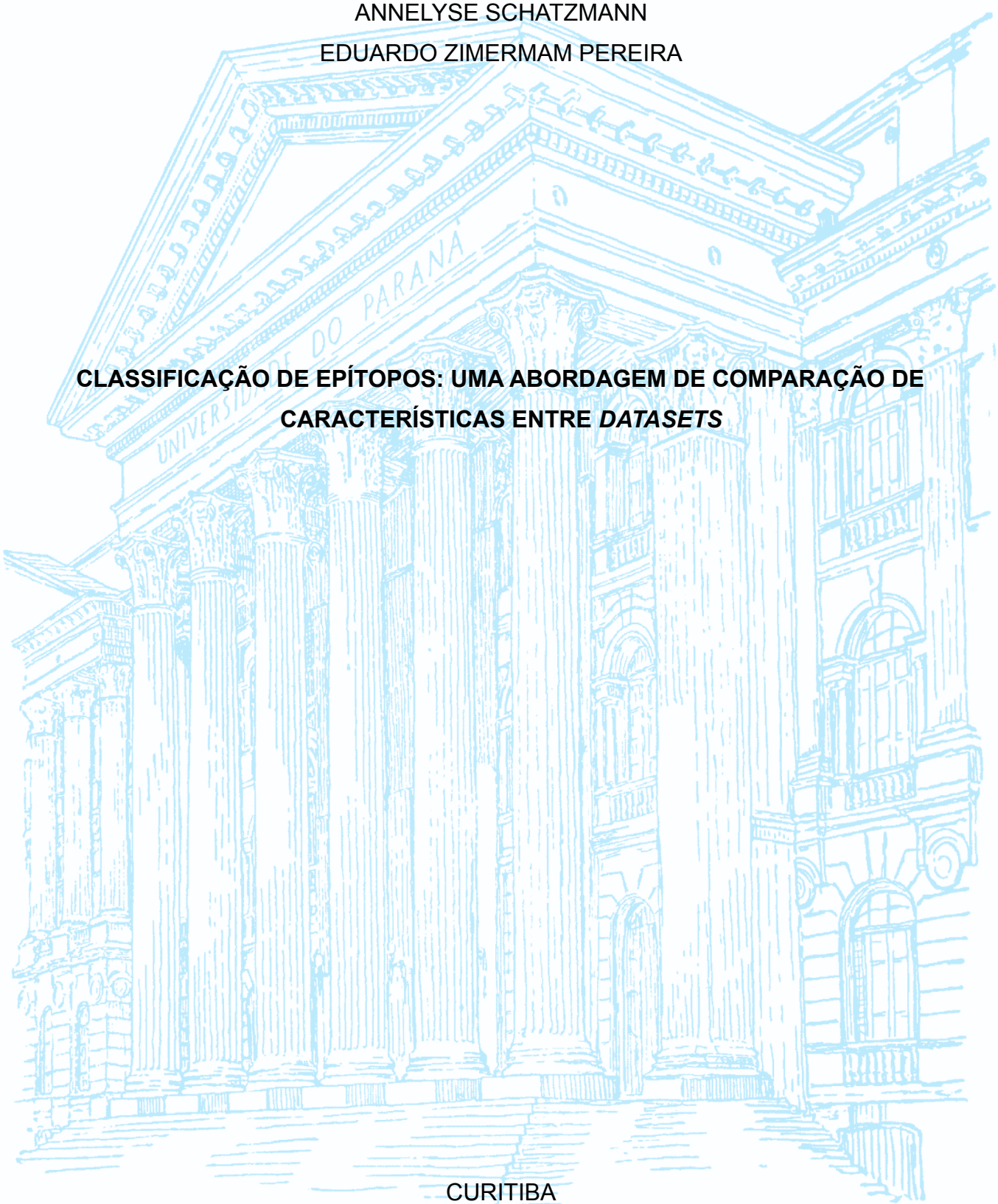
UNIVERSIDADE FEDERAL DO PARANÁ

ANNELYSE SCHATZMANN

EDUARDO ZIMERMAM PEREIRA



CLASSIFICAÇÃO DE EPÍTOPOS: UMA ABORDAGEM DE COMPARAÇÃO DE CARACTERÍSTICAS ENTRE *DATASETS*

CURITIBA

2021

ANNELYSE SCHATZMANN
EDUARDO ZIMERMAM PEREIRA

CLASSIFICAÇÃO DE EPÍTOPOS: UMA ABORDAGEM DE COMPARAÇÃO DE CARACTERÍSTICAS ENTRE *DATASETS*

Trabalho apresentado como requisito parcial à conclusão do curso de Bacharelado em Informática Biomédica, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Eduardo Jaques Spinosa.

CURITIBA

2021

AGRADECIMENTOS

Somos gratos primeiramente a Deus por nos permitir chegar até aqui.

Agradecemos aos nossos pais, Rogério Schatzmann, Gisele de Jesus Gralik e Rosemeri Zimermam que sempre nos apoiaram, incentivaram e abriram mão de nossas companhias em diversos momentos para alcançarmos o fim dessa jornada. Também agradecemos aos nossos irmãos mais novos, Anderson Zimermam Pereira, Carolina Zimermam Pereira e Helena Schatzmann por nos incentivarem e exigirem o nosso melhor. Agradecemos aos nossos amigos, por deixarem a nossa trajetória sempre mais leve com risadas, conversas e bons momentos.

Agradecemos aos mestres, estes que pavimentaram toda essa via que utilizamos para a conclusão desse projeto que marca uma vitória em nossas vidas e por toda dedicação e paciência durante tantos anos dentro e fora da sala de aula. Em especial, agradecemos o nosso orientador neste trabalho, Eduardo Jaques Spinosa, que desde o primeiro dia que nos encontramos dentro da sala de aula nos entende e compreende, demonstrando imenso carinho por nós.

Dedicamos esse trabalho aos nossos entes queridos Antônio Pereira (in memoriam) e Wilson Schatzmann (in memoriam), que infelizmente não podem presenciar tal momento, mas que com certeza estariam orgulhosos do nosso esforço e dedicação para alcançar algo que almejamos durante nossas vidas.

E acima de tudo agradecemos um ao outro, por todo amor, companheirismo, insistência, cobrança e prontidão frente às dificuldades que encontramos para chegar a este momento.

RESUMO

A classificação de epítopos ligantes por meio de métodos computacionais, como aprendizagem de máquina, tem grande importância para o desenvolvimento de vacinas, uma vez que o método auxilia a identificar epitopos que geram reação imunológica. Neste trabalho, tem-se como objetivo comparar o resultado do treinamento de um modelo SVM de predição de epítopos entre dois *datasets* distintos. O primeiro *dataset* contém as classes desbalanceadas e um tamanho variável das sequências de epítopos. Já no segundo, há classes balanceadas e um tamanho fixo para as sequências de epítopos. O modelo SVM tem como entrada um vetor de características decorrentes das sequências de epítopos de classe positivas e negativas. Dentre as 15 combinações de características iniciais apenas as quatro melhores foram escolhidas e testadas no segundo dataset para uma análise experimental, com o intuito de comparar o impacto que o *dataset* causa no treinamento do modelo. As taxas de acerto entre os *datasets* variam de 84% a 80% com uma área sob a curva ROC variando de 91% a 88%. Durante os testes, o que foi observado é que a combinação das características corretas e a escolha dos parâmetros corretos para o modelo SVM se mostram muito mais eficazes para alcançar bons resultados de predição do que as qualidades que o *dataset* impõe.

Palavras-chave: Epítopos, Classificação, Características, Máquina de Vetores de Suporte, Vacinas.

ABSTRACT

The classification of binders epitopes through computational methods, such as machine learning, has great importance for vaccine development since the method helps to identify epitopes that generate an immune response. In this work, the goal is to compare the training results of an SVM model for epitope prediction between two different datasets. The first dataset contains unbalanced classes and a variable size of the epitope sequences, while the second has balanced classes and a fixed size for the epitope sequences. The SVM model has as input a vector of characteristics calculated from the sequences of positive and negative class epitopes. Among the 15 combinations of initial features, only the four best ones were chosen and tested in the second dataset for an experimental analysis, in order to compare the impact that the dataset causes on the model's training. The accuracy between datasets range from 84% to 80% with an area under the ROC curve ranging from 91% to 88%. During testing, what was observed is that the combination of right characteristics and choosing the correct parameters for the SVM model proves to be much more effective in achieving good prediction results than the qualities that the dataset imposes.

Keywords: Epitopes, Classification, Features, Support Vector Machine, Vaccines

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

AAC	-	Composição de Aminoácidos
AAP	-	Escala de antigenicidade em pares
AAT	-	Escala de antigenicidade em trios
Ab	-	Anticorpo
Ag	-	Antígeno
AUC	-	Área sob a curva
BCEs	-	Epítopos de Células B
IEDB	-	*Immune Epitope Database*
MHC	-	Complexo Principal de Histocompatibilidade
NLP	-	*Natural Language Processing*
PyDPI	-	*Drug–protein interaction with Python*
RBF	-	*Radial Basis Function*
RNN	-	*Recurrent Neural Network*
SVC	-	*Support Vector Classification*
SVM	-	Máquina de Vetores de Suporte (*Support Vector Machine)*
TCEs	-	Epítopos de Células T

SUMÁRIO

1 INTRODUÇÃO

Ao longo de toda a história da humanidade, vários agentes causadores de doenças surgiram, entre os quais se destacam bactérias e vírus. No ano de 2019, o mundo viu surgir um novo agente causador de uma das maiores pandemias já registradas, o vírus Sars-CoV-2, causador da doença chamada COVID-19. Neste contexto, a cada surgimento de uma doença e de um novo agente causador, há a necessidade da sua contenção, controle e até mesmo, erradicação. As vacinas têm sido o meio mais efetivo para controle e erradicação de doenças em todo o mundo, e a sua eficácia está diretamente ligada ao resultado obtido no combate às mais variadas doenças (PARVIZPOUR et al., 2020).

