

UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
CURSO DE PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Dissertação de Mestrado

**Aplicações de *Ensemble Learning* para o Estudo do Efeito
de Mutações Pontuais em Estruturas Tridimensionais de
Proteínas**

Eduardo Kenji Hasegawa de Freitas

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Profa. Dra. Karina dos Santos Machado
Co-orientador: Prof. Dr. Adriano Velasque Werhli

Rio Grande, 2020

Ficha Catalográfica

F866a Freitas, Eduardo Kenji Hasegawa de.
Aplicações de *Ensemble Learning* para o estudo do efeito de mutações pontuais em estruturas tridimensionais de proteínas / Eduardo Kenji Hasegawa de Freitas. – 2020.
56 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2020.

Orientadora: Dra. Karina dos Santos Machado.

Coorientador: Dr. Adriano Velasque Werhli.

1. *Weka* 2. *Machine Learning* 3. *Ensemble Learning*
4. Classificação 5. Regressão I. Machado, Karina dos Santos
II. Werhli, Adriano Velasque III. Título.

CDU 004:577.1

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344



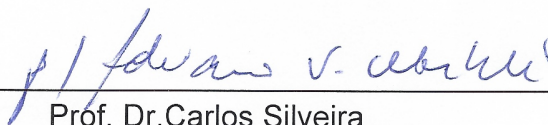
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

DISSERTAÇÃO DE MESTRADO

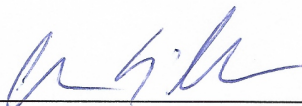
**Aplicações De Esemble Learning Para o Estudo do Efeito de Mutações
Pontuais em Estruturas Tridimensionais de Proteínas**

Eduardo Kenji Hasegawa de Freitas

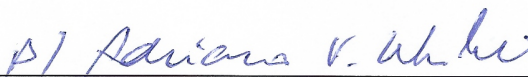
Banca examinadora:



Prof. Dr. Carlos Silveira

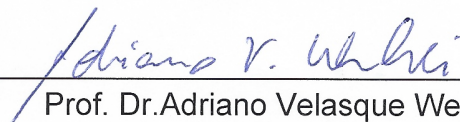


Prof. Dr. Cleo Zanella Billa



Prof.ª Dr.ª Karina dos Santos Machado

Orientadora



Prof. Dr. Adriano Velasque Werhli

Coorientadora

SUMÁRIO

RESUMO	6
ABSTRACT	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS E SIGLAS	11
1 INTRODUÇÃO	12
1.1 Objetivo Geral	14
1.2 Objetivos Específicos	14
1.3 Justificativa	14
1.4 Organização do Texto	15
2 REFERENCIAL TEÓRICO	16
2.1 Proteínas e níveis de informação estrutural	16
2.2 Mutações pontuais e seus efeitos em estruturas de proteínas	16
2.3 Energia livre de Gibbs (G)	17
2.4 Aprendizado de Máquina	18
2.4.1 Algoritmos de Classificação e Regressão	20
2.4.2 <i>Ensemble Learning</i>	22
2.4.3 Avaliação dos Algoritmos de Classificação e <i>Ensemble Learning</i>	24
2.4.4 Avaliação dos Algoritmos de Regressão	26
3 REVISÃO DA LITERATURA	27
3.1 Trabalho relacionado 1: CUPSAT	27
3.2 Trabalho relacionado 2: DUET	27
3.3 Trabalho relacionado 3: mCSM	28
3.4 Trabalho relacionado 4: SDM	28
3.5 Trabalho relacionado 5: MAESTRO	29
3.6 Trabalho relacionado 6: PoPMuSic	29
3.7 Trabalho relacionado 7: EN-MUTATE	29
4 METODOLOGIA	32
4.1 Bases de dados	33
4.2 Pré-processamento	33
4.3 Submissão de dados	35

4.4	Geração dos modelos	36
5	RESULTADOS	37
5.1	Experimento 1: Base de treinamento balanceada	37
5.1.1	Algoritmos de Classificação e <i>Ensemble</i>	38
5.1.2	Algoritmos de Regressão	38
5.2	Experimento 2: Base de treinamento e teste com proporção 70/30	39
5.2.1	Algoritmos de Classificação e <i>Ensemble</i>	40
5.2.2	Algoritmos de Regressão	41
5.3	Experimento 3: <i>Cross-validation</i>	41
5.3.1	Algoritmos de Classificação e <i>Ensemble</i>	42
5.3.2	Algoritmos de Regressão	42
6	DISCUSSÃO	44
6.1	Análise de limiares	46
7	CONCLUSÃO	50
	REFERÊNCIAS	52

RESUMO

HASEGAWA DE FREITAS, Eduardo Kenji. **Aplicações de *Ensemble Learning* para o Estudo do Efeito de Mutações Pontuais em Estruturas Tridimensionais de Proteínas**. 2020. 56 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

O refinar de propriedades das proteínas, através de mutações pontuais sobre seus aminoácidos é uma prática muito comum utilizada em processos da indústria bioquímica. Métodos computacionais acurados são necessários para realizar a predição sobre esses experimentos de mutações, tornando o design de proteínas mais eficiente. Por meio de bases de dados provenientes do Protherm, onde cada instância inclui dados numéricos, como variação da energia livre de Gibbs, mudança de entalpia, mudança de capacidade térmica, temperatura de transição, entre outros, são informações importantes para a compreensão da estabilidade da proteína. As predições do efeito da mutação na estrutura da proteína medido pela variação da energia de Gibbs ($\Delta\Delta G$) são divididas entre duas classes, estabilizante e desestabilizante, onde algoritmos de classificação e *ensemble* de classificadores, disponibilizados pelo *software* Weka, terão a função de determinar a acurácia dos modelos de predição. Através de três experimentos, que são diferenciados pelo pré-processamento dos dados de entrada para os modelos de predição, é avaliado o comportamento dos das predições cada ferramenta, proporcionando uma discussão de como a bioinformática pode se beneficiar desses resultados e como os modelos de predição criados podem prever o impacto de mutações pontuais na estrutura de proteínas.

Palavras-chave: Weka, machine learning, ensemble learning, classificação, regressão.

ABSTRACT

HASEGAWA DE FREITAS, Eduardo Kenji. **Ensemble Learning Applications for Studying the Effect of Single Point Mutations on Three Dimensional Protein Structures.** 2020. 56 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

The refining of protein properties, through point mutations on their amino acids, is a very common practice used in biochemical industry processes. Accurate computational methods are required to carry out the prediction on these mutation experiments, making protein design more efficient. Through databases from Protherm, where each instance includes numerical data, such as Gibbs free energy variation, enthalpy change, thermal capacity change, transition temperature, among others, are important for the understanding of protein stabilization. The predictions will be divided between two classes, stabilizing and destabilizing, where algorithms for classification and ensemble classifiers, available on the Weka software, have the objective to determine the accuracy of the prediction models. By making use of three experiments, that are unique in the way of data input pre-processing for the prediction models, it is evaluated the prediction behavior of each tool, providing a discussion on how bioinformatics can benefit from these results and how the created predicting models can predict the impact of point mutations on the structure of proteins.

Keywords: weka, machine learning, ensemble learning, classification, regression.

LISTA DE FIGURAS

1	Arquitetura de exemplo para uma rede de neurônios com camadas de entrada, ocultas e de saída adaptado de [Isokawa et al., 2012].	21
2	EN-MUTATEweb	30
3	Fluxograma da metodologia	32
4	Representação de relevância de valores da matrizes de confusão para os cálculos das métricas de avaliação: (a) Acurácia alta, Precisão baixa e Revocação alta; (b) Acurácia baixa, Precisão baixa e Revocação alta; (c) Acurácia alta, Precisão alta e Revocação baixa.	44
5	Fluxograma da análise de limiars	46
6	Comparação das matrizes de confusão das sete ferramentas agrupadas.	47
7	Comparação das matrizes de confusão das sete ferramentas individualmente.	48

LISTA DE TABELAS

1	Exemplo de entrada da metodologia EN-MUTATE para as ferramentas de predição	31
2	Número de instâncias, proteínas únicas, classes estabilizantes e classes desestabilizantes por base de dados.	33
3	Exemplo de entrada da base de dados SP1	34
4	Resultado do pré-processamento de SP1	34
5	Exemplo de entrada da base de dados S2648	34
6	Resultado do pré-processamento de S2648	34
7	Exemplo de entrada da base de dados S1948	34
8	Resultado do pré-processamento de S1948	35
9	Comparação de dados antes e depois do pré-processamento.	35
10	Número de instâncias, estabilizantes e desestabilizantes, para bases de treinamento e testes nos experimentos 1 e 2. O experimento 3 segue o algor de validação cruzada para treinamento e testes.	36
11	Métricas de avaliação para a classificação das ferramentas do Experimento 1	38
12	Métricas de avaliação para a classificação dos algoritmos do Experimento 1	38
13	Métricas de avaliação para os algoritmos de <i>Ensemble</i> de Classificadores do Experimento 1	39
14	Métricas de avaliação para a regressão das ferramentas do Experimento 1	39
15	Métricas de avaliação para a regressão dos algoritmos do Experimento 1	39
16	Métricas de avaliação para a classificação das ferramentas do Experimento 2	40
17	Métricas de avaliação para a classificação dos algoritmos do Experimento 2	40
18	Métricas de avaliação para os algoritmos de <i>Ensemble</i> de Classificadores do Experimento 2	41
19	Métricas de avaliação para a regressão das ferramentas do Experimento 2	41
20	Métricas de avaliação para a regressão dos algoritmos do Experimento 2	41
21	Métricas de avaliação para a classificação dos algoritmos do Experimento 3	42

22	Métricas de avaliação para os algoritmos de <i>Ensemble</i> de Classificadores do Experimento 3	43
23	Métricas de avaliação para a regressão dos algoritmos do Experimento 3	43

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
API	<i>Application Program Interface</i>
DNA	<i>DeoxyriboNucleic Acid</i>
FN	Falso Negativo
FP	Falso Positivo
FURG	Universidade Federal do Rio Grande
IA	Inteligência Artificial
PDB	<i>Protein Data Bank</i>
RBM	<i>Restricted Boltzman Machines</i>
RMSE	<i>Root Mean Squared Error</i>
RNA	<i>RiboNucleic Acid</i>
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
KDD	<i>Knowledge-Discovery in Databases</i>

1 INTRODUÇÃO

Dados biológicos são produzidos e adicionados constantemente em grandes bancos de dados, como o *Protein Data Bank* (ou PDB). De acordo com estatísticas sobre estruturas moleculares no PDB, no ano 2000 foram adicionadas 2.627 estruturas novas, totalizando 13.590 estruturas no banco de dados. Enquanto que até 15 de setembro de 2019 já foram inseridas 8.332 novas entradas totalizando 155.830 estruturas de proteínas [Berman et al., 2017]. Como consequência desse crescimento de dados, os computadores tornaram-se indispensáveis na pesquisa biológica, porque vêm crescendo em poder computacional para o processamento de grandes quantidade de dados.

São três os tipos de dados moleculares mais comuns abordados na bioinformática: o *DeoxyriboNucleic Acid* (DNA), o *RiboNucleic Acid* (RNA) e as proteínas [Verli, 2014]. O DNA tem a função de armazenar a codificação de proteínas enquanto que o RNA, tem como uma de suas funções realizar a tradução da informação retirada de células para auxiliar na produção de proteínas.

A tradução é dividida em três processos: iniciação, alongamento e terminação. O processo de iniciação necessita de alguns elementos base para a criação da proteína, são eles: um ribossomo, um RNA mensageiro com as instruções da proteína e um RNA transportador com o primeiro aminoácido da proteína. Esses elementos irão criar a configuração inicial da nova proteína. Com a formação inicial do polipeptídeo, inicia-se o processo de alongamento onde vão sendo adicionados mais aminoácidos ao efetuar-se ligações peptídicas que montam a sequência de aminoácidos. A terminação, então, ocorre quando os códons de parada são recebidos pelo RNAm reconhecidos pro proteínas chamadas de fatores de liberação. Os fatores de liberação modifica a enzima que realiza as ligações peptídicas e a fazem adicionar uma molécula de água no último aminoácido da cadeia, finalizando a sequência de aminoácidos.

Essas proteínas têm a habilidade de reconhecer e interagir com o DNA, onde elas evoluíram ao ponto de se poder identificar regiões específicas no DNA baseando-se apenas na sequência de nucleotídeos e no formato estrutural da molécula [Vytautas and de G. Bert, 2017]. Proteínas são as moléculas orgânicas mais abundantes em sistemas vivos e têm as mais diversas quantidades e funções dentre as macromoléculas. Elas podem ser estrutu-

rais, regulatórias, contráteis ou protetoras. Podem servir como transporte, armazenamento ou membranas, assim como podem ser enzimas ou toxinas. Cada célula, em um sistema vivo, contém milhares de proteínas onde cada uma tem uma função, fazendo com que suas estruturas sejam muito variadas [Rye et al., 2016].

Existem dois tipos de mutações pontuais: substituição de bases e mudança de quadros. A substituição ocorre quando troca-se a posição de uma base com outra. Já no segundo tipo, ocorre quando uma nova base é adicionada ou removida na sequência de DNA [Voet et al., 2014].

A fim de analisar melhor os experimentos de dados moleculares e, conseqüentemente, as mutações pontuais nas estruturas de proteínas, surge a bioinformática, que é definida como a aplicação de técnicas computacionais para entender e organizar as informações relacionadas com macromoléculas biológicas [Luscombe et al., 2001]. Essas técnicas envolvem também processamento de imagens e sinais, onde são extraídos resultados de grandes quantidades de dados brutos. A biologia aplicada em sequências tem suas aplicações mais focadas em mineração de dados e análises de dados de projetos genomas, alinhamento de sequência, redes metabólicas, métricas morfológicas e evolução virtual [Chou, 2004]. Enquanto que os baseados em estruturas têm suas aplicações na predição de estruturas 3D de proteínas e descoberta de relações entre estruturas e as funções da proteína. Este projeto tem como objetivo estudar aplicações em bioinformática estrutural [Chou, 2004].

