, 2OXU, 3KRY, 3EDZ, 3LK8, 4A3W, 2YIG, 4GQL, 2Y6C, 3E8R, 1HFS, 3B8Z, 3HBV, 3V96, 3WV3, 1I73, 2DDF, 1Y93, 3HYG, 1R54, 3O64, 4JA1, 3ZVS, 3DL1, 2FV5, 4WKE, 1ZP5, 3N2V, 4H76, 4GR3, 3F15, 2O36, 3LQ0, 4DV8, 4WZV, 5B5P, 4A7B, 3F16, 5CZM, 1RMZ, 4H84, 5D1U, 4JPA, 1SLM, 2CLT, 4WKI, 4FVL, 2XS4, 3EWJ, 1S4B, 5CXA, 2X7M, 2X3C, 1ZVX, 2D1O, 5D7W, 3F1A, 3LE9, 3KEK, 4DPE, 1ZS0, 2OVX, 4H1Q, 3HY7 e 1I76.