O problema de se obter uma vacina com uma elevada eficácia é que por muitas vezes esse resultado demora, pois depende de inúmeros testes e experimentos para que se chegue a uma taxa de eficácia minimamente aceitável. Em um mundo globalizado, onde as pessoas circulam livremente de um país para outro, considerando uma doença infecciosa, como é o caso da COVID-19, o tempo de criação de uma vacina é essencial, surgindo assim a necessidade do emprego de novos métodos de testes e experimentação a fim de diminuir o tempo necessário para conclusão da vacina. Em meio a este problema, algoritmos de aprendizado de máquina revelam-se grandes aliados, com seu poder de processamento de dados, identificação de padrões, simulação de experimentos e outros tantos processos que levavam meses para serem feitos em um laboratório, atualmente são feitos em poucas horas ou dias com um simples computador e uma base de dados vasta o suficiente para que resultados expressivos sejam gerados.

Dentre os vários experimentos existentes, neste trabalho destacamos a classificação e predição de epítopos. Epítopos são partes de uma molécula de antígeno, os quais são reconhecidos pelos anticorpos do sistema imunológico, além disso são grandes aliados no combate a uma doença infecciosa e consequentemente, um objeto de muito estudo para o desenvolvimento de vacinas. A seleção do melhor epítopo ligante dentre os vários que um antígeno pode ter, aumenta a possibilidade de uma reação imunológica adequada, além de menor

tempo de testes e consequentemente menor custo envolvido no desenvolvimento da vacina.

Neste trabalho vamos explorar o trabalho proposto por Bahai et al. (2021) sobre predição de epítopos, com o objetivo de avaliar os algoritmos testados, as características extraídas e entender como um mesmo algoritmo comporta-se frente a diferentes *datasets* de epítopos virais. Vamos buscar entender qual é o impacto de cada característica extraída no modelo de predição e qual combinação de características apresenta o melhor resultado.

No capítulo 2, explanamos toda a fundamentação biológica e computacional para a compreensão do trabalho proposto. No capítulo 3 mencionamos todos os trabalhos que têm relação com o que foi estudado para a concepção deste projeto. Nos capítulos 4 e 5 apresentamos os materiais, métodos e resultados que obtivemos através dos nossos experimentos. Por fim, no capítulo 6 incluímos as nossas considerações finais sobre o trabalho e possíveis experimentos que podem ser realizados no futuro.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Esse trabalho tem como objetivo comparar os resultados de um modelo de predição binária de epítopos. Tal modelo foi treinado com dois *datasets* de múltiplos vírus, os quais têm qualidades diferentes. Um deles tem classes desbalanceadas e tamanho das sequências de aminoácidos variável, já o outro *dataset*, com as classes melhor balanceadas e com o tamanho das sequências de aminoácido fixo.

Além disso, pretende-se, realizar melhorias nos parâmetros com o propósito de otimizar tal processo de predição com extração de características que tem relação com o tamanho das sequências. Por fim, mostrar como que o treinamento de um mesmo modelo de aprendizado de máquina pode sofrer alterações em seus resultados dependendo do *dataset* escolhido para o processo de treinamento e também, como o avanço de modelos de predição de epítopos podem estar impactando no desenvolvimento de novas vacinas.

1.1.2 Objetivos específicos

- Adquirir um modelo de predição de epitopos de célula B de vírus treinado com diversos peptídeos de diversos vírus.
- Adquirir um *dataset* de epítopos e não epítopos com um tamanho variável de aminoácidos nas sequências e as classes desbalanceadas.
- Adquirir um segundo *dataset* de epítopos e não epítopos com tamanho fixo das sequências de aminoácidos e as classes balanceadas.
- Variar os parâmetros do modelo a fim buscar melhores resultados para o primeiro *dataset*.
- Avaliar os resultados obtidos nos treinamentos com ambos os *datasets.*
- Comparar os resultados obtidos para avaliar qual é o impacto dos datasets utilizados nos treinamentos do modelo.

1.2 JUSTIFICATIVA

Ao longo de toda a pandemia que a sociedade moderna tem vivido, é perceptível a necessidade de empregar técnicas que aumentem a velocidade e a qualidade das vacinas desenvolvidas. O que um dia demorou anos para ser desenvolvida, testada e homologada para enfim ser disponibilizada à população, hoje passa pelo mesmo processo em meses, e muito dessa velocidade se dá ao fato de hoje termos a possibilidade de processar muitos dados e testar diferentes conclusões em ambientes virtualizados e ao amplo uso de técnicas de aprendizado de máquina. O reconhecimento de sequências de epitopos tem grande importância no meio, pois a seleção de epítopos ligantes pode ser usada para gerar uma resposta imunológica no corpo humano, e com isso auxiliando o desenvolvimento de vacinas, ou seja, menor tempo de testes e consequentemente menor custo envolvido.

Para o desenvolvimento de um modelo que apresente resultados satisfatórios é muito importante a sua construção, principalmente no que diz respeito às características das classes que serão usadas para treinamento e posterior predição.

Por esse motivo, nós testamos combinações de características com diferentes qualidades, e também, em *datasets* diferentes a fim de entender qual é o impacto que características distintas têm sobre o treinamento de um modelo totalmente voltado aos *datasets* virais.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 FUNDAMENTAÇÃO BIOLÓGICA

2.1.1 Antígeno

Um antígeno (Ag) é uma molécula constituída em grande parte de proteínas e descendente de um patógeno, que por sua vez é um microorganismo ou corpo estranho ao organismo causador de doenças ou não, podendo ser, por exemplo um vírus ou uma bactéria.

O antígeno por sua vez é capaz de despertar uma resposta imune gerando proteínas específicas conhecidas como anticorpos (Ab) (ABBAS et al., 2015). O reconhecimento desses antígenos no corpo humano ocorre por receptores presentes na superfície de células B e células T (MUTNEJA, 2014).

2.1.2 Imunidade inata e adaptativa

A função do sistema imunológico é se defender contra microorganismos infecciosos ou de substâncias estranhas. Possuímos dois tipos de imunidade: a imunidade inata e a imunidade adaptativa.

O sistema imunológico inato é a primeira defesa do corpo contra um patógeno e é composto primeiramente por uma barreira física entre o indivíduo e o ambiente, sendo esta a pele ou também tecidos de epitélio que fazem parte do trato gastrointestinal e respiratório. Caso haja uma abertura nos tecidos epiteliais e o patógeno consiga chegar até a corrente sanguínea ocorre então uma barreira celular onde os macrófagos e neutrófilos, ou células dendríticas, começam a gerar uma resposta inflamatória combatendo e destruindo esse patógeno (ABBAS et al., 2015).

A imunidade inata gera uma resposta rápida no controle de infecções porém essa resposta não é específica ao patógeno em questão e o sistema reage sempre

da mesma maneira, porém ela está diretamente ligada à resposta adaptativa uma vez que esta é acionada perante uma infecção (ABBAS et al., 2015).

Partindo dessa afirmativa, o sistema imune adaptativo ou adquirido é ativado a partir da exposição das células dendríticas presentes em uma infecção. As células da imunidade adquirida conhecidas como linfócitos geram uma resposta imune específica devido ao reconhecimento de antígenos (ABBAS et al., 2013). Existem dois tipos de linfócitos envolvidos na resposta imune adaptativa: os linfócitos T (comumente conhecido como célula T) que reconhecem fragmentos de antígenos e os linfócitos B (comumente conhecido como célula B) que identificam o antígeno em sua forma real (SELA-CULANG et al., 2013). Assim, cada célula gera um tipo de resposta imunológica adaptativa, que serão descritas no próximo tópico. Além disso, para que o sistema adquirido seja eficaz em sua resposta imunológica há algumas propriedades a serem levadas em consideração, como mostra a Tabela 2.1.

Tabela 2.1: Propriedades importantes do sistema imune adquirido.

Característica	**Função**
Especificidade	Garante que antígenos distintos gerem respostas específicas.
Diversidade	Permite que o sistema imunológico responda a uma grande variedade de antígenos distintos.
Memória	Leva a uma melhor resposta, devido a exposições repetidas ao mesmo antígeno
Expansão Clonal	Eleva o número de linfócitos antígeno-específicos para acompanhar o aumento dos microrganismos
Especialização	Gera resposta ótima para a defesa contra diferentes microorganismos
Contração e homeostasia	Permite ao sistema imunológico responder aos antígenos
Não reatividade própria	Previne a injúria do hospedeiro durante a resposta a anticorpos estranhos

FONTE**:** ABBAS, 2013

2.1.3 Células B e Células T

Células B e Células T podem gerar uma resposta ao mesmo antígeno, porém essa resposta se dá por meios distintos.

A imunidade humoral diz respeito a uma resposta realizada por células B que produzem e inserem moléculas de anticorpos na circulação ou meios líquidos do corpo, como a mucosa. Os anticorpos gerados reconhecem a estrutura terciária do antígeno (MUTNEJA, 2014) e os eliminam por meio de receptores específicos em sua membrana. Cada anticorpo é específico a um antígeno, essa resposta deve ocorrer antes que o patógeno consiga adentrar uma célula do hospedeiro, sendo assim uma resposta aos antígenos extracelulares (ABBAS et al., 2013).