Uma das áreas de pesquisa em bioinformática tem o objetivo de estudar o efeito de mutações pontuais na estrutura das proteínas e utilizar aproximações computacionais para caracterizar a variação termodinâmica que essas mutações causam. Foram desenvolvidas nas últimas décadas, uma série de ferramentas computacionais com técnicas químicas e físicas, tendo o objetivo comum de prever o efeito de mutações pontuais sobre a estrutura de uma proteína. Em geral, as ferramentas de predição são utilizadas por profissionais de várias áreas que, geralmente, não acessam as mesmas ferramentas para suas análises, o que implica em discrepâncias de resultados e dificuldade de interpretação, já que terão resultados diferentes para entradas de dados iguais. Dentre as ferramentas, estão: Dmutant [Zhou and Zhou, 2009], FoldX [Guerois et al., 2002], I-Mutant2.0 [Capriotti et al., 2004], CUPSAT [Parthiban et al., 2006], Eris [Yin et al., 2007], AUTO-MUTE [Masso and Vaisman, 2008], I-Mutant3.0 [Capriotti et al., 2008], PoPMuSiC [Dehouck et al., 2011], Pro-Maya [Wainreb et al., 2011], SDM [Catherine L. Worth, 2011], mCSM [Pires et al., 2013], NeEMO [Giollo et al., 2014], MUpro [Cheng et al., 2005], STRUM [Quan et al., 2016] e MAESTRO [Laimer et al., 2015]. Existem também ferramentas que se utilizam da técnica *ensemble learning*, como o DUET [Douglas E.V. Pires, 2014], ou o EN-MUTATEweb [Alex, 2017]. O *ensemble learning* visa agregar predições de vários modelos classificadores em apenas um resultado. Essas combinações podem ser tanto por votação (para classificações) como por média (para regressões). O resultado final é consi-

derado como o classificador *ensemble* que, na maioria das vezes, tem resultados melhores do que resultados individuais dos classificadores utilizados na combinação [Yang, 2017].

Sendo assim, este trabalho propõe utilizar algoritmos de aprendizado de máquina (classificação, regressão e *ensemble* de classificadores), para propor modelos que combinem os resultados de predição do efeito de mutações pontuais em estruturas de proteínas obtidos de diferentes ferramentas de predição.

1.1 Objetivo Geral

Aplicar *ensemble learning* para analisar o efeito de mutações pontuais em estruturas das proteínas.

1.2 Objetivos Específicos

Os principais objetivos esperados são:

- Selecionar as ferramentas de predição de mutações pontuais;
- Encontrar e pré-processar bases de dados para submeter nas ferramentas;
- Aplicar algoritmos de *ensemble learning* sobre os resultados das ferramentas
- Utilizar métricas de avaliação para analisar o desempenho do *ensemble learning* sobre os resultados individuais de cada ferramenta.

1.3 Justificativa

A predição sobre o impacto que mutações pontuais têm nas estruturas proteínas e, conseqüentemente, sobre os seres vivos é de uma grande importância para a pesquisa biológica. Uma vez que essas mutações podem ser tanto benéficas a seres vivos, como neutras ou letais.

Poder entender como as mutações se comportariam, apenas com previsões realizadas sobre a variação da variação de energia livre ($\Delta\Delta G$) nessas mutações é algo que vem sendo desenvolvido nos últimos anos [Magliery, 2015].

No entanto, as ferramentas existentes na comunidade acadêmica nem sempre obtêm resultados semelhantes para tipos de entradas iguais [Alex, 2017]. Essa variação nos resultados pode acabar influenciando negativamente nos experimentos realizados por pesquisadores.

As mutações, classificadas como estabilizantes, são utilizadas para o desenvolvimento de drogas e remédios para doenças e patogêneses. Portanto, para auxiliar os profissionais da área a descobrirem esses componentes seletos, é importante que a precisão na classificação de mutações estabilizantes seja ideal.

1.4 Organização do Texto

O trabalho está composto da seguinte forma:

- Capítulo 2: O referencial teórico tem o objetivo de apresentar os conceitos biológicos que serviram de base para a realização de estudos sobre como funcionam as mutações e como os algoritmos de aprendizado de máquina podem auxiliar na predição de tais experimentos;
- Capítulo 3: Esta seção tem como objetivo apresentar as principais ferramentas e trabalhos que influenciaram este trabalho, assim como introduzir como funcionam;
- Capítulo 4: A metodologia irá apresentar como foram realizados os objetivos específicos deste trabalho, identificando as bases de dados, o pré-processamento sobre os dados, as submissões à ferramentas e a aplicação de algoritmos de classificação, regressão e *ensemble*;
- Capítulo 5: Este capítulo irá apresentar todos os resultados de experimentos e predições realizados pela metodologia;
- Capítulo 6: Continuando após os resultados, este capítulo apresenta a discussão sobre os resultados, assim como uma comparação entre resultados de métodos, ou ferramentas, diferentes;
- Capítulo 7: Finalizando, a última seção apresenta as conclusões sobre os trabalhos feitos e seus resultados. Além disso, são listados possíveis trabalhos futuros a serem realizados.

2 REFERENCIAL TEÓRICO

Nesta seção, são apresentados conceitos de estruturas de proteínas, assim como suas mutações, até as métricas de avaliação de nossos modelos de predição, criados por algoritmos de aprendizado de máquina que realizam classificação, regressão ou *ensemble* dos valores apresentados pelas bases de dados utilizadas.

2.1 Proteínas e níveis de informação estrutural

Os aminoácidos são elementos que formam as proteínas e todos possuem a mesma estrutura fundamental, onde um átomo central de Carbono (C) que se conecta com um grupo amino (NH₂), um grupo carboxílico (COOH) e um átomo de hidrogênio. Dentre os aminoácidos presentes em proteínas diferenciadas pela cadeia lateral, 20 são os mais comuns, sendo que 10, dentre eles, são considerados essenciais em humanos, uma vez que não são possíveis de serem produzidos pelo corpo humano, mas são recebidos de acordo com a dieta diária. Portanto, a sequência e os tipos de aminoácidos determinam o formato, tamanho e função de uma proteína [Rye et al., 2016].

Existem quatro tipos de níveis estruturais nas proteínas: primárias, secundárias, terciárias e quaternárias. A estrutura primária é definida como qualquer sequência única de aminoácidos; A estrutura secundária é formada através do desdobramento da disposição das sequências de aminoácidos, mediadas principalmente por ligações de hidrogênio, em formatos, nos casos mais comuns, de Folhas Beta ou de Hélices Alfa (Helicoidal); A terciária se dá pelo arranjo tridimensional de todas as estruturas, em que essas interações podem ser não covalentes entre as cadeias laterais dos aminoácidos constituintes ou, em alguns casos, podem ser covalentes designadas por pontes de dissulfureto; A estrutura quaternária é formada quando as proteínas são constituídas por mais de uma cadeia polipeptídica e há uma disposição tridimensional de diferentes cadeias na mesma estrutura.

2.2 Mutações pontuais e seus efeitos em estruturas de proteínas

Para melhor entender como funcionam as mutações, é preciso entender o que são nucleotídeos e que o resultado de mutações trazem consequências para a estrutura de

proteínas e suas funções.

Os genes são bases de sequências de DNA que codificam proteínas, sendo que a ordem dessas bases determinam a ordem de aminoácidos e, conseqüentemente, as funções da proteína. Os nucleotídeos são unidades repetitivas nessa sequência de DNA, como por exemplo: timina (T), adenina (A), guanina (G) ou citosina (C) [Voet et al., 2014]. Como exemplos de um tipo de mutação que geralmente ocorre devido a erros na replicação do DNA, estão os pareamentos entre: Guanina com a Timina, a Adenina com a Citosina ou vice-versa.

As conseqüências podem ser de caráter silencioso, não-silencioso (“*missense*” [Zhang et al., 2012]) ou sem sentido (“*nonsense*”). Em caráter “*missense*”, a mutação no DNA atinge uma alteração da sequência de aminoácidos constituintes da proteína, podendo alterar sua estrutura e função. No caráter silencioso, o mRNA gerado, embora também contenha a informação alterada, resulta na formação da mesma proteína. Já o caráter “*nonsense*”, ocorre quando a alteração dos pares de bases gerou informação para uma parada prematura, que sinaliza o fim da síntese da proteína causando a ausência de aminoácidos na cadeia da proteína e, conseqüentemente, sem função [Auclair et al., 2006].

Com a modificação do DNA de um ser vivo, essas mutações podem passar para gerações seguintes. Embora existam mutações benéficas, neutras ou que não causam qualquer efeito, existem mutações maléficas, como exemplificado nos artigos [Dolzhanskaya et al., 2014] [Kucukkal et al., 2014] [Steffl et al., 2013] [Kucukkal et al., 2015]. O foco deste trabalho é sobre mutações pontuais, que são, geralmente, o resultado da modificação de apenas um nucleotídeo no genoma, resultando numa possível modificação em um aminoácido da proteína.

2.3 Energia livre de Gibbs (G)

Um passo para o entendimento da relação entre estrutura das proteínas e suas funções vem da predição do efeito de variação da estabilidade de proteínas em mutações pontuais. Como citado na introdução deste trabalho, diversas ferramentas avaliam a variação dessa energia nas mudanças causadas pela mudança das proteínas entre seu estado nativo e suas variantes, utilizando como informação tanto a sequência de aminoácidos como a estrutura das proteínas.

A energia livre de Gibbs é estabilizada quando a proteína está em equilíbrio constante, tanto em pressão como temperatura. Portanto, a proteína em seu estado nativo tem uma certa quantidade de energia G_{nativo} . Quando o processo de enovelamento da proteína ocorre, a energia livre de Gibbs sofre uma variação, proporcionando uma variação de energia ΔG_{nativo} . Da mesma forma que, se analisarmos a mesma proteína em um estado de mutação, temos, em seu equilíbrio, uma energia $G_{mutante}$ e, ao ocorrer o enovelamento desta proteína mutada, podemos calcular o $\Delta G_{mutante}$. Então, com o enovelamento das

proteínas nativa e mutante, pode-se calcular o $\Delta\Delta G$, que é calculado através da Equação 1 [Y. Sugita, 1998] [Fersht, 1993].

$$\Delta\Delta G = \Delta G_{nativo} - \Delta G_{mutante} \quad (1)$$

$$\Delta\Delta G_{(A \rightarrow B)} = -\Delta\Delta G_{(B \rightarrow A)} \quad (2)$$

Para se avaliar o efeito da mutação, utiliza-se como propriedade de predição a anti-simetria da variação de energia e sua variante reversa, como por exemplo a Equação 2, onde A e B são aminoácidos. Dependendo da interpretação de cada autor das ferramentas, pode-se obter valores positivos ou negativos de $\Delta\Delta G$ para um mesmo experimento, onde um autor pode ter realizado uma avaliação de A para B e o outro ter realizado uma avaliação inversa [Montanucci et al., 2019].

O principal atributo que será avaliado e utilizado por este trabalho é a variação da variação da energia livre de Gibbs ($\Delta\Delta G$), onde utilizam-se a unidade $kcal/mol$ para os valores quantitativos ou utilizam-se seu modo discretizado como estabilizante, para valores positivos (> 0), ou desestabilizante, para valores negativos (< 0).

Para a criação de modelos que possam avaliar o $\Delta\Delta G$, são propostas algumas opções de algoritmos de aprendizado de máquina (classificação, regressão e *ensemble* de classificadores) na seção a seguir

2.4 Aprendizado de Máquina

Estudos sobre AM iniciaram em meados de 1960 [Samuel, 1959]. O AM utiliza algoritmos para analisar dados, aprender com eles e, depois, determinar ou prever sobre algo no mundo. Assim, ao invés de seguir instruções específicas, a máquina treina utilizando grandes quantidades de dados com algoritmos capazes de dizer como aprender a realizar tal tarefa [FACELI et al., 2011].

Com sua base na inteligência artificial (IA) e com o passar dos anos, os algoritmos de AM passaram a incluir aprendizado por indução de árvores de decisão, programação lógica intuitiva, agrupamento, aprendizado por reforço, redes Bayesianas, entre outros [Pierre and Soren, 2002].

Este trabalho utiliza o aprendizado de máquina supervisionado, de forma que os algoritmos possam aplicar tudo que foi aprendido, com os dados de treinamento, sobre os dados de testes. Os dados de treinamento possuem atributos alvo já previamente definidos por especialistas da área, o $\Delta\Delta G$. Tendo um atributo em que o aprendizado de máquina possa se basear, é criado um modelo de predição que, com o decorrer do treinamento, define os próximos atributos alvos para que se exclua a necessidade de influência externa. Com um modelo pronto, é possível inserir dados de teste para que seja possível prever

que resultados o experimento, com as características atuais, pode alcançar. O algoritmo de aprendizado também pode comparar seu resultado com o resultado dos especialistas para, ainda mais, aprimorar seu modelo.

Dentre os algoritmos de aprendizado de máquina, estão os algoritmos de classificação, regressão e *ensemble*. Os classificadores criam modelos de predição para prever um, ou mais, rótulos para cada instância inserida. Já os algoritmos de regressão têm o objetivo de prever valores quantitativos.

Os métodos de classificação requerem exemplos que possam ser classificados em uma de duas, ou mais, classes. Podem ser tanto valores reais como variáveis discretizadas. Problemas que tenham duas classes são denominadas classificação binárias, enquanto que problemas de três, ou mais classes, como classificação multi-classes. É comum para modelos de classificação prever valores contínuos como a probabilidade de determinado exemplo ser de determinada classe. Portanto uma predição probabilística pode ser convertida como uma classe que componha a maior probabilidade, de acordo com as características do exemplo predito em questão. Esses modelos são avaliados, então, de acordo com a acurácia dos modelos para cada classe a ser classificada.

Os métodos de regressão requerem exemplos que estejam em formato quantitativo, sendo que podem ser, também, valores reais ou discretizados. Quando um exemplo de regressão contém mais de uma variável, o modelo é denominada de regressão de múltiplas variáveis. Esses problemas podem ser avaliados através do cálculo do *root mean squared error*, ou RMSE. Neste trabalho, os métodos de regressão são utilizados para que se possa analisar possíveis diferenças nas predições entre a utilização de discretização dos dados (estabilizante ou desestabilizante), e a utilização dos dados quantitativos.

No que se diferencia métodos de classificação de métodos de regressão está o fato de que modelos preditivos de regressão podem prever valores contínuos, enquanto que os modelos preditivos de classificação podem prever valores discretizados. Alguns algoritmos podem ser aplicados em ambas situações, como é o caso das *Support Vector Machines* (Seção 2.4.1.4) e o *MultiLayer Perceptron* (Seção 2.4.1.3), sendo modificados os devidos parâmetros de sua função. Também existem algoritmos que são específicos de cada método, como a Regressão Linear (Seção 2.4.1.5) para métodos de regressão e *Naive Bayes* (Seção 2.4.1.2) para métodos de classificação.