O complemento a imunidade celular é realizada por células T a antígenos intracelulares (uma vez que estão dentro de uma célula e não são acessíveis aos anticorpos) que foram ingeridos e fragmentados por macrófagos ou células dendríticas, também conhecidas como células apresentadoras de antígenos. Nesse caso, a célula T auxiliar reconhece o antígeno apresentado pela molécula MHC II (complexo principal de histocompatibilidade) presente nas célula apresentadora de antígeno e ativa macrógafos para realizarem a fagocitose (ingestão) e destruição do antígeno (MUTNEJA, 2014).

Por outro lado, além das células T auxiliares existem também as células T citotóxicas, que destroem as células infectadas por inteiro, juntamente com o microorganismo (ABBAS et al., 2013). A demonstração dos diferentes tipos de imunidade pode ser visualizada na Figura 2.3.

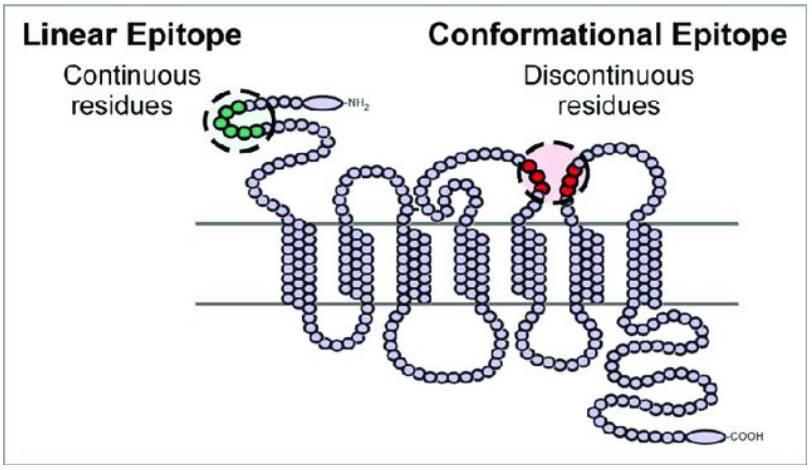
2.1.4 Epítopos

De acordo com Abbas (2015, p.100), um epítopo ou determinante antigênico, é caracterizado como “qualquer forma ou superfície disponível em uma molécula que possa ser reconhecida por um anticorpo”, ou seja, em uma macromolécula de um antígeno a região onde um anticorpo ou receptores MHC de célula T se ligam é chamada epítopo.

O reconhecimento de diferentes epítopos pode ser baseado na sequência simples e linear de aminoácidos, sendo esses os epítopos lineares, ou também,

identificando sua forma tridimensional decorrente do dobramento da proteína, conhecidos como epítopos conformacionais (MUTNEJA, 2014), como mostrado na Figura 2.2.

Figura 2.2**:** A figura demonstra o exemplo de um epítopo linear à direita, onde os aminoácidos estão em sequência. Já à esquerda o exemplo de um epítopo conformacional, considerando aminoácidos em duas partes distintas da sequência da proteína.



FONTE**:** DENG, 2017

Em um mesmo epítopo podemos ter a ligação de mais de um anticorpo ou receptor de célula T, sendo que estes geram respostas imunes específicas para o antígeno a que estão ligados (ABBAS et al., 2013).

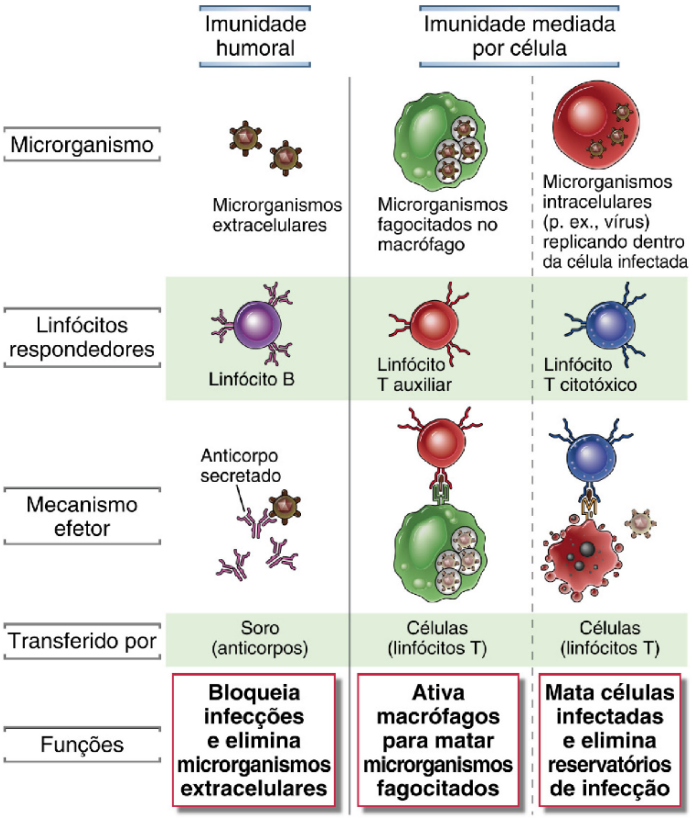
2.1.5 Vacinas

No entendimento de Zhang et al. (2015, p.2), "Uma vacina é uma preparação biológica que estimula a produção de anticorpos para induzir uma resposta imune a uma doença específica." Segundo esse mesmo autor, dentre os diferentes tipos de vacinas, a vacina baseada em epítopo recentemente tem atraído amplo interesse pois são projetadas para desencadear as respostas imunológicas das células B e T (ZHANG et al., 2015).

Assim, citado por Parvizpour et al. (2020, p.1034), "em 1985 Jacob et al. criou a primeira vacina baseada em epítopos contra a doença infecciosa causada por Vibrio cholerae e Escherichia coli." Desse modo, a ideia principal de uma vacina baseada em epítopos seria reconhecer os epítopos de células B (BCEs), sendo

aqueles epítopos reconhecidos apenas por células B, ou epítopos de células T (TCEs) sendo aqueles epítopos reconhecidos apenas por células T, e induzir uma resposta imunológica.

Figura 2.3: Demonstração da imunidade humoral gerada pela célula B, onde ocorre a geração de anticorpo e da imunidade mediada por células do tipo T, onde a célula T auxiliar depende de macrófagos e as células T citotóxicas eliminam as células infectadas por um antígeno.



FONTE: ABBAS, 2013

2.2 FUNDAMENTAÇÃO COMPUTACIONAL

2.2.1 Aprendizado de Máquina

Segundo Géron (2019, p.4), "Aprendizado de Máquina é a ciência (e a arte) da programação de computadores para que eles possam aprender com os dados". Ou seja, um aprendizado de máquina tem a habilidade de aprender

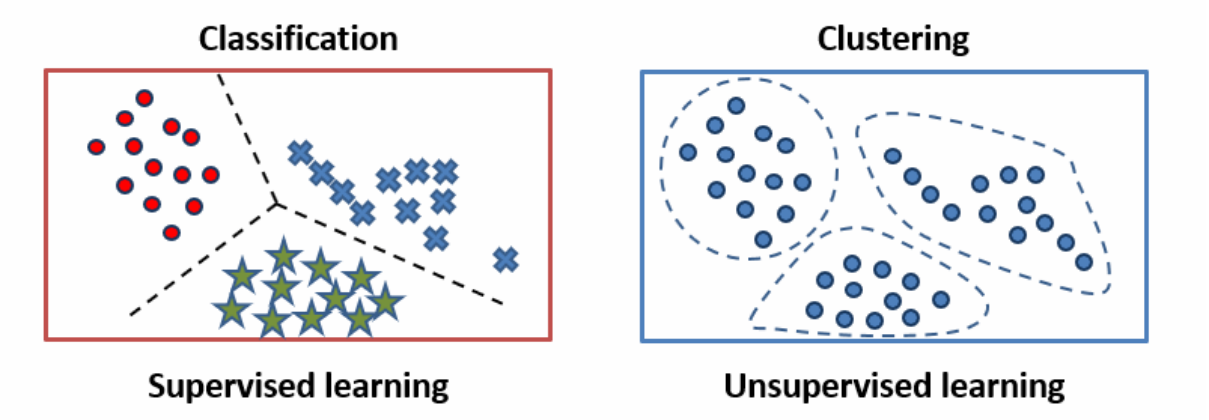
automaticamente um problema, identificar padrões e características e se adaptar à mudança dos dados de entrada.

Algoritmos de aprendizado de máquina são muito utilizados para resolver problemas que dependem de muito esforço manual, ou de grande quantidade de dados, ou ainda, problemas complexos onde padrões não são aparentes, mas que são importantes, ou até mesmo, obrigatórios para a solução do problema proposto.

Existem duas categorias de algoritmos mais comuns, a que utiliza aprendizado supervisionado e o aprendizado não supervisionado. O que as diferencia é o conjunto de dados de treinamento utilizado em ambas. O aprendizado supervisionado fornece um rótulo para cada amostra presente no conjunto de dados, como por exemplo, epítopo ou não epítopo. A classificação é a tarefa mais comum do aprendizado supervisionado, uma vez que o modelo é treinado com amostras, de duas classes ou mais e com isso aprende a classificar novas entradas com o rótulo adequado.

Em contrapartida, o aprendizado não supervisionado não contém rótulos no conjunto de treinamento. O *Clustering* é a tarefa mais comum do aprendizado não supervisionado, na qual o modelo realiza um agrupamento por grupos semelhantes (GÉRON, 2019) baseado em padrões e semelhanças entre as amostras fornecidas. Podemos perceber a diferença dos tipos de aprendizado na Figura 2.4.