Devido a natureza de nossos dados, utilizados em todos os modelos de predição deste trabalho, aplicamos o conceito de dois níveis de *ensemble*. Um nível utiliza os algoritmos de classificação e regressão para criar modelos a partir dos resultados de ferramentas preditivas, enquanto que, o outro nível, aplica um consenso de algoritmos de classificação sobre os resultados das ferramentas. O segundo nível citado seria o *Ensemble Learning*, que é mais detalhado na Seção 2.4.2.

2.4.1 Algoritmos de Classificação e Regressão

Nesta seção são introduzidos os algoritmos que serão utilizados para gerar modelos de predição utilizados na metodologia aqui apresentada. Para os métodos de classificação, foi utilizado o J48, o *Naive Bayes*, o *Multilayer Perceptron* e o *Support Vector Machines*, enquanto que, para os métodos de regressão, utilizamos a Regressão Linear, o *Multilayer Perceptron* e o *Support Vector Machines*.

2.4.1.1 J48

O J48 é um algoritmo de classificação simples que utiliza árvores de decisão binárias. Ele executa através de tuplas de instâncias, analisando cada atributo, ou característica da instância, para criar as regras de predição do atributo alvo. Nessa metodologia, o J48 tem a possibilidade de complementar dados faltantes nos dados de treinamento, utilizando-se das instâncias próximas ao dado faltante e deduzindo sua classificação. Após a definição de todas as regras de decisão da árvore, o algoritmo retorna e retira todas as regras que não obtiveram peso de influência sobre as decisão final da predição da classe.

2.4.1.2 Naive Bayes

O *Naive Bayes* é uma técnica que constrói classificadores, utilizando vetores de características, em que os dados provém de um determinado grupo de dados. Não é formado por apenas um algoritmo, mas toda a família de algoritmos *Naive Bayes* segue a ideia de que cada característica, da entrada de dados para o algoritmo, é independente das outras características, dada uma variável de classe.

O princípio estatístico do *Naive Bayes* envolve calcular a probabilidade de cada uma das classes para determinada característica dentre toda a população de características na base de dados. A classe com maior probabilidade, para aquela característica, é o resultado da predição. Portanto, o modelo prepara cada uma das características da base de testes e determina a probabilidade de cada uma delas ser uma das classes. Com os modelos criados, o algoritmo pode aplicar sobre a base de testes e predizer a classe resultado para cada instância de acordo com as características nela encontrada [Prabhakaran, 2018].

2.4.1.3 Multilayer Perceptron

O *Multilayer Perceptron* segue o conceito de camadas para o aprendizado de máquina, onde se divide, no mínimo, em três camadas. Como pode ser exemplificado na Figura 1, a primeira camada é a de entrada de dados e a última camada é a de saída de dados, as camadas ocultas podem ser multiplicadas de acordo com a necessidade da dimensão de dados e a necessidade de utilizar algoritmos mais complexos.

Essa arquitetura de múltiplas camadas é inspirada pelo nosso entendimento de biologia sobre cérebros onde existem várias conexões entre neurônios e eles comunicam-se entre si. A única diferença é que cada neurônio, no modelo computacional, não pode

se conectar com qualquer outro neurônio e ignorar a distância física. Então as redes de neurônios artificiais têm camadas discretas, conexões e direcionamento de propagação de dados.

Por exemplo, pode-se utilizar uma imagem, dividi-la em diversas matrizes de *pixel*, ou características, e as inserir na primeira camada da rede de neurônios. Essa camada faz suas tarefas, repassa as informações para a próxima camada e, caso faça parte da camada oculta, envia novamente as informações para a próxima camada, até que se alcance a camada de saída para gerar os resultados do algoritmo,

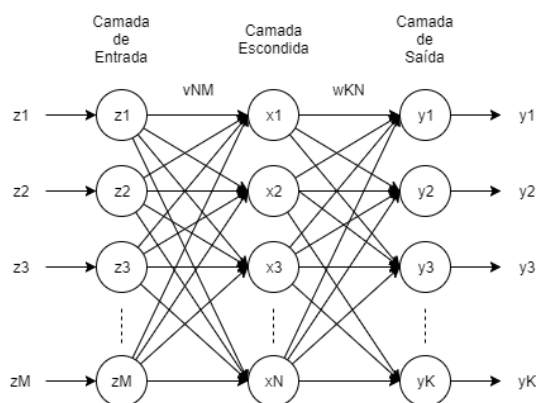


Figura 1: Arquitetura de exemplo para uma rede de neurônios com camadas de entrada, ocultas e de saída adaptado de [Isokawa et al., 2012].

2.4.1.4 Support Vector Machines

As *Support Vector Machines* [Li et al., 2011] combinam controle generalizado com técnicas para tratar diversas dimensões de dados. Para poder alcançar diferentes arquiteturas de modelos, é possível alterar o mapeamento de seu *kernel* com diversas funções, já implementadas, para que os dados sejam tratados de acordo com suas dimensões. Os métodos *kernel* realizam análise de padrões em todos os pares de dados formados no espaço gráfico em análise. Esses métodos podem ser utilizados para sequência de dados, gráficos, textos, imagens e, inclusive, vetores.

Para definir qual *kernel* seria o mais apropriado para a nossa dimensão de dados, o *framework* oferece a oportunidade de realizar comparações entre as diferentes funções disponíveis e escolher a melhor opção, ou seja, a que tenha melhores resultados, tanto para classificação como para regressão.

As funções disponíveis pela libSVM, uma biblioteca disponibilizada em [Chang and Lin, 2011] para integração *framework* Weka, permite a utilização de quatro funções de *kernel*: *linear*, *polynomial*, *radial basis function* e *sigmoid*. Ao alterar essas funções de mapeamento, é possível realizar comparações e determinar qual teve melhores resultados. Em nossos experimentos, utilizamos o *radial basis function*, pois temos dados não lineares e com poucas características. De acordo com [Pochet et al., 2004], caso existam um

grande número de características, ou atributos, o *kernel* ideal seria o linear.

2.4.1.5 *Regressão Linear*

É um algoritmo de regressão onde que o termo linearidade, na álgebra, se refere a uma relação linear entre duas ou mais variáveis. A regressão linear prediz uma variável dependente (y) baseada em uma variável independente (x). Portanto, esta técnica de regressão encontra a relação entre um valor de entrada x e um valor de saída y .

2.4.2 *Ensemble Learning*

O *ensemble learning* funciona como uma forma de comitê, utilizando-se de resultados de vários modelos preditivos, uma vez aplicados sobre a mesma base de dados, para atingir melhores resultados. O uso de técnicas para amostragem de dados é uma das formas chave para esse tipo de aprendizado com dados não balanceados. Essas técnicas envolvem replicar os dados utilizados para treinamento múltiplas vezes para montar múltiplos modelos, onde cada um aprende os seus limites de decisão ao comparados com outro sub-grupo de uma amostragem, pesos e sub-grupos de características. Essa necessidade ocorre muito com a bioinformática, pois essa área tem amostragem de dados muito limitados, uma vez que os dados provém de experimentos que, muitas vezes caros, têm muito tempo de coleta e processamento de amostras [YANG et al., 2014].

Dados não balanceados ocorrem quando há bases de dados que contêm muito mais instâncias de uma classe do que outra. No aprendizado de máquina, o *ensemble* de classificadores têm o objetivo de aumentar a acurácia dos mesmos ao combiná-los. No entanto, como cada classificador pode funcionar de uma forma diferente, o *ensemble* deve ser desenvolvido, geralmente, de forma específica para cada aplicação. O EN-MUTATE tem o objetivo de realizar classificações ternárias e binárias.

Em quesitos de aprendizado de máquina, a combinação de resultados ao invés de modelos individuais está baseado nos conceitos de estatística, computabilidade e representabilidade [DIETTERICH, 2000]. Onde que:

- Estatística: encontrar entre o espaço de hipóteses, a melhor decisão. No entanto, pode haver problemas com o aprendizado caso tenha muitas hipóteses diferentes ao utilizar os dados de treinamento.
- Computabilidade: divide dados para vários algoritmos de aprendizado realizarem uma busca local, onde que a saída de cada um desses algoritmos fornece uma solução aproximada do problema.
- Representabilidade: pode ocorrer, como acontece em várias aplicações de AM, não ter nenhuma solução verdadeira sendo representada por nenhuma das hipóteses. Em casos de amostras de treinamentos pequenos, esses algoritmos irão explorar apenas

um conjunto limitado de hipóteses e, caso encontre uma hipótese ajustável em relação aos dados de treinamento, a pesquisa por novas soluções será interrompida.

Com os algoritmos *ensemble*, descritos na seção seguinte, nossa metodologia aplica o *ensemble* de classificadores sobre os algoritmos de classificação introduzidos na seção anterior (2.4.1).

2.4.2.1 *Bagging*

O *Bagging*, ou *Bootstrap Aggregating*, treina classificadores diferentes com réplicas do grupo de dados alvo, ou seja, novos grupos de dados são criados ao criar instâncias randômicas dos dados originais [GALAR et al., 2012]. Com a combinação estatística de cada um dos algoritmos de classificação, ou regressão, é possível aprimorar a estabilidade e acurácia de cada um desses algoritmos, além de evitar o *overfit* do modelo.

2.4.2.2 *Random Forest*

O algoritmo *Random Forest* [Ho, 1998] é uma combinação de preditores de árvore de decisões, em que cada árvore depende dos valores de um vetor aleatório *bagging* (Seção 2.4.2.1) amostrado. Portanto, ao ser aplicado sobre os dados aleatórios formados pelas árvores do algoritmo *bagging*, o *Random Forest* calcula uma média das predições de cada uma dessas árvores para gerar um modelo final. A linha de execução deste algoritmo segue os seguintes passos:

Passo 1: Seleciona subgrupos (" N ") de dados da base de treinamento;

Passo 2: Treina " N " árvores de decisão para o mesmo número de subgrupos. Sendo que um subgrupo aleatório é utilizado para treinar uma árvore de decisão;

Passo 3: Cada árvore, individualmente e independentemente, prediz os valores na base de testes;

Passo 4: Realiza a predição final ao escolher a decisão mais votada em todas as árvores de decisão.

2.4.2.3 *AdaBoost*

O *Adaptive Boosting* é um algoritmo utilizado na conjunção de outros algoritmos de aprendizado de máquina para de aumentar o desempenho dos mesmos. Combina-se os resultados desses outros algoritmos, aplicando-se pesos, para representar a saída final de um classificador aprimorado (*boosted*). Ele é adaptativo no sentido em que os algoritmos mais fracos, combinados, são modificados com os pesos para que os valores classificados erroneamente influenciem menos na combinação final. Portanto, se os algoritmos têm resultados melhores do que predições aleatórias, permite o AdaBoost formar um modelo de aprendizado ótimo.

Seu algoritmo pode ser entendido com os seguintes passos:

Passo 1: Inicializa os pesos (w) de cada instância existente nos dados de treinamento, onde N , na 3, é o número de instâncias e T representará o número de iterações do algoritmo;

$$w_{ti} = \frac{1}{N} \quad (3)$$

Passo 2: Realiza o treinamento de uma árvore de decisão.

Passo 3: Calcula o peso da taxa de erro (e) dessa árvore de decisão. Onde e é o número de predições incorretas sobre N , levando em consideração o peso definido para cada instância.

Passo 4: Utilizando a equação 4, calcula-se o peso dessa árvore de decisão no agrupamento de árvores. Quanto maior for e , menor impacto terá esta árvore na decisão da predição final.

$$w_n = (\text{taxa de aprendizado}) * \log\left(\frac{1 - e_n}{e_n}\right) \quad (4)$$

Passo 5: Com o peso novo calculado, atualiza os pesos das instâncias incorretamente preditas nesta árvore de decisão. O novo peso da instância será igual ao peso antigo vezes o peso desta árvore;

Passo 6: Repete o Passo 2 até que o número de árvores de decisão definido seja alcançado;

Passo 7: Aplica os modelos na base de dados de testes e leva em consideração o peso de cada árvore em sua predição, sendo que árvores com maior peso terão mais impacto na decisão da predição.

2.4.3 Avaliação dos Algoritmos de Classificação e *Ensemble Learning*

Em teoria, os métodos de AM têm de passar por todos os dados várias vezes e encontrar padrões nas entradas e, baseando-se nesses padrões, poder prever casos que ocorrerão no futuro. Após ter novas predições, pode-se fazer com que o algoritmo passe novamente pelos dados, com novas entradas, e melhore sua performance [Shaikh, 2017].

Uma forma de avaliar os modelos preditivos gerados por algoritmos de classificação é a matriz de confusão. Conforme as definições de Kohavi and Provost [1998], ela é uma matriz $T \times T$, onde T é o tamanho da matriz ou o número de classes diferentes nos dados de treinamentos.

A partir dos dados em uma matriz de confusão, pode-se calcular uma série de valores que servem como avaliação do modelo de classificação, sendo que a diagonal principal

da matriz representa os elementos que correspondem às instâncias que foram preditas corretamente. Então, segue as definições:

- Verdadeiro Negativo: representa a taxa de instâncias negativas corretas (VN);
- Verdadeiro Positivo: representa a taxa de instâncias positivas corretas (VP);
- Falso Positivo: representa a taxa de instâncias positivas incorretas (FP);
- Falso Negativo: representa a taxa de instâncias negativas incorretas (FN).

2.4.3.1 Acurácia

$$A = \frac{VP + VN}{VP + VN + FP + FN} \quad (5)$$

A acurácia representa a relação de quantos resultados foram verdadeiros, tanto positivos quanto negativos, com o total de resultados na matriz de confusão. ‘

Além da acurácia, existem outras três métricas que auxiliam na avaliação do modelo e que, também, estão relacionadas com a matriz de confusão. Essas medidas auxiliares se dão necessárias para casos em que existem um número muito maior de verdadeiro negativos do que verdadeiro positivos, nos trazendo uma acurácia muito próxima do 100% que, nem sempre, representa a verdade sobre os dados preditos. Portanto, as métricas nas seções a seguir, verificariam o quão balanceado estariam os resultados.

2.4.3.2 Precisão

$$P = \frac{VP}{VP + FP} \quad (6)$$

A precisão é a relação de quantas instâncias são realmente positivas com quantas instâncias, classificadas como positivas, são negativas. Para determinados modelos, como por exemplo os detectores de mentira, é importante saber quantas positivos não são realmente negativos para que, o mesmo, seja um modelo preciso e se possa verificar quando a pessoa que está sendo avaliada pelo detector de mentiras não é um ótimo mentiroso.

2.4.3.3 Revocação

$$R = \frac{VP}{VP + FN} \quad (7)$$

Em contra partida à precisão, a revocação verifica a relação entre as instâncias realmente positivas com as instâncias positivas classificadas como negativas. Essa medida é importante em modelos que predizem classificações onde o positivo é muito importante, como a detecção de uma doença. Um exemplo seria um modelo que realizaria a predição

sobre se uma pessoa teria câncer caso o positivo fosse resultado, mas obteve uma predição errada com o falso negativo, o custo desse erro seria muito alto.

2.4.3.4 Medida-F

$$F = 2 * \frac{P * R}{P + R} \quad (8)$$

A Medida-F é necessária quando queremos descobrir o quão balanceadas estão a Precisão e a Revocação. É mais relevante em casos que os modelos retornam classificações desbalanceadas (muitos positivos e poucos negativos, ou vice-versa) e precisamos saber quantos falsos, negativos e positivos, influenciariam nos resultados preditos.

2.4.4 Avaliação dos Algoritmos de Regressão

Diferentemente dos algoritmos de classificação, os algoritmos de regressão possuem suas próprias métricas para avaliação e comparação entre algoritmos.

2.4.4.1 RMSE

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (predito - alvo)^2}{n}} \quad (9)$$

O *Root Mean Squared Error* é o desvio padrão dos erros de predição, ou resíduos. Resíduos são uma medida de quão longe da linha de regressão estão os dados preditos pelos modelos de predição. Portanto, o valor mais próximo de zero significa que as predições estavam mais próximas do valor de referência, ou atributo alvo, que seria a predição mais correta.

2.4.4.2 Coeficiente de Correlação

$$r = \frac{\sum_{i=1}^n (predito_i - predito_{mdia})(alvo_i - alvo_{mdia})}{\sqrt{\sum_{i=1}^n (predito_i - predito_{mdia})^2 \sum_{i=1}^n (alvo_i - alvo_{mdia})^2}} \quad (10)$$

O Coeficiente de Correlação utilizado neste trabalho é o de Pearson (r). Ele mede a força e direção da relação linear entre duas variáveis, no nosso caso seriam o atributo alvo e o valor predito pelos algoritmos de regressão, ou pelas ferramentas. Um valor de 1 seria uma relação positiva perfeita, onde que ambas estariam com aumentos positivos em seus valores. Já um coeficiente de -1 , significaria que as variáveis teriam uma relação inversa, onde que uma diminui da mesma forma que a outra aumenta. Valores de coeficiente maiores que 1 ou menores que -1 significaria a ocorrência de erro nas predições e, conseqüentemente, nos cálculos.

3 REVISÃO DA LITERATURA

Os trabalhos relacionados citados, nas seções a seguir, são as ferramentas em que foram submetidas nossa base de dados, com exceção do EN-MUTATE. Os resultados de predições, individualmente, serviram de entrada de nossa aplicação da metodologia *ensemble learning*. Todas essas ferramentas têm formulários de entradas semelhantes e, conseqüentemente, saídas semelhantes, que nos proporciona um controle melhor sobre os dados.

3.1 Trabalho relacionado 1: CUPSAT

O CUPSAT [Parthiban et al., 2006] é uma ferramenta online que analisa e prediz as mudanças de estabilidade de proteínas para mutações pontuais. Utiliza-se do ambiente estrutural específico de potenciais átomos e potenciais ângulos de torção, no desnovelamento, para prever o $\Delta\Delta G$. Requer a proteína no formato de arquivo do PDB e o local em que acontecerá a mutação para que possa nos retornar seus resultados. Seus resultados consistem em informar as características estruturais sobre a mutação desse local, como: acessibilidade do solvente, estrutura secundária e ângulos de torção.

Também analisa a habilidade de aminoácidos mutados de se adaptarem aos ângulos de torção. Utiliza diversos testes de validações (*split-sample*, *jack-knife* e *k-fold*) para garantir a confiabilidade, a precisão e a transferibilidade dos métodos de predição, que garantem uma acurácia de mais de 80% em todos os testes.

3.2 Trabalho relacionado 2: DUET

O DUET [Douglas E.V. Pires, 2014] é uma abordagem computacional integrada para prever mutações na estabilidade de proteínas. A aplicação tem o objetivo de combinar outras duas ferramentas, semelhante ao método de *ensemble learning*, também utilizadas no EN-MUTATEweb, para realizar predições e otimizar resultados. Esta ferramenta utiliza *Support Vector Machines* (SVMs), um algoritmo de classificação, para gerar seus modelos. Todo seu treinamento foi realizado por grupos de dados com pouca redundância e validados com grupos de dados aleatórios.

Sua aplicação é em formato *web*, onde recebe entradas de estruturas do PDB ou códigos de quatro letras para proteínas, informações da mutação e a cadeia identificadora. Seus resultados são apresentados em uma nova página com as previsões dos métodos individuais (SDM e mCSM) assim como os resultados combinados obtidos pelo DUET.

3.3 Trabalho relacionado 3: mCSM

No mCSM [Pires et al., 2013], cada mutação é representada por um vetor característico que é utilizado para treinar e testar métodos de previsão de aprendizado de máquina, tanto em regressão como classificação. Seus cálculos são realizados em duas etapas:

- Para um local de mutação, define-se o ambiente do resíduo de tipo nativo pelos átomos dentro de uma distância " r " de seu centro geométrico. Através de pares e cálculos das distâncias entre átomos desse ambiente gera uma matriz de distância de átomos, que contam para a grande gama de distâncias, tanto curtas como longas. Com essa matriz, padrões são extraídos e sintetizados como um vetor de características.
- Para entender as mudanças de átomos, é introduzido um contator de farmacóforos, que são a região da molécula de um ligante que está intimamente ligada ao seu receptor. Cada um dos vinte resíduos de aminoácidos são representados por um vetor diferente, onde cada posição demonstra a frequência de cada farmacóforo naquele resíduo. A diferença entre o estado nativo e os vetores de farmacóforos mutantes são adicionados no vetor de características da mutação.

Esses vetores de características são, depois, utilizados para definir qual padrão determinada mutação está seguindo, para então prever a variação de estabilidade da proteína.

3.4 Trabalho relacionado 4: SDM

Esta ferramenta [Catherine L. Worth, 2011] é uma função estatística de energia potencial com o objetivo de prever o efeito que polimorfismos de nucleotídeo único terão na estabilidade de proteínas. Utilizando-se das frequências de substituição de aminoácidos em ambientes específicos dentro de famílias de proteínas, a ferramenta calcula a pontuação de estabilidade que é análoga à variação de energia livre entre o estado nativo e mutante da proteína.

A ferramenta apresenta resultados melhores que outros métodos na questão de sensibilidade ao prever mutações, o que a torna útil para prever se uma mutação irá impactar a estrutura e formar uma doença.

3.5 Trabalho relacionado 5: MAESTRO

Com o objetivo de prever mudanças na estabilidade sobre mutações pontuais de proteínas, o MAESTRO [Laimer et al., 2015] baseia-se na estrutura da proteína e, embora tenha um poder de predição semelhante aos outros métodos, se diferencia por causa dos seguintes pontos:

- MAESTRO implementa um sistema de multiagentes para o aprendizado;
- disponibiliza os valores de $\Delta\Delta G$ e estimativa de confiabilidade de tal mutação;
- disponibiliza escaneamento em profundidade de mutações em vários pontos onde os tipos de mutação podem ser compreensivamente controlados;
- disponibiliza um modo específico para predições de ligações dissulfureto estabilizantes

É uma ferramenta versátil para nossa área e tem executáveis tanto para Linux quanto para Windows, além de ser liberado para a comunidade para fins não comerciais.

3.6 Trabalho relacionado 6: PoPMuSic

O PopMuSic [Dehouck et al., 2011] é um servidor web, de predição das mudanças termodinâmicas em mutações pontuais de proteínas, que utiliza uma combinação linear de potenciais estatísticos cujos coeficientes dependem da acessibilidade do solvente do resíduo mutado.

É uma ferramenta rápida, que permite realizar predições para todas possíveis mutações de uma proteína, tamanho médio, em menos de um minuto. Tem a possibilidade de detectar quais e quão ótimos são os aminoácidos de cada proteína, para que se possa fazer experimentos de mutações, demonstrando, também, fraquezas estruturais ao quantificar quanto esses locais são otimizados para função, da proteína, ao invés de estabilidade.

3.7 Trabalho relacionado 7: EN-MUTATE

A proposta EN-MUTATE [Alex, 2017] baseia-se no pressuposto de que a combinação de vários resultados para um mesmo propósito obtém, mais frequentemente, resultados mais satisfatórios do que decisões individuais [Zhou, 2012]. Seu principal foco foi a criação de um ferramenta online que reúne o resultado de outras 7 ferramentas, descritas em sua literatura [Alex, 2017]. Essas ferramentas são capazes de prever efeitos na estabilidade de uma proteína em ocorrências de mutações pontuais, que causam a variação da variação de energia livre $\Delta\Delta G$, o qual será o resultado resgatado das ferramentas.

Submit your job

Specify structure: PDB code or PDB file

PDB code:

Chain:

Amino acid (native):

Position:

Amino acid (mutant):

pH:

Temperature:

Ensemble algorithm:

$\Delta\Delta G$ classification:

This page will be automatically updated every 30 seconds.
If you wish to view these results at a later time, please bookmark this page.
Don't worry, the results will be kept on the server.

Comprehensive Prediction Results

Mutation site						
Protein	Chain	Amino acid (native)	Position	Amino acid (mutant)	pH	Temperature
1A23.pdb	A	HIS (H)	32	LEU (L)	7	25

Prediction parameters

Ensemble algorithm	$\Delta\Delta G$ classification
STACKING	TERNARY: Destabilizing, Neutral, Stabilizing

Predicted stability change

Ref.	Tool	$\Delta\Delta G$ (Kcal/mol)	Classification	Status
/	I-Mutant	-1.66	Destabilizing	Success
/	CUPSAT	-0.6	Destabilizing	Success
/	SDM	0.6	Stabilizing	Success
/	mCSM	-1.46	Destabilizing	Success
/	DUET	-0.3	Neutral	Success
/	iRDP	0.3	Neutral	Success
/	MAESTRO	-2.33	Destabilizing	Success
/	EN-MUTATE	∅	Destabilizing	Success

Figura 2: EN-MUTATEweb

No EN-MUTATE, foram executados experimentos de mineração de dados com técnicas de classificação sobre valores experimentais termodinâmicos que exemplificam o impacto que uma mutação pontual causa na estrutura de uma proteína. O funcionamento de EN-MUTATE se divide em quatro etapas:

- Etapa 1: Ferramentas de predição - Como pode ser visto na Tabela 1, esta etapa consiste em submeter experimentos nas ferramentas web (I-Mutant, CUPSAT, SDM, mCSM, DUET, iRDP e MAESTRO) para que as mesmas efetuem predições e devolvam um valor de $\Delta\Delta G$ da mutação informada.
- Etapa 2: Valores preditos - Através de *scripts*, o EN-MUTATE envia entradas para as ferramentas web de predição para, assim, obter os valores experimentais $\Delta\Delta G$.
- Etapa 3: Treino e teste dos classificadores - realiza-se um treinamento com os classificadores com técnica de aprendizado supervisionado. Como entrada, o atributo classe ($\Delta\Delta G$ experimental) é discretizado em intervalos referentes a estabilidade de proteínas, que são nomeados como: estabilizante, neutra e desestabilizante.
- Etapa 4: Classificação final - após obter modelos preditivos provenientes do treinamento com os classificadores, é possível realizar uma classificação sobre novas entradas na plataforma EN-MUTATEWeb.

Atributo	Descrição	Exemplo
ID	Identificador da mutação dentro do conjunto de dados	1
PDB	Código PDB da proteína	1A23
Mutação	Aminoácido nativo seguido da posição a ser alterada e o mutante	H32L
pH	pH do experimento	7
Temperatura	Temperatura do experimento	25

Tabela 1: Exemplo de entrada da metodologia EN-MUTATE para as ferramentas de predição

4 METODOLOGIA

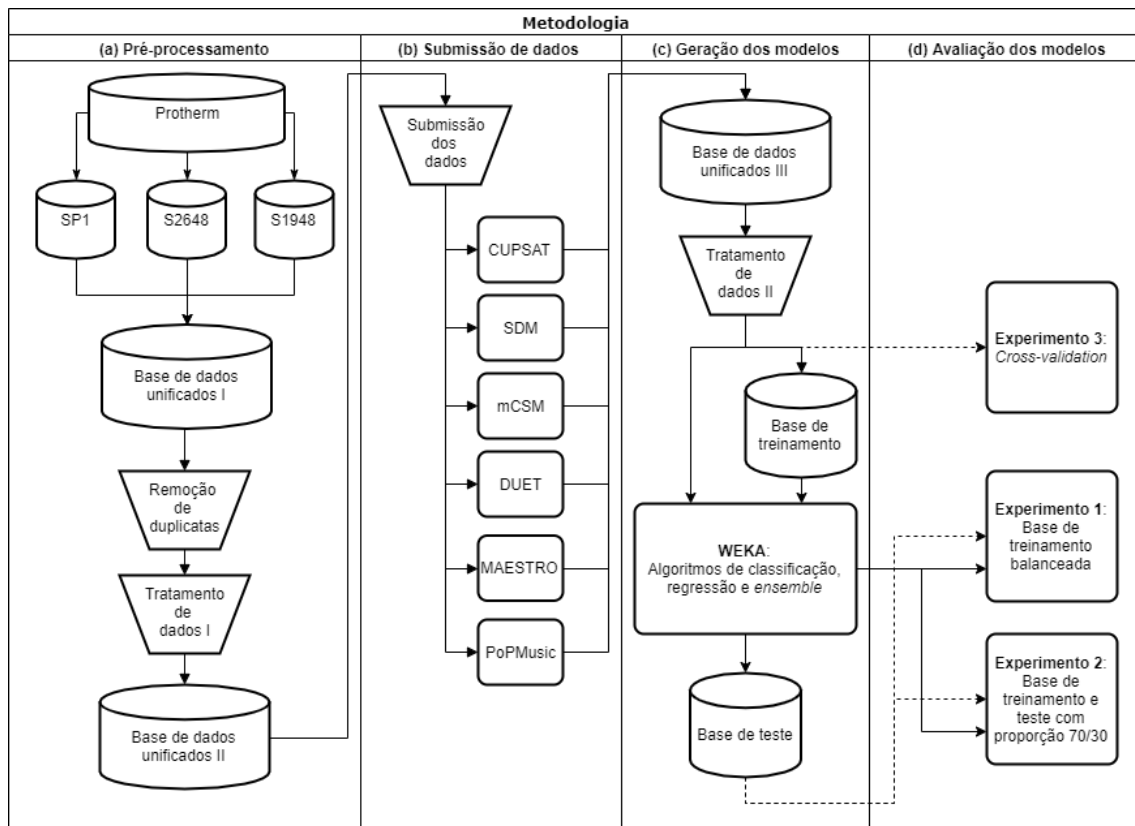


Figura 3: Fluxograma da metodologia

A figura 3 resume a metodologia proposta neste trabalho e suas etapas são mais detalhadas nas seções seguintes. Após a reunião de três bases de dados, necessitamos realizar o pré-processamento dos mesmos para que se possa obter uma base de dados unificada. Após o a realização dos tratamentos necessários, realiza-se a submissão da base de dados unificada em cada uma das seis ferramentas, previamente introduzidas na Seção 3 de literatura dos trabalhos estudados.