Figura 2.4: Exemplos de categorias de aprendizado de máquina. À direita a demonstração de como o método de Classificação separa os dados já rotulados e à esquerda o método de Clustering e o agrupamento dos dados semelhantes entre si.



FONTE: TYPES, 2021

2.2.2 Máquina de Vetores de Suporte (SVM)

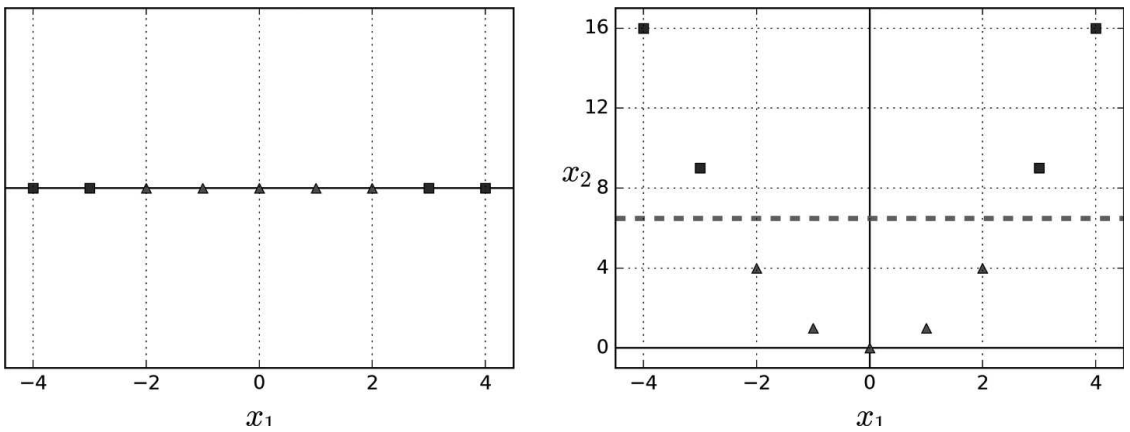
Máquina de Vetores de Suporte ou SVM, é um algoritmo de aprendizado de máquina supervisionado com a possibilidade de realizar classificação linear e não linear, regressão e detecção de *outliers.* É um modelo extremamente versátil para problemas de classificação complexos, porém com *datasets* de tamanho considerado pequeno (GÉRON, 2019). Para caracterizar o modelo SVM, precisamos falar sobre as margens de classificação. Segundo Géron (2019), a classificação de margem rígida apresenta dois problemas principais: o primeiro é que os dados precisam ser linearmente separáveis. E o segundo é que o modelo é extremamente sensível a *outliers.* Já na classificação de margem suave o modelo tolera erros, sendo mais flexível criando vetores de suporte definidos pelo parâmetro C do modelo. Os vetores de suporte são os pontos de cada classe mais próximos ao hiperplano de separação das classes. Um valor menor de C leva a um modelo mais flexível, porém com mais violações da margem definida entre o hiperplano e os vetores de suporte (GÉRON, 2019).

A classificação linear pode resolver inúmeros problemas, porém nem todos os conjuntos de dados podem ser separados linearmente. Em 1992, Boser et al. (1992) propôs uma solução criando uma maneira de separar dados não lineares. O conceito é baseado em elevar a dimensão dos dados, fazendo com que dados que estão em uma única dimensão e que não poderiam ser separados, quando elevados a uma dimensão superior, sejam capazes de ter um hiperplano que os separe. Elevar a dimensão dos dados pode ser feito adicionando uma característica polinomial aos dados. O exemplo da Figura 2.5 mostra como que o conceito funciona, no lado esquerdo da figura, é constatado que os dados são impossíveis de serem separados linearmente, já do lado direito, temos a inclusão de características polinomiais e o hiperplano entre as classes pode ser definido.

O SVM implementa tal possibilidade no que chamamos de *kernel.* Segundo Noble (2006), o *kernel* provê uma solução ao problema adicionando a dimensão necessária aos dados, tal operação é realizada aplicando uma função aos dados para que toda a amostra passe a ter dimensões mais altas, por exemplo, saindo de uma dimensão 2D, onde o hiperplano de divisão das classes é uma reta, para uma dimensão 3D onde, de fato, o hiperplano de divisão das classes torna-se um plano.

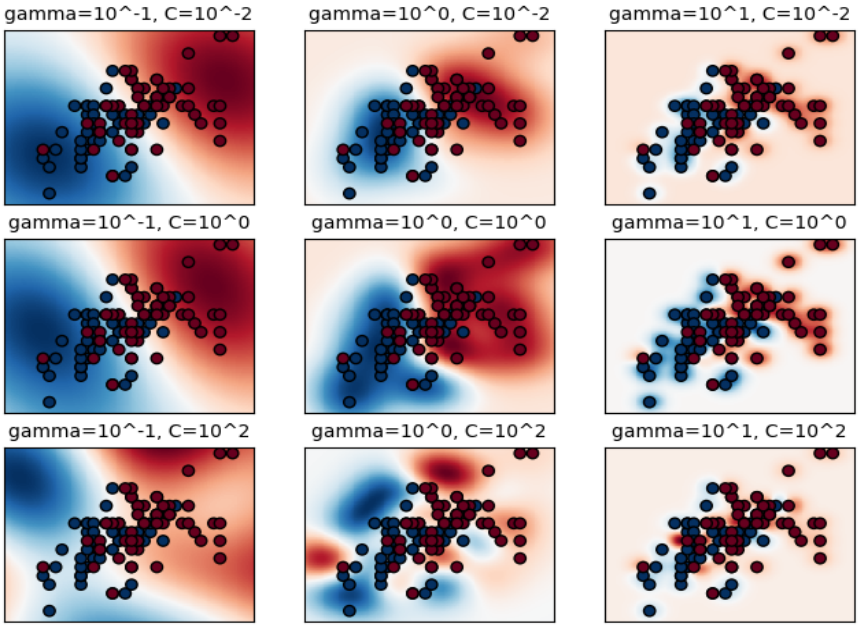
O *kernel* RBF é baseado em uma função de base radial e tem como entrada dois parâmetros: o *gamma* que define a influência de uma única amostra de treinamento, e o C que se comporta como um parâmetro de regularização do modelo. No *kernel* RBF é imprescindível definir bem os parâmetros de entrada, principalmente o parâmetro *gamma*, pois um *gamma* muito pequeno faz com que o modelo fique muito restrito e possivelmente não consiga capturar a complexidade dos dados. Já um valor muito grande faz com que o treinamento sofra um sobreajuste excessivo, independente do valor do parâmetro C para tentar corrigir este comportamento. Para altos valores em C, a margem da função de decisão é menor, sendo aceita se a função de decisão que melhor classifica as amostras de treinamento. Já para valores menores, a margem da função de decisão é maior, fazendo com que a função de decisão seja simplificada, o que resulta em uma perda de precisão. Na Figura 2.6 está demonstrado como os parâmetros *gamma* e C influenciam o desempenho do modelo (PEDREGOSA, 2011a).

Figura 2.5: Exemplo da adição de mais uma característica polinomial em um conjunto de dados não linear, à esquerda dados com apenas uma característica $x1$ não podem ser separados linearmente, porém à direita, ao adicionar mais uma característica $x2$ aos dados obtemos uma estrutura 2D resultando em uma separação linear.



FONTE: GÉRON, 2019

Figura 2.6: Da direita para a esquerda temos o crescimento do parâmetro *gamma* e da parte superior para a inferior o parâmetro C em crescimento. A imagem demonstra o impacto que os parâmetros exercem sobre o resultado do modelo, resultado que é representado pelas áreas azuis e vermelhas.



FONTE: PEDREGOSA, 2011a

3 TRABALHOS RELACIONADOS

3.1 Prediction of Continuous B-Cell Epitopes in an Antigen Using Recurrent Neural Network

Na concepção de Saha (2006, p.1), "Os métodos experimentais usados para caracterizar epítopos são demorados e exigem grandes recursos". Assim, a predição de epítopos de célula B tem grande importância no desenvolvimento de vacinas. O estudo deste mesmo autor está disponível em um servidor chamado ABCpred (http://crdd.osdd.net/raghava/abcpred/) e tem como objetivo prever, a partir de uma sequência de antígeno, epítopos lineares de células B contendo um total de 20 aminoácidos. Nesse trabalho, dois algoritmos foram testados, porém a rede RNN apresentou melhores resultados, tendo sido treinada e testada com um conjunto de 700 epitopos e 700 não epitopos. No entanto, Saha verificou que apenas um dos quesitos de desempenho passa dos 70% como podemos verificar na Tabela 3.1.

Tabela 3.1: Resultado da RNN utilizando um threshold de 0.5 para diferentes tamanhos e epitopos.