Obtendo os resultados de cada ferramenta, e seus devidos tratamentos, pode-se dar início à criação de nossos modelos de predição, utilizando-se de algoritmos de classificação, regressão e *ensemble*. Com a aplicação dos modelos sobre os dados de treinamento, podemos realizar nossas análises individuais de cada algoritmos e seus resultados.

4.1 Bases de dados

As bases de dados (SP1, S2648 e S1948), não disjuntas, demonstradas pela tabela 2, são agrupamentos de experimentos de mutações pontuais sobre a estrutura de proteínas, retirados da base disponibilizada pelo ProTherm [K. Abdulla Bava and Sarai, 2004].

O ProTherm [K. Abdulla Bava and Sarai, 2004] é uma base de dados termodinâmicos para proteínas e mutações, contendo mais de 300 atributos de diversos parâmetros termodinâmicos para proteínas tipo-selvagem e proteínas mutantes. No entanto, são necessários apenas alguns desses parâmetros, pois estamos analisando apenas mutações pontuais sobre estruturas de proteínas. Nosso parâmetro principal, e atributo alvo, é a variação da energia livre de Gibbs ($\Delta\Delta G$), que é importante para o entendimento do mecanismo de estabilização das proteínas.

Base de Dados	Instâncias	Proteínas	Estabilizantes	Desestabilizantes
SP1	2.648	99	2.046	602
S2648	2.648	100	568	2.080
S1948	1.948	58	562	1.386
Total	7.244	298	3.738	4.068

Tabela 2: Número de instâncias, proteínas únicas, classes estabilizantes e classes desestabilizantes por base de dados.

Esses agrupamentos de dados da grande base ProTherm, foram utilizados, ou criados, por outras ferramentas, com objetivos semelhantes ao nosso, para o treinamento de seus modelos. O SP1 e o S2648 foram utilizadas pelas ferramenta PoPMuSic [Dehouck et al., 2011], SDM [Catherine L. Worth, 2011] e MAESTRO [Laimer et al., 2015], enquanto que a S1948 foi utilizada pela ferramenta I-Mutant [Capriotti et al., 2004]. Embora o I-mutant não esteja mais incluído em nosso *ensemble* de ferramentas, devido a problemas técnicos na submissão das mutações em sua aplicação *web*, é uma ferramenta que se assemelha com nossos objetivos de realizar predições sobre mutações pontuais.

Como pode ser analisado na Tabela 2, conseguimos agrupar uma base de dados balanceada no quesito $\Delta\Delta G$ para a determinação de mutações estabilizantes e desestabilizantes. Para que possamos utilizar esses dados para a criação de modelos, é necessário a submissão dos mesmos, depois de um pré-processamento, para as ferramentas que compõem nosso *ensemble* de resultados.

4.2 Pré-processamento

Como pré-processamento de nossos dados, precisamos unificar as três bases (SP1, S2648 e S1948) em apenas uma, para que possamos submeter nas ferramentas escolhidas.

As três bases de dados têm formatos diferentes de apresentar seus atributos e dados, além de conter diversas duplicatas de experimentos.

Primeiramente, realizou-se a unificação dos dados através de transformações e substituições de dados. As tabelas 3 até 8 demonstram exemplos do antes e depois dos dados de cada base de dados. Essa unificação de dados está representado pela "Base de dados unificados I" no fluxograma 3.

Como pode ser observado na Tabela 3, a *Variation* deve ser dividida em 4 atributos diferentes para que se possa submeter nos formulários das ferramentas. A *Variation* se torna *CHAIN*, *NATIVEAA*, *POSITION* e *MUTATION*, exemplificado pela Tabela 4. As tabelas seguintes também necessitam de separação dos valores de *Variation*, para que possam se tornar *NATIVEAA*, *POSITION* e *MUTATION*, uma vez que a *CHAIN* já é disponibilizada por outro atributo.

PDB	Variation	ddG	pH	T
1a5e	W15.A{D}	-0.19	8.5	20
1a5e	L37.A{S}	-0.81	8.5	20

Tabela 3: Exemplo de entrada da base de dados SP1

PDBID	CHAIN	NATIVEAA	POSITION	MUTATION	DDG	PH	TEMP
1a5e	A	TRP.W	15	ASP.D	-0.19	8.5	20
1a5e	A	LEU.L	37	SER.S	-0.81	8.5	20

Tabela 4: Resultado do pré-processamento de SP1

PDB	Variation	ddG	pH	T	CHAIN
1A43A	G156A	-2.4	7.3	25	A
1A43A	E159D	-4.55	7.3	25	A

Tabela 5: Exemplo de entrada da base de dados S2648

PDBID	CHAIN	NATIVEAA	POSITION	MUTATION	DDG	PH	TEMP
1A43	A	GLY.G	156	ALA.A	-2.4	7.3	25
1A43	A	GLU.E	159	ASP.D	-4.55	7.3	25

Tabela 6: Resultado do pré-processamento de S2648

PDB	Variation	ddG	pH	T
1A23	H32L	4.6	7	30
1A23	H32L	5.3	7	30

Tabela 7: Exemplo de entrada da base de dados S1948

PDBID	CHAIN	NATIVEAA	POSITION	MUTATION	DDG	PH	TEMP
1A23	A	HIS.H	32	LEU.L	4.6	7	30
1A23	A	HIS.H	32	LEU.L	5.3	7	30

Tabela 8: Resultado do pré-processamento de S1948

Após o pré-processamento das bases e o agrupamento de todos os dados, podemos remover duplicatas de experimentos iguais. Como cada experimento é realizado diversas vezes por diferentes especialistas, às vezes os mesmos, os resultados de suas medições reais do $\Delta\Delta G$ podem variar de experimento para experimento.

Uma vez que experimentos classificados como estabilizantes são mais raros, de acordo com [Luscombe et al., 2001], no meio de pesquisa de elementos biológicos, os experimentos que estejam com variações entre estabilizante e desestabilizante, foi determinado que os experimentos desestabilizantes deveriam permanecer em nossa base de dados.

Base de Dados	Instâncias	Proteínas	Estabilizantes	Desestabilizantes
SP1, S2648, S1948	7.244	298	3.738	4.068
MP3904	3.904	151	951	2.953

Tabela 9: Comparação de dados antes e depois do pré-processamento.

Com a unificação e a remoção de duplicatas, formamos a "Base de dados unificados II", representado pela Figura 3 (a). Como pode ser analisado na Tabela 9, denominamos nossa base de dados como MP3904.

4.3 Submissão de dados

Com os dados devidamente preparados, iniciamos a submissão, representado pela coluna (b) da Figura 3, dos mesmos em cada uma das ferramentas escolhidas. No entanto, todas as ferramentas, aqui utilizadas, necessitam uma submissão manual de cada uma das entradas. Utilizamos o *plugin* do *browser Firefox* denominado iMacros para a realização dessa tarefa.

O iMacros é uma linguagem em formato de *script*, que permite usuários simular atividades online para preenchimento de formulários, *uploads* e *downloads* de imagens e arquivos, ou até importar e exportar informações de bases de dados, arquivos CSV, arquivos XML, entre outros. Esse formato de manipulação de dados é realizado através de *scripts* elaborados especificamente para essas ferramentas.

A única ferramenta que tem um formato de submissão diferente é o PoPMuSic, que nos permite submeter apenas o código PDB de cada uma das 151 proteínas, nos retornando resultados de todas as mutações possíveis para cada proteína. O resultado de todas as ferramentas foram devidamente armazenadas em um banco de dados, local, para que

possa ser extraído e tratado para as seguintes etapas de nossa metodologia.

4.4 Geração dos modelos

Para a geração dos modelos, realizou-se três experimentos com metodologias diferentes em relação a separação de dados: O experimento 1 utiliza uma base de treinamento balanceado para estabilizantes e desestabilizantes; O experimento 2 utiliza uma base de treinamento e testes com proporção 70% e 30%, respectivamente; O experimento utiliza o método de *cross-validation* para a criação de bases de treinamento e testes na geração de modelos.

Diferentemente da metodologia EN-MUTATEweb [Alex, 2017], em que os resultados das ferramentas são discretizados entre estabilizante ou desestabilizante, utilizou-se o valor $\Delta\Delta G$ real predito pelas ferramentas. Portanto, após o "Tratamento de Dados II", como pode ser visto na coluna (c) da Figura 3, iniciou-se a geração de modelos de predição através do *software* Weka e, também, a criação de bases de treinamentos e testes (Tabela 10) que são utilizadas pelos experimentos apresentados nos resultados. O experimento 3 tem quantidades de instâncias para treinamento e testes criados aleatoriamente pelo modelo de validação cruzada, portanto não destacou-se os valores quantitativos das instâncias.

Exp.	Treinamento			Teste		
	estab.	desestab.	total	estab.	desestab.	total
Exp. 1	500	500	1000	451	2453	2904
Exp. 2	666	2066	2732	285	887	1172

Tabela 10: Número de instâncias, estabilizantes e desestabilizantes, para bases de treinamento e testes nos experimentos 1 e 2. O experimento 3 segue o algor de validação cruzada para treinamento e testes.

Quanto à configuração de parâmetros dos algoritmos, foram utilizados seus valores padrões, já configurados pelos próprios desenvolvedores dos mesmos. No entanto, alguns dos algoritmos tiveram exceções em alguns de seus parâmetros, sejam eles:

- **LibSVM:** houve a necessidade de alterar o *kernel* para *radial basis function* e alterar o *SVMType* entre C-SVC, para função de classificação, ou nu-SVR, para função de regressão;
- **Bagging:** Como é um *ensemble* de classificadores, foi necessário escolher sobre qual classificador iria trabalhar. Portanto, o parâmetro *classifier* foi alternado entre: LibSVM, J48, MultiLayerPerceptron e NaiveBayes;
- **Boosting,** ou AdaboostM1: Da mesma forma que o *Bagging*, alternou-se o parâmetro *classifier* entre: LibSVM, J48, MultiLayerPerceptron e NaiveBayes.

5 RESULTADOS

Neste capítulo serão apresentados os resultados de todos os algoritmos de *ensemble* e sua comparação com as ferramentas individuais. Em ambos os tipos de *ensemble*, foram efetuados os cálculos sobre as ferramentas em relação a apenas os dados de testes, a fim de obtermos comparações mais confiáveis.

5.1 Experimento 1: Base de treinamento balanceada

Com a importância de conseguir prever mutações estabilizantes, para a criação de drogas e remédios, priorizamos uma base de treinamento para os modelos com dados balanceados. Assim, os modelos terão informações balanceadas de ambos os tipos de classes para que possa realizar previsões estabilizantes mais corretamente.

Os modelos de classificação e *ensemble* terão seu foco na acurácia e precisão para a classificação mais correta de mutações estabilizantes. Enquanto que os modelos de regressão têm o mesmo objetivo que os outros experimentos, onde precisamos prever o $\Delta\Delta G$ com um RMSE baixo e um coeficiente de correlação mais próximo de 1.

Como visto na Tabela 9, existe uma grande discrepância nos totais de classes em nossa base MP3904. A utilização de uma base de treinamento com mesmo número entre as duas classes pode criar modelos de previsão que alcancem bons resultados para a previsão de mutações pontuais em estruturas de proteínas que tenha um $\Delta\Delta G$ que sejam estabilizantes. Portanto, para que se tenha modelos devidamente balanceados, criou-se uma base de treinamento com 500 de cada classe onde, cada uma dessas instâncias, foram escolhidas de forma aleatória, totalizando 1000 instâncias para o treinamento. A base de testes restante é formada pelas 2.904 instâncias restantes.

Para comparar os resultados de nossos modelos, foi efetuado o cálculo das métricas sobre os de todas as ferramentas individuais de acordo com nossa base de testes, assim como sobre os resultados de nossos modelos.

5.1.1 Algoritmos de Classificação e *Ensemble*

Para os resultados das métricas de avaliação das seis ferramentas, que compõem nossas instâncias, percebe-se na Tabela 11 que todas as ferramentas obtiveram acurácias boas ao serem calculadas sobre nossa base de testes. A base de testes é composta por 2.904 instâncias, onde 451 são estabilizantes e 2.453 são desestabilizantes.

Ferramentas de Classificação	Acurácia	Precisão	Revocação	Medida-F
CUPSAT	75,55%	0,31	0,54	0,39
SDM	65,29%	0,23	0,60	0,33
mCSM	81,96%	0,35	0,31	0,34
DUET	78,17%	0,34	0,58	0,43
MAESTRO	70,01%	0,24	0,53	0,34
PoPMusic3,1	78,68%	0,29	0,35	0,32

Tabela 11: Métricas de avaliação para a classificação das ferramentas do Experimento 1

Levando em consideração que este experimento tinha o objetivo de conseguir precisões altas, as precisões, as revocações e as medidas-F das ferramentas obtiveram resultados muito abaixo do padrão aceitável, ao se compararem com os algoritmos aplicados por este trabalho.

Tanto os algoritmos de classificação (Tabela 12), como os algoritmos de *ensemble* de classificadores (Tabela 13), obtiveram valores bons de acurácia e precisão, com destaque para o LibSVM (75,52% e 0,76), o Multilayer Perceptron (74,4% e 0,76) e o Random Forest (80,92% e 0,70).

Algoritmos de Classificação	Acurácia	Precisão	Revocação	Medida-F
NaiveBayes	63,40%	0,84	0,26	0,40
MultilayerPerceptron	74,4%	0,76	0,32	0,46
LibSVM (radial)	75,52%	0,76	0,34	0,47
J48	76,96%	0,70	0,35	0,47

Tabela 12: Métricas de avaliação para a classificação dos algoritmos do Experimento 1

5.1.2 Algoritmos de Regressão

Utilizando-se do atributo alvo em formato quantitativo, realizamos os cálculos das métricas de avaliação do RMSE e coeficiente de correlação, novamente, para os dados de testes de 2.904 instâncias. Nota-se que as ferramentas mCSM, DUET e PoPMuSic3.1, na Tabela 14, obtiveram valores semelhantes ao que foi predito pelos algoritmos de regressão aplicados, presentes na Tabela 15.

Quando se analisa o atributo alvo $\Delta\Delta G$ em nossa base de dados MP3904, temos um desvio padrão de 1,61, portanto um RMSE abaixo desse valor é um resultado positivo. No entanto, nenhum algoritmo, ou ferramenta obteve um coeficiente alto o suficiente para

<i>Ensemble de Classificadores</i>	Acurácia	Precisão	Revocação	Medida-F
Random Forest	80,92%	0,70	0,40	0,51
Bagging (LibSVM)	75,79%	0,76	0,35	0,48
Bagging (J48)	75,41%	0,76	0,34	0,47
Bagging (MultilayerPerceptron)	74,45%	0,75	0,33	0,46
Bagging (NaiveBayes)	63,71%	0,84	0,26	0,40
Boosting (LibSVM (radial))	70,49%	0,79	0,30	0,44
Boosting (J48)	75,21%	0,71	0,33	0,45
Boosting (MultilayerPerceptron)	74,07%	0,76	0,33	0,46
Boosting (NaiveBayes)	69,94%	0,80	0,30	0,43

Tabela 13: Métricas de avaliação para os algoritmos de *Ensemble de Classificadores* do Experimento 1

Regressão das Ferramentas	<i>RMSE</i>	<i>Coefficiente de Correlação</i>
CUPSAT	1,90	0,48
SDM	1,65	0,48
mCSM	1,25	0,66
DUET	1,26	0,67
MAESTRO	1,46	0,56
PoPMusic3.1	1,27	0,66

Tabela 14: Métricas de avaliação para a regressão das ferramentas do Experimento 1

uma maior confiabilidade nos modelos, uma vez que o coeficiente indica a relação linear entre o atributo alvo e o valor predito.

Regressão dos Algoritmos	<i>RMSE</i>	<i>Coefficiente de Correlação</i>
LinearRegression	1,25	0,65
MultilayerPerceptron	1,18	0,66
LibSVM (radial)	1,26	0,63

Tabela 15: Métricas de avaliação para a regressão dos algoritmos do Experimento 1

5.2 Experimento 2: Base de treinamento e teste com proporção 70/30

Para estudar o comportamento das ferramentas individuais e nossos modelos com dados não balanceados, preparou-se uma base de treinamento com 70%, de nossa base MP3904, enquanto que os testes foram realizados sobre os 30% restantes das instâncias. As instâncias são separadas aleatoriamente, para que não tenha uma sequência de dados semelhantes de mesmas proteínas, podendo causar distorção na geração dos modelos preditores.

Neste experimento, temos o objetivo de analisar o comportamento dos modelos de predição para uma base de treinamento muito maior que a base de testes. A base de treinamento é formada por 2.732 instâncias, 666 estabilizantes e 2.066 desestabilizantes,

enquanto que a base de testes é formada por 1.172, 285 estabilizantes e 887 desestabilizantes.

Com dados desbalanceados, pendendo para os desestabilizantes, espera-se que os algoritmos de classificação e os *ensemble* de classificadores obtenham uma revocação mais alta, uma vez que será mais fácil de identificar mutações desestabilizantes, por termos mais características apontado para essa classe.

Os modelos de classificação e *ensemble* podem ser avaliados por todas as métricas citadas na Seção 2.4.3, para comparação com as ferramentas individuais sobre as mesmas instâncias separadas para testes.

5.2.1 Algoritmos de Classificação e *Ensemble*

Ao analisar a Tabela 16, com os resultados das ferramentas de predição, percebe-se que, as mesmas, obtiveram resultados bem variados. Observa-se que algumas ferramentas (mCSM e PoPMuSic3.1) conseguiram classificar mais mutações estabilizantes do que desestabilizantes, enquanto que outras (SDM e MAESTRO) o vice-versa, assim como algumas obtiveram precisão e revocação balanceadas. No entanto, nenhuma obteve bons valores para as métricas, com excessão da acurácia.

Ferramentas de Classificação	Acurácia	Precisão	Revocação	Medida-F
CUPSAT	72,18%	0,43	0,51	0,47
SDM	66,98%	0,38	0,60	0,47
mCSM	78,07%	0,57	0,36	0,45
DUET	77,05%	0,52	0,59	0,56
MAESTRO	68,43%	0,40	0,60	0,48
PoPMusic3.1	75,34%	0,49	0,42	0,45

Tabela 16: Métricas de avaliação para a classificação das ferramentas do Experimento 2

Já, ao analisar as Tabelas 17 e 18, encontramos um padrão de resultados esperados, que seriam o valor de revocação mais alto em relação ao de precisão, uma vez que nosso treinamento do modelo teve mais acesso a mutações desestabilizantes. Com excessões do algoritmo de classificação NaiveBayes e do *ensemble* do classificador *Bagging* Naive Bayes, todos obtiveram uma acurácia superior a 80% e com revocação chegando a até 0,85.

Algoritmos de Classificação	Acurácia	Precisão	Revocação	Medida-F
NaiveBayes	72,78%	0,78	0,46	0,58
MultilayerPerceptron	83,28%	0,43	0,77	0,56
LibSVM (radial)	84,13%	0,47	0,79	0,59
J48	81,83%	0,35	0,77	0,49

Tabela 17: Métricas de avaliação para a classificação dos algoritmos do Experimento 2

<i>Ensemble de Classificadores</i>	Acurácia	Precisão	Revocação	Medida-F
Random Forest	83,79%	0,40	0,85	0,55
Bagging (LibSVM)	83,79%	0,45	0,80	0,57
Bagging (J48)	83,87%	0,52	0,74	0,61
Bagging (MultilayerPerceptron)	84,30%	0,56	0,73	0,63
Bagging (NaiveBayes)	72,78%	0,78	0,46	0,58
Boosting (LibSVM (radial))	84,64%	0,52	0,77	0,62
Boosting (J48)	81,83%	0,60	0,63	0,62
Boosting (MultilayerPerceptron)	83,28%	0,44	0,78	0,56
Boosting (NaiveBayes)	80,03%	0,67	0,58	0,62

Tabela 18: Métricas de avaliação para os algoritmos de *Ensemble de Classificadores* do Experimento 2

5.2.2 Algoritmos de Regressão

Assim como no primeiro experimento, a classificação do atributo alvo não influencia nos resultados dos algoritmos de regressão, que utilizam valores quantitativos. Neste experimento, os algoritmos de regressão na Tabela 20 obtiveram um valor de coeficiente superior a 0,7, o que já aumenta a confiabilidade no modelo de predição. Assim como as ferramentas mCSM e DUET obtiveram bons resultados para o RMSE e o coeficiente de correlação.

Regressão das Ferramentas	RMSE	Coeficiente de Correlação
CUPSAT	1,90	0,40
SDM	1,53	0,49
mCSM	1,15	0,67
DUET	1,13	0,69
MAESTRO	1,37	0,55
PoPMusic3.1	1,18	0,64

Tabela 19: Métricas de avaliação para a regressão das ferramentas do Experimento 2

Regressão dos Algoritmos	RMSE	Coeficiente de Correlação
LinearRegression	1,09	0,70
MultilayerPerceptron	1,32	0,71
LibSVM (radial)	1,05	0,73

Tabela 20: Métricas de avaliação para a regressão dos algoritmos do Experimento 2

5.3 Experimento 3: *Cross-validation*

O *cross-validation* é uma técnica estatística para analisar como os resultados se comportam com grupos de dados independentes. Com essa técnica, é possível criar modelos

mais acurados dos algoritmos de classificação, regressão e *ensemble*. São utilizados, em toda sua dimensão, os dados de nossa base MP3904.

Foi utilizado o *cross-validation* de 20 iterações, sendo cada iteração um particionamento da base de dados em diversas partes para, então, utilizar uma das partes como base de treinamento, criar o modelo e validar suas previsões sobre outra parte da base. A técnica aplica esse mesmo método em partes diferentes para cada iteração, previamente determinada como parâmetro na função do algoritmo. Após realizar todas as análises, os resultados de todas as iterações são combinadas para criar uma estimativa do desempenho do modelo.

O Experimento 3, diferentemente dos outros dois experimentos, não tem a capacidade de se comparar com os valores das ferramentas, já que a *cross-validation* cria vários subgrupos de treinamento e testes, de acordo com o número de iterações, para chegar em um consenso que possa ser aplicado no modelo de previsão.

5.3.1 Algoritmos de Classificação e *Ensemble*

Ao comparar os algoritmos de classificação (Tabela 21) com o *ensemble* de classificadores (Tabela 22), obtivemos acurácias altas e maiores que 80%, mas, também, obtivemos variações nos resultados de precisão e revocação. Alguns com mais precisão que revocação, como é o caso do LibSVM, o Random Forest e o *Bagging* (NaiveBayes), outros mais balanceados, como o *Bagging* (J48), o *Boosting* (J48) e o *Boosting* (NaiveBayes), e alguns com mais revocação que precisão, nos algoritmos *Bagging* (LibSVM e Multilayer-Perceptron).

Algoritmos Classificação	Acurácia	Precisão	Revocação	Medida-F
NaiveBayes	72,75%	0,78	0,46	0,58
MultilayerPerceptron	81,02%	0,48	0,64	0,55
LibSVM (radial)	82,43%	0,73	0,44	0,55
J48	81,74%	0,48	0,67	0,56

Tabela 21: Métricas de avaliação para a classificação dos algoritmos do Experimento 3

5.3.2 Algoritmos de Regressão

Como pode se observar na Tabela 23, os algoritmos de regressão obtiveram, mais uma vez, resultados razoáveis com RMSE relativamente baixos e um coeficiente de correlação próximo de 0,7.

<i>Ensemble de Classificadores</i>	Acurácia	Precisão	Revocação	Medida-F
Random Forest	83,61%	0,85	0,39	0,54
Bagging (LibSVM)	82,38%	0,44	0,73	0,55
Bagging (J48)	83,66%	0,50	0,74	0,60
Bagging (MultilayerPerceptron)	82,02%	0,48	0,69	0,56
Bagging (NaiveBayes)	72,69%	0,78	0,46	0,58
Boosting (LibSVM (radial))	81,48%	0,46	0,68	0,55
Boosting (J48)	82,22%	0,57	0,66	0,61
Boosting (MultilayerPerceptron)	80,87%	0,50	0,64	0,56
Boosting (NaiveBayes)	78,10%	0,66	0,54	0,60

Tabela 22: Métricas de avaliação para os algoritmos de *Ensemble de Classificadores* do Experimento 3

Regressão dos algoritmos	<i>RMSE</i>	<i>Coefficiente de Correlação</i>
LinearRegression	1,18	0,67
MultilayerPerceptron	1,24	0,63
LibSVM (radial)	1,14	0,70

Tabela 23: Métricas de avaliação para a regressão dos algoritmos do Experimento 3

6 DISCUSSÃO

Para melhor entender os experimentos, é importante analisar como se comportam as métricas de avaliação perante as matrizes de confusão. Para modelos de classificação, a acurácia, a precisão e a revocação são calculadas a partir dos valores de verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo, como definido na seção 2.4.3.

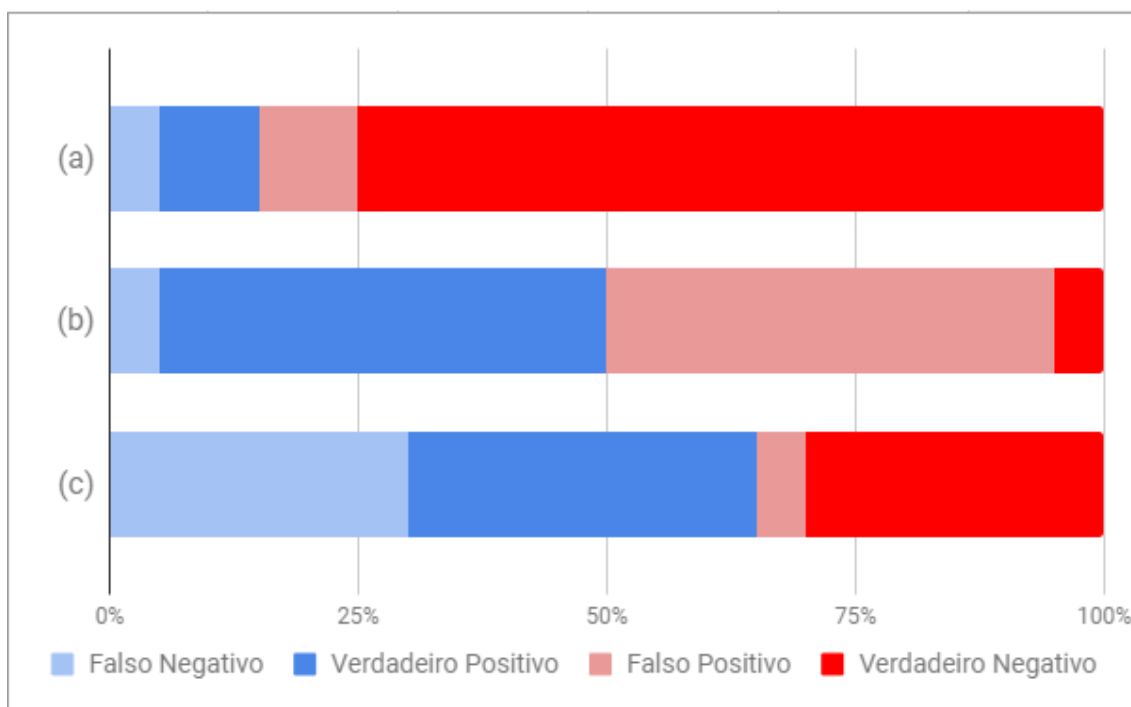


Figura 4: Representação de relevância de valores da matrizes de confusão para os cálculos das métricas de avaliação: (a) Acurácia alta, Precisão baixa e Revocação alta; (b) Acurácia baixa, Precisão baixa e Revocação alta; (c) Acurácia alta, Precisão alta e Revocação baixa.