Window Size	Sensitivity (%)	Specificity (%)	PPV (%)	Accuracy (%)	MCC
10	58.71	64.14	61.78	61.43	0.2293
12	53.57	61.71	58.30	57.64	0.1534
14	52.43	65.29	60.12	58.86	0.1786
16	*67.14*	*64.71*	*65.61*	*65.93*	*0.3187*
18	58.70	65.0	62.06	61.86	0.2373
20	57.14	71.57	66.51	64.36	0.2871

FONTE: SAHA, 2006

3.2 Conformational B-cell epitopes classification using machine learning techniques

Salem (2013, p.2) afirma que "É bem conhecido e comprovado que aproximadamente 90% dos epítopos de células B são conformacionais", e com isso seu estudo baseia-se na criação de um modelo de classificação de epitopos e não epitopos conformacionais utilizando SVM, sendo o objetivo obter uma área sob a

curva (AUC) maior de 75%, o maior percentual atingido até o momento do seu trabalho. Desta maneira foi adquirido um *dataset* de estruturas 3D do complexo antígeno-anticorpo, com um total de 71 complexos. O autor divide seu estudo em dois passos: o primeiro consiste na criação de uma matriz de características (m x n), onde as linhas (m) representam a quantidade de resíduos encontrados nos complexos e suas colunas (n) representam as características dos mesmos. Por sua vez, o segundo passo é responsável pelo processo de classificação, onde do conjunto de 71 complexos, 61 são separados para o conjunto de treinamento e 10 para o conjunto de testes. Além disso o SVM é treinado usando uma técnica de validação cruzada k-fold com k = 10 e o *kernel* que obteve o melhor resultado foi o Radial Basis Function (RBF) = 2, chegando a 97% de acurácia, como podemos verificar na Tabela 3.2.

Tabela 3.2: Comparação dos resultados do SVM utilizando *kernel* RBF e Linear.

Kernel fn / Performance	Linear	RBF with sigma equals		
		1	2	3
Accuracy	0.506	0.9685	0.9776	0.9695
Sensitivity	0.506	0.937	0.9756	0.9654
Specificity	0.506	1	0.9798	0.9736

FONTE: SALEM, 2013

3.3 EpitopeVec: linear epitope prediction using deep protein sequence embeddings

Pelas observações de Bahai et al. (2021) os métodos tradicionais para determinar epitopos de células B são caros e demorados. Além disso esse mesmo autor cita que, “Em particular, a identificação de epítopos de células B (BCEs) é importante para aplicações, como projeto de vacina baseada em peptídeo”, abordando assim um estudo com a finalidade de prever BCEs lineares a partir de um modelo de machine learning. O autor seleciona oito conjuntos de dados de tamanhos diversos de epítopos originários de cinco métodos anteriores já criados e

propõe um novo *dataset* nomeado de Viral o qual utiliza um conjunto de dados de vírus do IEDB (*Immune Epitope Database*) com 4432 epítopos e 8460 não epítopos.

O primeiro passo do estudo é a criação de um vetor de extração de características, dentre essas o conjunto: composição de aminoácidos (AAC), escala de antigenicidade em par (AAP), escala de antigenicidade em trios (AAT), representação de sequência (Protvec), entre outras. Em seguida, o modelo utilizado é o SVM com *kernel* RBF. O treinamento ocorre com o conjunto de dados BCPred e este é induzido a uma validação cruzada de cinco vezes. Diversas combinações de características foram testadas, sendo que a que obteve o melhor resultado nas três métricas avaliadas foi a combinação AAC + AAP + AAT + Protvec, conforme a Tabela 3.3.

Tabela 3.3: Comparação da acurácia para cada combinação de características.

Feature	**ROC_AUC**	**Accuracy**	**F1**
AAC + DPC	0.709	66.2	0.642
AAP + AAT	0.871	80.34	0.793
Protvec + 4-mers	0.691	67.73	0.672
AAP + AAT + AAC + DPC	0.875	80.84	0.802
AAC + AAP + AAT + Protvec	0.889	81.31	0.811

FONTE: BAHAI, 2021

Para testar e avaliar o modelo Viral treinado com o *dataset* Viral, Bahai et al. (2021) realizou dois testes distintos. Em um primeiro teste, foi realizada a predição de um conjunto de 10 epítopos linear de SARS-CoV-1 compilados por Grifoni et al. (2020) do IEDB, tendo sido predito com sucesso 7 dos 10 epítopos lineares. Para um segundo teste, o autor compilou do IEDB um conjunto de 19 peptídeos. Dentre esses, 9 são epítopos de SARS-CoV-2, onde o preditor Viral predisse 7 dos 9 epítopos corretamente, e 10 não epítopos, que o preditor Viral indicou corretamente como não sendo epítopos.

De tal forma, Bahai observou que o preditor Viral tem alta precisão com as duas classes possíveis para o problema e também com *datasets* independentes.

4 MATERIAIS E MÉTODOS

Para o desenvolvimento deste projeto utilizamos o trabalho intitulado "*EpitopeVec: linear epitope prediction using deep protein sequence embeddings*" realizado por Bahai et al. (2021). Nosso objetivo foi aplicar a técnica descrita no trabalho mencionado com *dataset*s especificamente virais e realizar um compilado dos resultados obtidos dentre as possíveis combinações de características que apresentaram melhor desempenho no método testado.

4.1 MATERIAIS

4.1.1 *Datasets*

Contamos com a utilização de *datasets* binários, ou seja, *datasets* que contém apenas duas classes, uma que chamamos de positiva, ou seja, quando o peptídeo é um epítopo, e a que chamamos de negativa, naturalmente, quando o mesmo não é um epítopo.

Inicialmente, utilizamos o *dataset* Viral construído por Bahai et al. (2021), o qual é composto por 4431 sequências positivas e 8460 sequências negativas com tamanhos que variam de 6 a 46 aminoácidos. De acordo com o mesmo autor, para a construção do *dataset* foram utilizados três processos distintos. No primeiro processo foram capturados peptídeos reportados no IEDB como epítopos e não epítopos. No segundo processo, foi utilizada a técnica denominada CD-HIT para eliminação de peptídeos homólogos, sendo utilizado um *cut-off* de 80% para as sequências positivas e 90% para as sequências negativas. Por fim, um peptídeo pode ser considerado um epítopo como também um não epítopo em diferentes técnicas de neutralização. Para solucionar tal problema, o terceiro processo utilizado foi a remoção de todas as sequências iguais em ambas as classes.

No entretanto, a quantidade de sequências positivas é diferente do que está no projeto de BAHAI, pois tivemos que realizar um processo de curadoria do

dataset, removendo as sequências que, em sua composição existia uma letra “X”. Essa representa uma posição que pode ser assumida por dois aminoácidos diferentes. A extração das características (que depende da biblioteca *PyProt*), não suporta tal letra, invalidando a sequência e não realizando a valoração da característica, sendo que esse comportamento é esperado tendo em vista que a presença de um aminoácido ou outro pode significar que essa sequência não é um epítopo.

O segundo *dataset*, chamado de DeepVacPred, utilizado neste trabalho foi construído por Yang et al. (2021) em seu trabalho intitulado “*An in silico deep learning approach to multi-epitope vaccine design: a SARS-CoV-2 case study*”, sendo que este *dataset* foi escolhido por apresentar algumas particularidades que achamos interessantes de serem testadas frente ao *dataset* que foi escolhido inicialmente. Em um primeiro momento, este *dataset* contém 4925 peptídeos em ambas as classes, sendo assim um *dataset* balanceado. Além disso, todos os peptídeos presentes no *dataset* contém exatamente o mesmo tamanho de 10 aminoácidos, fazendo com que as características extraídas não fiquem desbalanceadas, principalmente aquelas que dependem do tamanho da sequência diretamente. O *dataset* todo foi extraído do IEDB, sendo que na classe positiva, foram coletados os últimos 4925 epítopos conhecidos até a data de extração dos dados realizada pelo autor.

Da mesma maneira que com o *dataset* de Bahai et al. (2021), no *dataset* de Yang et al. (2021) tivemos que realizar o processo de curadoria, sendo necessário, neste caso, remover 2 sequências da classe negativa e 1 sequência da classe positiva que continham as letras “X” e “Z” identificando um aminoácido não identificado. Deste modo, o total de sequências no *dataset* DeepVacPred são de 4922 peptídeos na classe negativa e 4923 peptídeos da classe positiva.

Os detalhes de cada *datasets* utilizados, bem como seus autores, quantidades de peptídeos no total e tamanho dos peptídeos presentes estão na Tabela 4.1.

Tabela 4.1: Detalhes sobre os *datasets* utilizados no trabalho.

Dataset	Autor	Tamanho da Sequência	Classe Negativa	Classe Positiva	Tamanho Dataset
Viral	BAHAI et al. (2021)	Variado (6 a 46 aminoácidos)	8.460	4.431	12.891
DeepVacPred	YANG et al. (2021)	10 aminoácidos	4.923	4.922	9.845

FONTE: Os autores

4.1.2 Algoritmos e bibliotecas

Para todos os treinamentos, validações e testes realizamos a implementação do código, salvo pequenas bibliotecas e trechos de código que utilizamos do código disponibilizado no estudo de Bahai et al. (2021). Iremos abordar as principais ferramentas e bibliotecas, pois a quantidade de recursos utilizada é extensa, com isso estamos disponibilizando em um repositório[1] o código fonte de toda a aplicação para que seja consultada em caso de necessidade.

A linguagem utilizada na implementação do projeto é a linguagem Python em sua versão 3.8.10, escolhida por dar suporte em grande parte das bibliotecas e *frameworks* de manipulação de dados, principalmente dados biológicos. Além do que essa é a linguagem em que os autores deste projeto têm mais conhecimento e facilidade de uso, sendo uma escolha natural para realização de tal trabalho.