Ao considerar Negativos como classificações desestabilizantes e Positivos como classificações estabilizantes, a figura 4 representa a quantificação das predições, que podem afetar os resultados. Este gráfico utiliza dados meramente ilustrativos em prol da análise de dados.

No caso (a), grande parte dos dados são desestabilizantes e grande parte deles foram

classificados corretamente como Verdadeiros Negativos, portanto a acurácia é alta, por ser uma relação entre VP e VN sobre todas as classificações. No entanto, os VP's e FP's têm valores próximos, causando um valor de precisão baixo. Precisão baixa implica que grande porcentagem das mutações classificadas como estabilizantes são, na verdade, desestabilizantes. Essa incoerência pode acabar atrasando ou impedindo avanços na criação de novas drogas.

No caso (b), obtêm-se tanto acurácia, como precisões baixas, uma vez que a predição de Positivos e Negativos acertam apenas 50% das vezes. Modelos que tenham esses resultados, não são ideais para este propósito.

O caso (c) representa algo próximo do ideal para o objetivo deste trabalho, onde a acurácia e a precisão seriam altas, uma vez que temos uma relação de VP e FP favorável, exemplificando maior quantidade de acertos em classificações estabilizantes.

Classificar corretamente as mutações estabilizantes que auxiliam os profissionais da área a descobrirem componentes seletos no descobrimento de drogas e remédios para doenças e patogêneses, nota-se que as métricas de avaliação mais interessantes para a predição de mutações pontuais seriam a acurácia e a precisão do modelo de predição.

Ao avaliar as tabelas 11, 12 e 13 no primeiro experimento, percebe-se que as ferramentas utilizadas como entrada de nossos algoritmos de classificação, têm tanto acurácias quanto precisões altas.

Como podemos, de certa forma, ignorar os valores de revocação, que indica a porcentagem de estabilizantes que foram classificados erroneamente como desestabilizantes, a medida-F acaba, por si só, não tendo muito impacto em nossa comparação de modelos de predição para o primeiro experimento.

Com essas observações, percebe-se uma grande melhora na precisão dos algoritmos ensemble de classificação em relação às ferramentas individuais, comprovado pelos resultados dos modelos ao obterem uma acurácia e precisão maiores do que 70%.

O segundo experimento demonstra que com uma grande disparidade de dados para as duas classes, os modelos tendem a reconhecer mais características de apenas uma classe, uma vez que faltam informações para a outra. Com a divisão de 70 para 30 nas bases de treinamento e testes, respectivamente, houve uma melhora na acurácia, se comparado com o primeiro experimento que teve uma base de treinamento pequena. Com o balanceamento dos dados de treinamento e, conseqüentemente, mais dados para o mesmo, pode-se concluir que os modelos de predição podem ser melhorados.

O terceiro experimento utiliza o *cross-validation*, que tem o objetivo de enfrentar o problema de desbalanceamento de dados ao utilizar vários subgrupos de dados, utilizados como treinamento e testes, para que se possa chegar em um consenso mais confiável para a geração de novos modelos de predição.

Os algoritmos de regressão, que retira a limitação de classes e foca na predição de valores $\Delta\Delta G$, é consistente nas aplicações de seus algoritmos nas métricas de RMSE e

Coefficiente de Correlação em todos os experimentos. Podem ser utilizados primeiramente para que, então, possam ser feitas as classificações do $\Delta\Delta G$.

Outro ponto a ser analisado é o fato de que as ferramentas de predição obtiveram, na maioria das vezes, resultados relativamente ruins. Quando comparado com os resultados de 5, das 6 ferramentas, utilizadas pelo trabalho de [Alex, 2017]. Percebe-se que, para os dados utilizados no EN-MUTATE, as ferramentas obtiveram bons resultados em todas as métricas de avaliação aqui utilizadas. Isso pode se dar pelo fato de que, para bases de dados diferentes, as ferramentas são, deveras, inconsistentes.

A seção a seguir, propõe uma outra discussão, realizada com o intuito de estudar como a variação na definição do limiar, que separa desestabilizantes ($\Delta\Delta G \leq 0$) de estabilizantes ($\Delta\Delta G > 0$) pode afetar o comportamento dos modelos de predição.

6.1 Análise de limiares

Levando em consideração que o trabalho está aberto a melhoras, é importante que tenhamos formas de obter resultados melhores e mais aprimorados na criação de modelos de predição, não só para modelos de técnica *ensemble*, como também para diferentes modelos. Com esse intuito, fizemos uma análise de como os limiares, que definem quando um $\Delta\Delta G$ é desestabilizante ou estabilizante, podem influenciar no comportamento dos modelos de predição.

O seguinte fluxograma representa o estudo realizado sobre mutações preditas pelo EN-MUTATEweb [Alex, 2017] e como seus resultados seriam diferentes, ou não de acordo com o limiar escolhido.

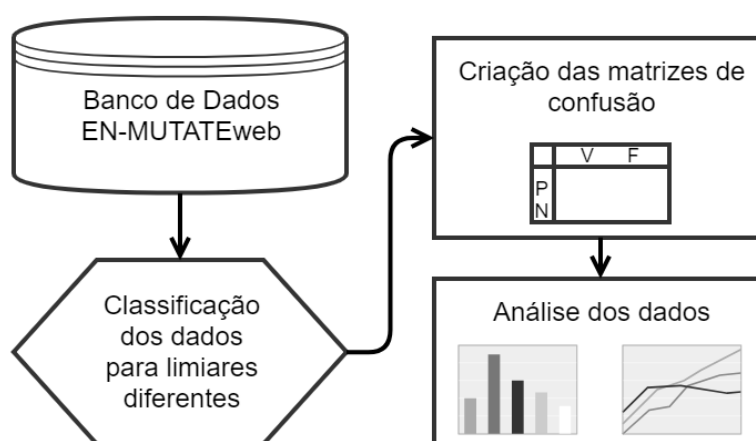


Figura 5: Fluxograma da análise de limiares

O SDM [Catherine L. Worth, 2011], que está entre as ferramentas utilizadas, é um exemplo de utilização de um limiar diferente do padrão de 0 (zero). Essa ferramenta utiliza a pontuação de 2 kcal mol^{-1} em seu $\Delta\Delta G$ para definir se a mutação será estabilizante

(maior que 2) ou desestabilizante (menor ou igual a 2).

Os gráficos na Figura 6, representam quantas ocorrências, de cada um dos elementos de uma matriz de confusão (verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos), existem nas 3.432 instâncias da base de dados. Os limiares utilizados para a análise são definidos empiricamente e têm uma variação de -4 (quatro negativo) até 4 (quatro) com passos de $0,5$ entre cada limiar, formando uma seleção de 17 limiares diferentes para análise de como os resultados podem se alterar.

Verificando o gráfico novamente, percebe-se que as ferramentas têm um comportamento similar quanto a sua predição, mas, no entanto, percebe-se também que quanto mais verdadeiros positivos (estabilizante) obtemos, menos falsos negativos (desestabilizantes) alcançamos, ou vice-versa. Vale ressaltar que os verdadeiros negativos seriam as classificações de desestabilizante quando deveriam ser estabilizantes e, os falsos positivos, seriam a opção contrária à anterior.

Os motivos dessa inversão de valores podem ser resultado do grande desbalanceamento nas mutações da base de dados de entrada, onde, em grande maioria, são desestabilizantes. Com o desbalanceamento entre essas duas classes, os modelos de predição das ferramentas tendem a, algumas vezes, classificar erroneamente os experimentos. A acurácia das ferramentas, nessas mesmas 3.432 instâncias, variam entre $53,47\%$ e $63,26\%$ [Alex, 2017].

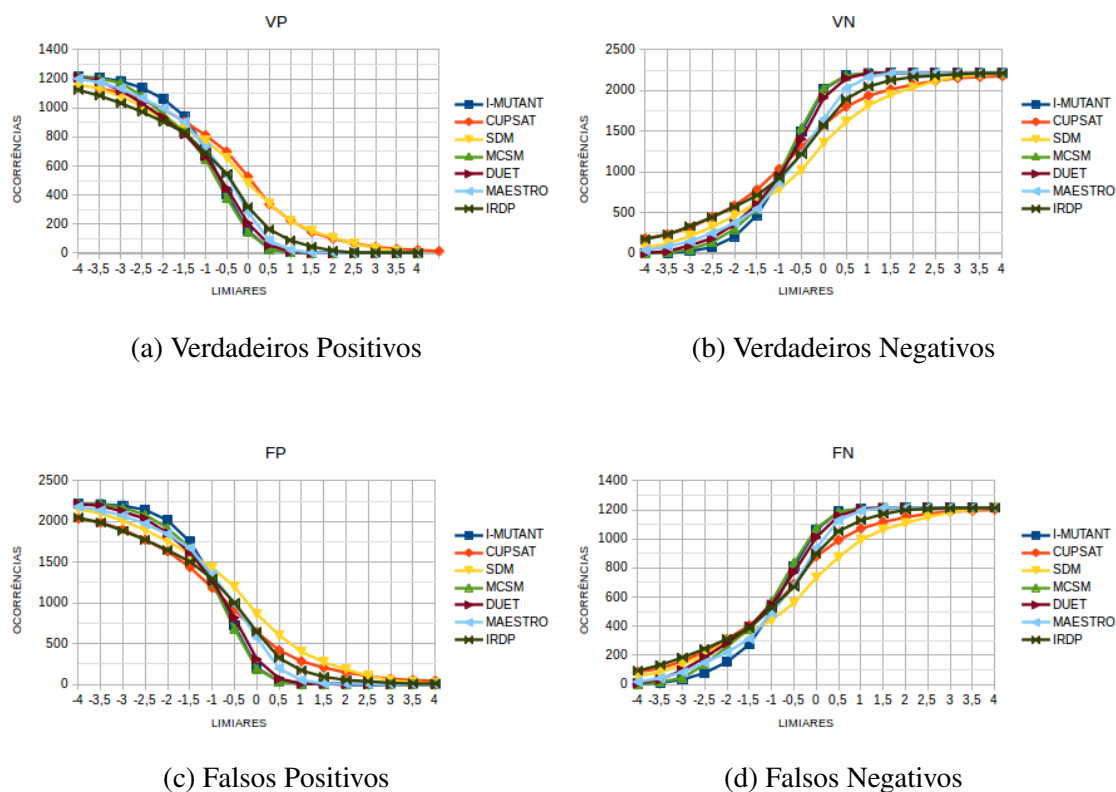
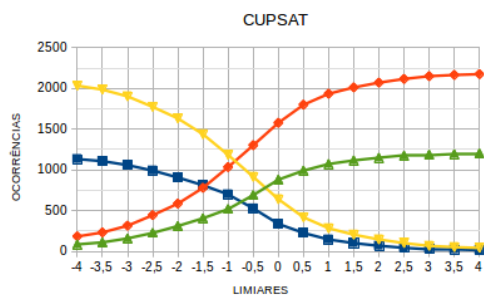
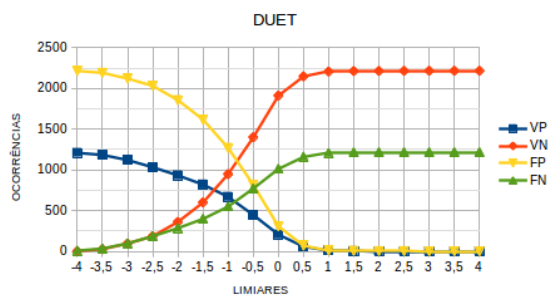


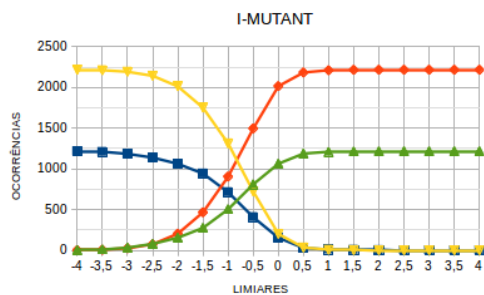
Figura 6: Comparação das matrizes de confusão das sete ferramentas agrupadas.



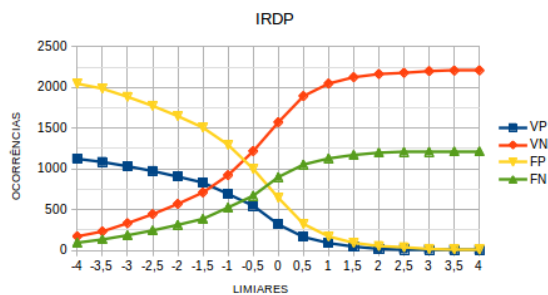
(a) CUPSAT



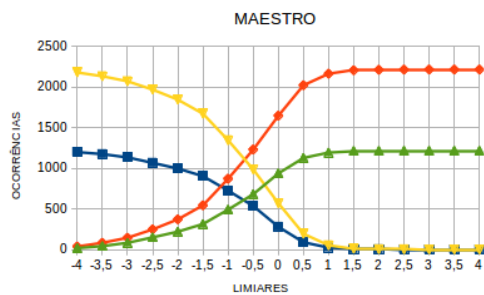
(b) DUET



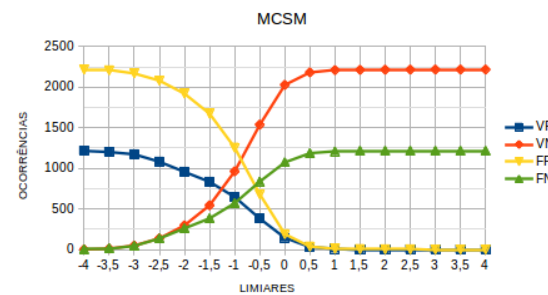
(c) I-MUTANT



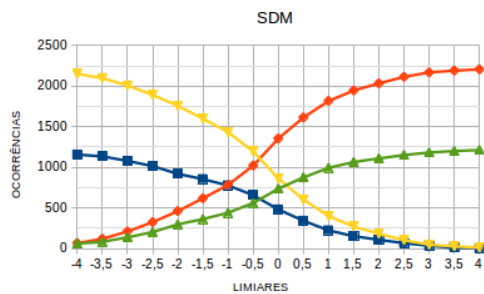
(d) iRDP



(e) MAESTRO



(f) MCSM



(g) SDM

Figura 7: Comparação das matrizes de confusão das sete ferramentas individualmente.