A principal biblioteca que utilizamos é a *Scikit-learn* em sua versão 1.0.1, conforme colocado pela própria documentação da ferramenta por PEDREGOSA et al. (2011b)

> *Scikit-learn* é uma biblioteca de código aberto que tem suporte para aprendizados supervisionados e não supervisionados. Também oferece ferramentas para ajustes de modelos, pré-processamentos de dados, seleção e validação de modelos e muitos outros utilitários. (PEDREGOSA, 2011b)

Escolhemos tal recurso por ter ampla documentação, além de ser um requisito obrigatório para a estruturação desse projeto, pois demais bibliotecas que Bahai utilizou e nós também estamos utilizando necessitam desta ferramenta. Utilizamos a biblioteca *Scikit-learn* para a implementação do modelo SVM, utilizando

[1] O repositório pode ser acessado em: https://github.com/EduardoZimermam/epitope-prediction-svm

o método SVC, para a implementação da validação cruzada de tamanho 5 e também para a variação dos parâmetros de entrada do modelo utilizando a função *GridSearchCV.*

Além disso, *PyPro* é um módulo da biblioteca *PyDPI*, conforme CAO et al. (2013) declara em seu trabalho:

> *PyPro* é um conjunto de ferramentas para computar características estruturais e físico-químicas comumente usadas de proteínas e peptídeos a partir de sequências de aminoácidos, descritores moleculares de moléculas de drogas a partir de sua topologia e descritores de interação proteína-proteína e proteína-ligante. (CAO, 2013, p.3086)

Para a extração da característica AAC, demonstrada no próximo capítulo, utilizamos o método *GetAAComp*, este que é implementado pela biblioteca *PyPro*.

4.1.3 Características das sequências de proteínas

As características que serão apresentadas a seguir foram selecionadas com base na combinação de características que obteve os melhores resultados exibidos na última linha da Tabela 3.3 durante os testes realizados por Bahai et al. (2021).

a. Composição de aminoácidos (AAC)

A composição de aminoácido é representada por um vetor simples de 20 posições, onde cada posição representa um dos 20 aminoácidos conhecidos, sendo sua representação dada da seguinte maneira:

$$AAC(P) = (f_1, f_2, f_3, ..., f_{20}) \tag{1}$$

Onde:

$$f_i = \frac{R_i}{N} \tag{2}$$

Sendo que, i é a representação do aminoácido, Ri é a quantidade de aminoácidos do tipo i que aparecem na sequência, e N o tamanho da sequência que está sendo observada. Este método nos oferece uma visualização da frequência que um determinado aminoácido ocorre em um peptídeo, sendo importante para compreender os aminoácidos que normalmente estão associados a um epítopo.

b. Escala de antigenicidade em pares (AAP)

A escala de antigenicidade em pares foi proposta por Chen et al. (2007) e é relacionada à frequência que um par de aminoácidos ocorre em cada classe do *dataset* de treinamento. Essa característica é importante para determinarmos qual é a relação que encontramos entre os aminoácidos que estão na classe positiva quando comparados com a classe negativa.

Em termos práticos, a obtenção dessa característica é realizada em duas fases, cada qual com suas etapas distintas. Na primeira fase temos o objetivo de gerar um dicionário de escala de antigenicidade para todos os pares possíveis de aminoácidos, o que nos dá um total de 400 possibilidades (20 x 20) de pares. A obtenção do dicionário é dada pela equação abaixo:

$$Raap_i = log(\frac{f^+aap_i}{f^-aap_i})$$

(3)

Sendo que, i é o par de aminoácidos dentre os 400 possíveis, f^+aap_i é a frequência de ocorrência do par de aminoácidos na classe positiva do *dataset* e, finalmente, o f^-aap_i é a frequência do mesmo par de aminoácidos na classe negativa do *dataset*. Para evitar que se tenha um domínio de uma característica individual no treinamento do modelo, é realizada uma normalização dos valores entre -1 e +1 através da seguinte equação:

$$2 * \frac{Raap_i - min(Raap)}{max(Raap) - min(Raap)} - 1$$

(4)

Com isso, temos a conclusão da primeira fase do processo de cálculo da característica. Este dicionário é gerado apenas uma vez para o *dataset* completo, e caso o *dataset* tenha alguma modificação, sendo adicionado ou removido algum peptídeo, seu cálculo será necessário mais uma vez.

A segunda fase da extração da característica é o cálculo de uma média aritmética da escala de antigenicidade em pares de um peptídeo. Se faz necessária a utilização de uma média porque os peptídeos contêm quantidades diferentes de aminoácidos. Dessa maneira, para não termos quantidades diferentes de características, obtemos uma média simples para representar a característica para um peptídeo. O cálculo é realizado da seguinte maneira:

$$AAP = \frac{\sum_{1}^{n-1} Raap_i}{n-1} \tag{5}$$

Sendo que, N é o tamanho do peptídeo e i representa o par de aminoácidos. Ao final do cálculo da escala de antigenicidade em pares, teremos um valor que representa a relação dos aminoácidos em pares dentro de cada peptídeo.

c. Escala de antigenicidade em trios (AAT)

A escala de antigenicidade em trios foi proposta por Yao et al. (2012), sendo que o seu cálculo e extração é muito semelhante ao processo de cálculo de AAP. A diferença principal entre os dois métodos é que em AAT utilizamos trios de aminoácidos, elevando o espaço amostral de possibilidades de trios de aminoácidos, chegando a 8000 trios diferentes (20 x 20 x 20) no dicionário calculado. A equação para o cálculo de AAT é a seguinte:

$$Raat_i = log\frac{f^{+}aati}{f^{-}aati} \tag{6}$$

Sendo que, i é o trio de aminoácidos dentre os 8000 possíveis, $f^{+}aat$ é a frequência que um trio de aminoácidos ocorre na classe positiva do *dataset* e, finalmente, o $f^{-}aat$ é a frequência do mesmo trio de aminoácidos na classe negativa do *dataset*. Novamente, há a necessidade de normalização dos valores gerados, realizada através da seguinte equação:

$$2 * \frac{Raat_i - min(Raat)}{max(Raat) - min(Raat)} - 1 \tag{7}$$

Sendo que, $Raat_i$ é a razão da frequência de um trio de aminoácidos dentre os 8000 possíveis. Para cada trio de aminoácido que está no dicionário o seu valor será normalizado, evitando que uma característica se destaque mais que outra durante o treinamento do modelo.

Após a geração do dicionário de dados é necessário o cálculo para extrair a característica de cada peptídeo. A equação que realiza este processo é levemente diferente que a equação utilizada na extração da característica AAP, como podemos ver a seguir:

$$AAT = \frac{\sum_1^{n-2} Raat_i}{n-2} \tag{8}$$

A equação acima é realizada para cada peptídeo e ao final do seu cálculo teremos um valor que é a média aritmética simples entre a escala de antigenicidade de cada trio de aminoácidos do peptídeo.

d. Sequências de incorporação ProtVec

ProtVec é um projeto proposto por Asgari e Mofrad (2015) onde se utiliza um algoritmo de *Natural Language Processing (NLP).* O objetivo do projeto é descobrir similaridades entre os aminoácidos vizinhos em uma proteína baseando-se em uma estratégia k-mers.

Em NLP, comumente se prediz qual é a palavra que será utilizada no texto com base no contexto e nas palavras que foram utilizadas anteriormente em uma frase. No ProtVec a frase é definida como sendo a sequência de aminoácidos presentes em uma proteína, onde uma palavra da frase é considerada por uma janela sobreposta k-mers definida previamente e o objetivo não seria predizer quais são os aminoácidos subsequentes àquela determinada janela, mas sim, predizer a relação daquela palavra com todas as outras possibilidades de palavras existentes de mesmo tamanho com base em uma extração de 546.790 sequências de proteínas extraídas do database Swiss-Prot.

O cálculo de todas as possibilidades existentes juntamente com o valor de suas relações foi disponibilizado publicamente, por isso, nós realizamos o *download* do arquivo binário e o utilizamos no projeto para extrair a característica desejada.

A extração da característica é iniciada pela geração de todas as possibilidades de 4-mers dentro dos *datasets*. Esse valor para o tamanho da janela deslizante foi definido pelo dicionário utilizado. Após a primeira etapa, é buscada a relação de cada palavra gerada com todas as outras possibilidades de palavras geradas quando calculado o dicionário. O resultado dessa operação será para cada palavra um array de 300 posições, pois 300 possibilidades de palavras diferentes foram calculadas com o Swiss-Prot.

A etapa seguinte é calcular o produto escalar entre todos os vetores que representam todas as palavras de um mesmo peptídeo. Com esse cálculo, ao final, cada peptídeo do *dataset* de treinamento será representado por um vetor de 300 posições.

4.1.4 Modelo SVM

Para nos mantermos fidedignos ao trabalho base deste projeto e reproduzirmos os testes realizados nos *datasets* que escolhemos, utilizamos o modelo *Support Vector Classification (SVC)* que está disponível na biblioteca *Scikit-learn* e que utiliza a implementação de *Support Vector Machines (SVM)* da biblioteca LibSVM, assim como o *kernel Radial Basis Function (RBF)* que também foi utilizado nos testes realizados pelo trabalho base.

4.2 MÉTODOS

4.2.1 Combinação das características das sequências

Nos resultados apresentados na Seção Suplementar 2[2] (BAHAI et al., 2021) do trabalho base, verificou-se que o *dataset* BCPred é o utilizado para geração dos resultados demonstrados na Tabela 3.3, enquanto que o nosso objetivo no trabalho era verificar os resultados e o desempenho de cada característica com *datasets* virais. Para tal, realizamos diferentes combinações de características a fim de encontrar a melhor combinação, baseada na curva ROC. Na Tabela 4.2 podemos verificar todas as combinações que foram testadas para os resultados que obtivemos e que estão apresentados ao longo deste trabalho.

4.2.2 Treinamento e validação do modelo SVM

Para o treinamento e validação do modelo utilizamos um método disponibilizado na biblioteca *Scikit-learn* chamado *GridSearch.* O objetivo do *GridSearch* é criar uma grade de combinações entre os parâmetros de um modelo de inteligência artificial (PEDREGOSA, 2011c), em nosso caso o SVC. Logo, os valores que foram utilizados como parâmetros C e *gamma*, pois utilizamos o *kernel* RBF, podem ser visualizados na Tabela 4.3. O total de combinações dos parâmetros C e *gamma* foi de 77 candidatos que foram treinados e validados para todas as combinações de características já mencionadas, obtendo assim as métricas de validação e desempenho do modelo treinado com um *cross-validation* de tamanho 5.

[2] A seção suplementar 2 pode ser acessada no link: https://bityli.com/hOVRid

Tabela 4.2: Combinação de características e os valores dos melhores parâmetros obtidos do *GridSearch.*

Combinação de Características	C	Gamma
AAC	1	0,0001
AAT	1.000	0,0001
AAP	1	0,0001
ProtVec	250	0,0001
AAC + AAP	1	0,01
AAC + AAT	1.000	0,0001
AAP + AAT	250	0,001
AAC + ProtVec	0,001	0,0001
AAT + ProtVec	50	0,0001
AAP + ProtVec	250	0,0001
AAC + AAP + AAT	500	0,0001
AAC + AAP + ProtVec	100	0,0001
AAC + AAT + ProtVec	50	0,0001
AAP + AAT + ProtVec	100	0,0001
AAC + AAP + AAT + ProtVec	100	0,0001

FONTE: Os autores

Tabela 4.3: Todos os valores utilizados para cada parâmetro no *GridSearch.*

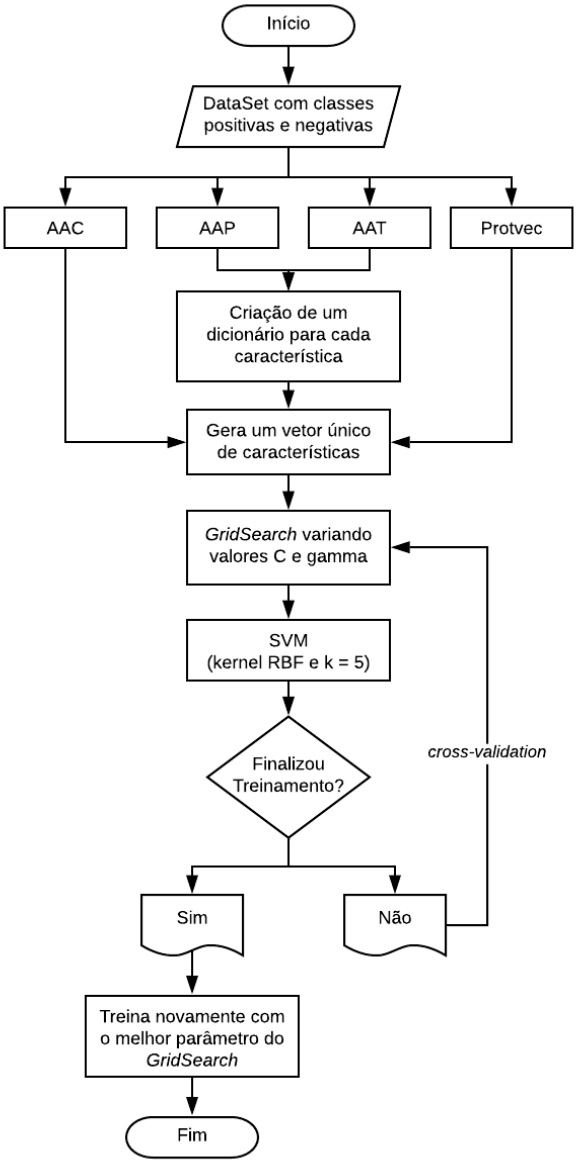
Parâmetros	
C	0,0001 0,001 0,01 0,1 1 25 50 100 250 500 1000
gamma	0,0001 0,001 0,01 0,1 1 10 100

FONTE: Os autores

Para apresentar os melhores resultados, configuramos o *GridSearch* para ranquear as melhores escolhas de parâmetros através da área sob a curva ROC para cada combinação, como mostra a Tabela 4.2. Após identificar o melhor parâmetro, o modelo é treinado novamente, com o objetivo de que o modelo treinado possa ser utilizado posteriormente, caso seja necessário. O fluxo dos eventos e tarefas para o treinamento e validação do modelo é demonstrado na Figura 4.2.

Levando em consideração os melhores parâmetros extraídos do *GridSearch* para cada combinação de características, obtivemos o valor de desempenho do modelo. Escolhemos as melhores combinações de características, demonstradas no capítulo de resultados deste mesmo documento, com os seus parâmetros no *dataset* Viral, e assim realizamos o treinamento das mesmas características com os mesmos parâmetros com o *dataset* DeepVacPred a fim de entender o comportamento dos treinamentos e os resultados obtidos quando temos um *dataset* com qualidades específicas.

Figura 4.1: Fluxo demonstrando os passos realizados no nosso modelo. Em resumo, após ser adquirido o dataset de epítopos é realizada a extração das características e suas combinações, o vetor de características resultantes é a entrada do modelo SVM que utiliza o método *GridSearch* com todos os valores possíveis para C e gamma. Ao final o Modelo é treinado novamente com os melhores parâmetros obtidos.



FONTE: Os autores

5 APRESENTAÇÃO DOS RESULTADOS

Neste projeto dividimos os experimentos em duas partes. A primeira parte tem como objetivo testar diferentes combinações das características que apresentaram melhores resultados no artigo de (BAHAI et al., 2021) para o dataset Viral presente no mesmo trabalho. Após obter as 4 combinações de características que apresentaram os melhores resultados baseados na acurácia e área sob a curva ROC, testamos os mesmos parâmetros e combinações de características para um novo dataset com qualidades diferentes, a fim de compreender se o resultado apresentado era consistente ou o dataset Viral de (BAHAI et al., 2021) apresentava características únicas que poderiam estar deixando o modelo sobreajustado.

Todos os resultados obtidos foram extraídos de nossa implementação própria do trabalho de (BAHAI et al., 2021), pois o trabalho disponibilizado pelo autor teve que ser adaptado para que a funcionalidade esperada fosse obtida.

5.1 Variação dos parâmetros C e *gamma*

Os primeiros testes realizados tiveram como objetivo variar os parâmetros de entrada do modelo com o objetivo de encontrar os melhores parâmetros para as características apresentadas. Todas as combinações de características tiveram como entrada os mesmos valores de parâmetros, como mostrado na Tabela 4.3, sendo que o método *GridSearch* variou as entradas do modelo através da estratégia de uma grade de combinações.

Apresentamos no Apêndice A deste trabalho os resultados das variações dos parâmetros para o modelo considerando todas as combinações de características que testamos. Ao todo foram 77 candidatos treinados e avaliados para cada combinação de características com a finalidade de extrair tais resultados. Por conta de tal valor ser alto para apresentar de uma maneira minimamente interessante, mostramos nas tabelas apenas os dez melhores resultados ranqueados pela área sob a curva ROC.

Em nossas análises sobre os valores alcançados, notamos que quando o modelo tinha apenas uma característica para o treinamento, os parâmetros de entrada não apresentavam uma forte influência sobre os resultados, o que mudou ao serem adicionadas uma ou mais características ao conjunto. Com o modelo sendo treinado com as quatro características, o que gerava um vetor de 322 posições, somente nos dez primeiros resultados tivemos uma variação de aproximadamente sete pontos percentuais na acurácia do modelo. Com tal observação feita, notamos que quanto mais alta a dimensão dos dados de entrada, maior é o impacto que os parâmetros têm sobre o modelo.

Os melhores resultados foram obtidos com um *gamma* pequeno que ficou entre 0.001 e 0.0001, mostrando que os dados do dataset não tem uma complexidade alta e são facilmente classificáveis. Já em contrapartida, o parâmetro C teve melhor performance com valores altos, indicando que a margem da função de decisão foi bem pequena. Os melhores resultados foram obtidos com um C entre 250 e 500. Com valores menores que estes o modelo apresentava leve piora em seu funcionamento, pois a margem de decisão estava ficando grande demais aceitando classificações errôneas e nos casos onde o C era maior que 500, o modelo também apresentou piora, mas neste caso foi porque a margem ficou muito pequena e com um *gamma* também muito pequeno o modelo fica muito restritivo.