De acordo com esta análise, portanto, um limiar entre -1 e $-0,5$, teriam os resultados

mais balanceados, podendo alcançar totais de desestabilizantes e de estabilizantes mais semelhantes. No entanto, os gráficos também demonstram que criaria um maior número de verdadeiros negativos e falsos positivos.

Outra abordagem, que pode vir a ser utilizada, é a classificação com classes ternárias, ou seja, utilizar dois limiares para separar mutações de estabilizantes e desestabilizantes e incluir uma terceira classe de inconclusivo. Essa abordagem é sugerida pelo autor do artigo em [Olney, 2018], que realiza um estudo sobre os limiares e quais seriam esses valores ótimos, que no caso: desestabilizante $\leq -0,6 \text{ kcal mol}^{-1}$ < inconclusivo $\leq 0,3 \text{ kcal mol}^{-1}$ < estabilizante.

7 CONCLUSÃO

Para que se possa avaliar métodos de Aprendizado de Máquina, é necessário que exista uma grande diversidade de dados. Esses dados devem representar, balanceadamente, as classes que serão avaliadas pelos modelos de predição gerados. No entanto, com a grande diferença de mutações pontuais desestabilizantes, ao se comparar com estabilizantes, torna esta tarefa un tanto difícil.

Este trabalho teve o objetivo de propor formas de contornar esta situação, ao utilizar-se de métodos, que possam melhorar os resultados de predições, ao criar modelos que reaproveitassem predições de outras ferramentas. Ao avaliar o resultado dos experimentos propostos, é possível concluir que:

- No experimento 1, demonstra que ter bases de treinamento balanceados tendem a melhorar os resultados dos modelos de predição nas métricas de acurácia e de precisão;
- Ter muitos atributos alvo de uma só classe, como no experimento 2, tendem a aumentar a diferença entre as métricas de precisão e revocação, uma vez que terão muito mais características definidas de uma classe, do que a outra, na predição do atributo alvo.
- Os algoritmos de *ensemble* de classificadores obtiveram, em geral, melhores resultados que os algoritmos de classificação, assim como quando comparado com as ferramentas de predição;
- Mesmo utilização de diversos métodos, a limitação de se ter muito menos dados de mutações estabilizantes, dificultou a geração de modelos confiáveis, perceptível nos valores de medida-F não muito próximos de 1.

Essa falta de confiabilidade acaba indo contra a importância de se ter predições corretas, sempre que possível, para poder identificar mutações malélicas [Fersht, 1993] [Y. Sugita, 1998], ou benéficas, que podem ser utilizadas na criação de remédios para patogêneses. Isso reforça o fato de que é necessário encontrar novas formas capazes de sobrepor a

insuficiência de confiabilidade na utilização de algumas ferramentas na predição de mutações pontuais na estrutura de proteínas.

Embora os algoritmos de *ensemble* de classificadores tenham obtido resultados mais confiáveis que as ferramentas individuais, ainda assim estão abertos a melhorias, que poderão ser realizadas em trabalhos futuros.

Uma das propostas de trabalhos futuros, envolve a combinação de ferramentas utilizando algoritmos genéticos, para determinar o peso de cada valor $\Delta\Delta G$ predito pelas ferramentas. O artigo "*CompScore: boosting structure-based virtual screening performance by incorporating docking scoring functions components into consensus scoring*" [Perez-Castillo et al., 2019] propõe um aumento de desempenho de 45% ao aplicar algoritmos genéticos sobre resultados de diversas ferramentas e encontrar a combinação de pontuação, ou pesos, que maximizassem o desempenho de cada uma das ferramentas quando aplicado o consenso de resultados.

Existe a possibilidade, também, de adicionar novas ferramentas para serem aplicadas no *ensemble* de classificadores, uma vez que quanto mais características tivermos para avaliar, mais robustas serão as predições de cada mutação, proporcionando mais chances de o modelo de predição identificar a classe, ou $\Delta\Delta G$, mais corretos.

REFERÊNCIAS

- D. C. Alex. EN-MUTATE: predição do impacto de mutações pontuais em proteínas utilizando ensemble learning, 2017.
- J. Auclair, M. P. Busine, C. Navarro, E. Ruano, G. Montmain, F. Desseigne, J. C. Saurin, C. Lasset, V. Bonadona, S. Giraud, and et al. Systematic mrna analysis for the effect ofmlh1 andmsh2 missense and silent mutations on aberrant splicing. *Human Mutation*, 27(2):145–154, 2006. doi: 10.1002/humu.20280.
- H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalova, and P. Bourne. The protein data bank nucleic acids research, 2017. URL www.rcsb.org.
- E. Capriotti, P. Fariselli, and R. Casadio. A neural-network-based method for predicting protein stability changes upon single point mutations. *Bioinformatics*, 20(Suppl 1): i63–i68, 2004. doi: 10.1093/bioinformatics/bth928.
- E. Capriotti, P. Fariselli, I. Rossi, and R. Casadio. A three-state prediction of single point mutations on protein stability changes. *BMC Bioinformatics*, 9(Suppl 2), 2008. doi: 10.1186/1471-2105-9-s2-s6.
- T. L. B. Catherine L. Worth, Robert Preissner. Sdm — a server for predicting effects of mutations on protein stability and malfunction. 39:W215–W222, 2011.
- C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- J. Cheng, A. Randall, and P. Baldi. Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins: Structure, Function, and Bioinformatics*, 62(4):1125–1132, 2005. doi: 10.1002/prot.20810.
- K.-C. Chou. Structural bioinformatics and its impact to biomedical science. 2004.
- Y. Dehouck, J. M. Kwasigroch, D. Gilis, and M. Rومان. Popmusic 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinformatics*, 12(1), 2011. doi: 10.1186/1471-2105-12-151.

T. G. DIETTERICH. Ensemble methods in machine learning. *Multiple classifiers systems*, page 1–15, 2000.

N. Dolzhanskaya, M. A. Gonzalez, F. Sperziani, S. Stefl, J. Messing, G. Y. Wen, E. Alexov, S. Zuchner, and M. Velinov. A novel p.leu(381)phe mutation in presenilin 1 is associated with very early onset and unusually fast progressing dementia as well as lysosomal inclusions typically seen in kufs disease. *Journal of Alzheimers Disease*, 39(1):23–27, Jul 2014. doi: 10.3233/jad-131340.

T. L. B. Douglas E.V. Pires, David B. Ascher. Duet: a server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic Acids Research*, 42:W314–W319, 2014.

K. FACELI, A. C. LORENA, J. GAMA, and A. CARVALHO. Inteligência artificial – uma abordagem de aprendizado de máquina., 2011.

A. R. Fersht. Protein folding and stability: the pathway of folding of barnase. *FEBS Letters Volume 325, Issues 1–2, 28 June 1993*, pages 5–16, 1993.

M. GALAR, A. FERNANDEZ, E. BARRENECHEA, H. BUSTINCE, and F. HERRERA. A review on ensembles for the class imbalance problem: bagging, boosting, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C(Applications and Reviews) v.42 n.4*, pages 463–484, 2012.

M. Giollo, A. J. Martin, I. Walsh, C. Ferrari, and S. C. Tosatto. Neemo: a method using residue interaction networks to improve prediction of protein stability upon mutation. *BMC Genomics*, 15(Suppl 4), 2014. doi: 10.1186/1471-2164-15-s4-s7.

R. Guerois, J. E. Nielsen, and L. Serrano. Predicting changes in the stability of proteins and protein complexes: A study of more than 1000 mutations. *Journal of Molecular Biology*, 320(2):369–387, 2002. doi: 10.1016/s0022-2836(02)00442-4.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, Aug 1998. ISSN 0162-8828. doi: 10.1109/34.709601.

T. Isokawa, H. Nishimura, and N. Matsui. Quaternionic multilayer perceptron with local analyticity. *Information*, 3:546–831, 2012.

H. U. K. K. K. Abdulla Bava, M. Michael Gromiha1 and A. Sarai. Protherm, version 4.0: thermodynamic database for proteins and mutants. *Nucleic Acids Research*, 32, 2004.

R. Kohavi and F. Provost. *Glossary of Terms: Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, 1998. URL <http://robotics.stanford.edu/~ronnyk/glossary.html>.

- T. Kucukkal, Y. Yang, S. Chapman, W. Cao, and E. Alexov. Computational and experimental approaches to reveal the effects of single nucleotide polymorphisms with respect to disease diagnostics. *International Journal of Molecular Sciences*, 15(6):9670–9717, 2014. doi: 10.3390/ijms15069670.
- T. G. Kucukkal, M. Petukh, L. Li, and E. Alexov. Structural and physico-chemical effects of disease and non-disease nssnps on proteins. *Current Opinion in Structural Biology*, 32: 18–24, 2015. doi: 10.1016/j.sbi.2015.01.003.
- J. Laimer, H. Hofer, M. Fritz, S. Wegenkittl, and P. Lackner. Maestro - multi agent stability prediction upon point mutations. *BMC Bioinformatics*, 16(1), 2015. doi: 10.1186/s12859-015-0548-6.
- H.-D. Li, Q.-S. Xu, and Y. Liang. Support vector machines for classification and regression. *Support Vector Machines and Their Application in Chemistry and Biotechnology*, page 15–48, Mar 2011. doi: 10.1201/b10911-3.
- N. Luscombe, D. Greenbaum, and M. Gerstein. What is bioinformatics? an introduction and overview. *Yearbook of Medical Informatics*, 2001.
- T. J. Magliery. Protein stability: computation, sequence statistics, and new experimental methods. *Current opinion in structural biology*, 33:161–168, 2015.
- M. Masso and I. I. Vaisman. Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis. *Bioinformatics*, 24(18):2002–2009, 2008. doi: 10.1093/bioinformatics/btn353.
- L. Montanucci, E. Capriotti, Y. Frank, N. Ben-Tal, and P. Fariselli. Ddgun: an untrained method for the prediction of protein stability changes upon single and multiple point variations. *BMC Bioinformatics*, 20(S14), 2019. doi: 10.1186/s12859-019-2923-1.
- R. Olney. A systematic exploration of ddg cutoff ranges in machine learning models for protein mutation stability prediction. *Journal of Bioinformatics and Computational Biology*, 16, 2018.
- V. Parthiban, M. M. Gromiha, and D. Schomburg. Cupsat: prediction of protein stability upon point mutations. *Nucleic Acids Research*, 34(Web Server), Jan 2006. doi: 10.1093/nar/gkl190.
- Y. Perez-Castillo, S. Sotomayor-Burneo, K. Jimenes-Vargas, M. Gonzalez-Rodriguez, M. Cruz-Montegudo, V. Armijos-Jaramillo, M. N. D. S. Cordeiro, F. Borges, A. Sánchez-Rodríguez, E. Tejera, and et al. Compscore: boosting structure-based virtual screening performance by incorporating docking scoring functions components into consensus scoring. 2019. doi: 10.1101/550590.

- B. Pierre and B. Soren. *Bioinformatics The Machine Learning Approach*. The MIT Press, 2002.
- D. E. V. Pires, D. B. Ascher, and T. L. Blundell. mcsM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics*, 30(3):335–342, 2013. doi: 10.1093/bioinformatics/btt691.
- N. Pochet, F. D. Smet, J. A. K. Suykens, and B. L. R. D. Moor. Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. *Bioinformatics*, 20(17):3185–3195, Jan 2004. doi: 10.1093/bioinformatics/bth383.
- S. Prabhakaran. How naive bayes algorithm works? (with example and full code): ML, Nov 2018. URL <https://www.machinelearningplus.com>.
- L. Quan, Q. Lv, and Y. Zhang. Strum: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics*, 32(19):2936–2946, 2016. doi: 10.1093/bioinformatics/btw361.
- C. Rye, R. Wise, O. V. Jurukovski, J. DeSaix, J. Choi, and Y. Avissar. *Biology*. OpenStax Biology, 2016.
- A. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44, 1959.
- F. Shaikh. Deep learning vs. machine learning – the essential differences you need to know, 2017. URL <https://www.analyticsvidhya.com>.
- S. Stefl, H. Nishi, M. Petukh, A. R. Panchenko, and E. Alexov. Molecular mechanisms of disease-causing missense mutations. *Journal of Molecular Biology*, 425(21):3919–3936, 2013. doi: 10.1016/j.jmb.2013.07.014.
- H. Verli. *Bioinformática da Biologia à Flexibilidade Molecular*. Sociedade Brasileira de Bioquímica e Biologia Molecular, 1 edition, 2014.
- D. Voet, J. G. Voet, and C. W. Pratt. *Fundamentos de Bioquímica-: A Vida em Nível Molecular*. Artmed Editora, 2014.
- G. Vytautas and L. de G. Bert. Alchemical free energy calculations for nucleotide mutations in protein-dna complexes, 2017.
- G. Wainreb, L. Wolf, H. Ashkenazy, Y. Dehouck, and N. Ben-Tal. Protein stability: a single recorded mutation aids in predicting the effects of other mutations in the same amino acid site. *Bioinformatics*, 27(23):3286–3292, 2011. doi: 10.1093/bioinformatics/btr576.

A. K. Y. Sugita. Dependence of protein stability on the structure of the denatured state: Free energy calculations of i56v mutation in human lysozyme. *Biophysical Journal, Volume 75, Issue 5, November 1998*, pages 2178–2187, 1998.

P. YANG, P. D. YOO, J. FERNANDO, B. B. ZHOU, Z. ZHANG, and A. Y. ZOMAYA. Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications. *IEEE Transactions on cybernetics v.44 n.3*, pages 445–455, 2014.

Y. Yang. *Temporal data mining via unsupervised ensemble learning*. Elsevier, 2017.

S. Yin, F. Ding, and N. V. Dokholyan. Eris: an automated estimator of protein stability. *Nature Methods*, 4(6):466–467, 2007. doi: 10.1038/nmeth0607-466.

Z. Zhang, M. A. Miteva, L. Wang, and E. Alexov. Analyzing effects of naturally occurring missense mutations. *Computational and Mathematical Methods in Medicine*, 2012:1–15, 2012. doi: 10.1155/2012/805827.

H. Zhou and Y. Zhou. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Science*, 11(11):2714–2726, 2009. doi: 10.1110/ps.0217002.

Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.