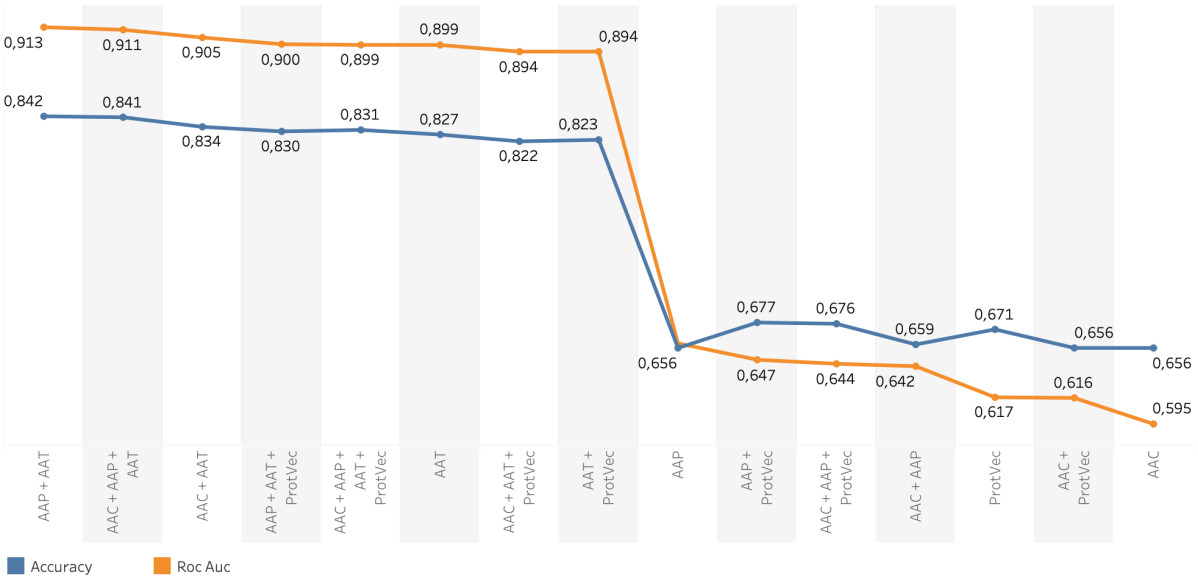
5.2 Combinação de características para o *dataset* Viral

Realizamos combinações entre todas as características de composição dos aminoácidos, antigenicidade em pares, antigenicidade em trios e ProtVec. As combinações foram realizadas inicialmente com uma característica, o que totalizou quatro combinações, uma para cada característica. Após os quatro primeiros testes, iniciaram as combinações duas a duas, tendo como resultado um total de seis combinações. Naturalmente, seguimos as combinações, agora realizando combinações de três a três, totalizando quatro combinações. Finalmente, executamos a combinação final com as características em conjunto totalizando uma combinação. Ao final de todo o processo testamos quinze combinações possíveis entre as características escolhidas. Os resultados obtidos para os melhores parâmetros, bem como todas as combinações testadas estão apresentadas no

Gráfico 5.1 em ordem decrescente de desempenho baseado na área sob a curva ROC.

O resultado apresentado pela característica AAT nos surpreendeu, pois seu desempenho foi muito melhor que algumas combinações de até três características, como é o caso da combinação AAC + AAT + ProtVec. O que não deixamos de notar é que essa composição tem a mesma característica que apresentou melhor resultado. Entendemos que isso aconteceu porque tanto AAC e ProtVec estavam fazendo com que o modelo perdesse desempenho frente ao modelo com AAT apenas. Analisando o motivo para que tal comportamento ocorresse, entendemos que AAT é extremamente descritiva sobre as sequências, pois o objetivo da antigenicidade é descrever a relação entre os aminoácidos e tal relação é muito importante para determinar um epítopo, sendo que este é um sítio de ligação. Assim, percebemos que os epítopos tem grandes semelhanças entre si, principalmente em sua composição de aminoácidos.

Gráfico 5.1: *Ranking* das melhores combinações de características



FONTE: Os autores

Como era esperado inicialmente, uma vez que demonstrado também nos resultados de Bahai et al. (2021), o modelo que apresentou o pior desempenho era constituído apenas de uma característica, a AAC, com uma área sob a curva ROC de 59.5% e uma acurácia de 65.6%. Isso ocorre pois AAC não descreve bem as

sequências, já que, o que caracteriza fortemente um epítopo é a organização dos aminoácidos e suas relações.

O melhor resultado foi apresentado pela combinação AAP + AAT com uma área sob a curva ROC de 91.4% e uma acurácia de 84.2%. Tal combinação é extremamente interessante, pois são duas características que observam a relação dos aminoácidos presentes no peptídeo, e são simples de serem extraídas, o que reforça a importância dessas duas características para os modelos treinados.

Dentre todas as características que testamos, AAT foi a que apresentou melhor desempenho, sendo a única característica presente em todas combinações que apresentaram desempenho superior a 80% nas duas métricas avaliadas. Como destaque negativo, esperávamos que a característica ProtVec tivesse um desempenho melhor, pois em sua construção é utilizado técnicas avançadas de NLP e por ser uma característica que também está relacionada a composição e organização dos aminoácidos dentro do peptídeo, assim como AAT. O que entendemos como uma diferença fundamental, é que a AAT trata o *dataset* utilizado para treinamento como ponto de partida para geração do dicionário de antigenicidade. Isso quer dizer que AAT faz uso de como realmente os peptídeos presentes estão organizados no *dataset*. Já quando se trata de ProtVec, o *dataset* para treinamento da NLP é diferente do utilizado para o treinamento do nosso modelo, pois considera um dataset de proteínas gerais, não contendo apenas epítopos.

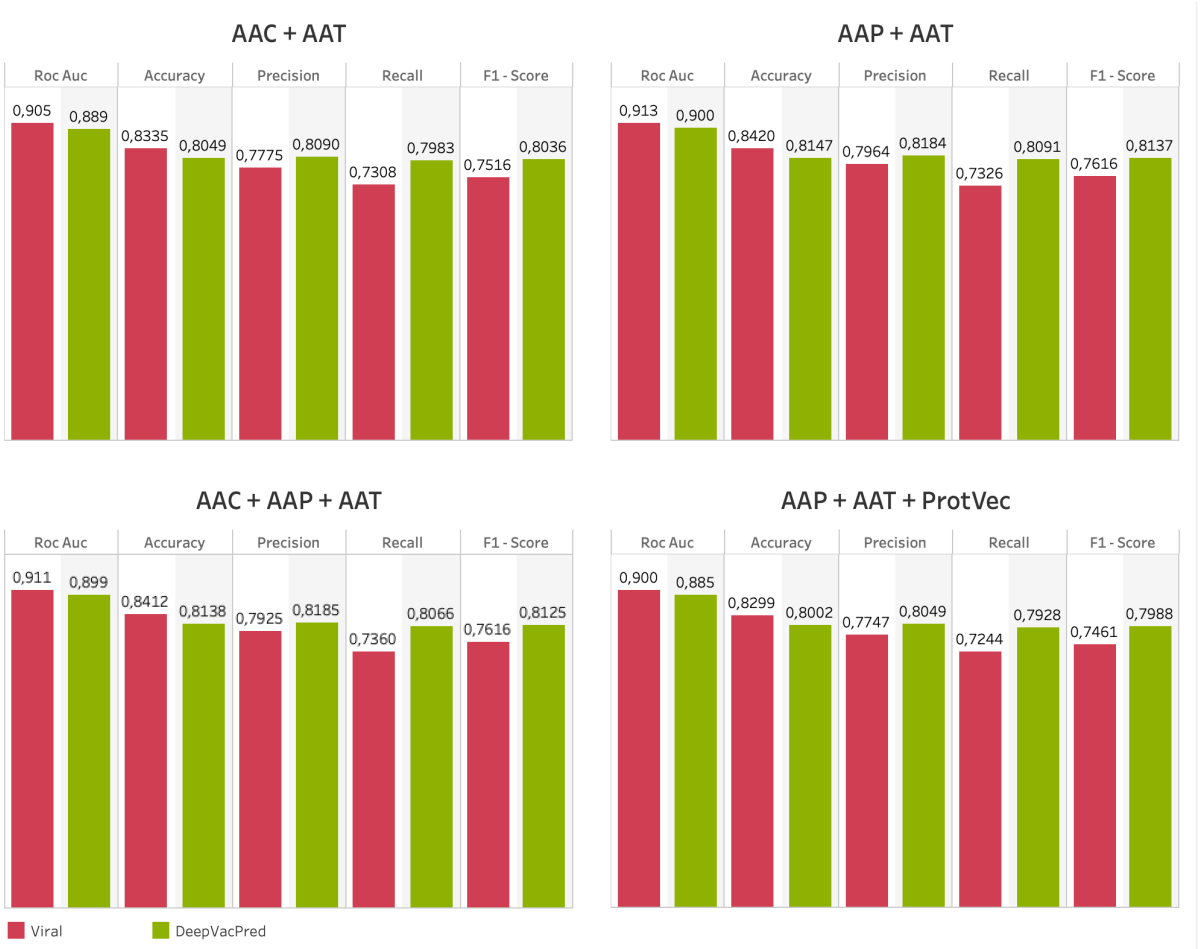
5.3 Comparação entre os resultados obtidos com os *datasets* Viral e *DeepVacPred*

Para a comparação entre os *datasets* Viral e *DeepVacPred* selecionamos as quatro melhores combinações de características demonstradas no Gráfico 5.1. Inicialmente, a ideia era testar apenas as três primeiras, porém, para testar o uso de todas as características no *dataset DeepVacPred* e pela característica ProtVec não estar presente nas três primeiras, resolvemos aumentar o número para que fosse possível tal teste. Comparamos os resultados agora com cinco métricas de desempenho, pois queríamos alcançar uma visualização mais profunda sobre as diferenças entre os dados apresentados ao mesmo modelo. Importante ressaltar que

os parâmetros utilizados foram os mesmos independente do *dataset* testado, pois o nosso objetivo aqui é verificar o desempenho das características frente aos dados com qualidades diferentes.

No Gráfico 5.2, podemos verificar os resultados obtidos para as quatro melhores composições de características. Para essas combinações o modelo apresentou a precisão, o *recall* e, consequentemente, um F1-*score* melhor quando o *dataset* usado para treinamento foi o *DeepVacPred*. Como o problema é classificar epítopos, ou seja, que a classe positiva seja avaliada e as amostras positivas sejam classificadas o mais corretamente possível, olhamos para a precisão como sendo uma ótima métrica para determinar a performance do modelo estudado.

Gráfico 5.2: Avaliação das métricas para cada *dataset* utilizado.



FONTE: Os autores

A precisão teve melhora em todos os testes realizados com o *dataset DeepVacPred*. Avaliamos que este comportamento é pelo fato do *dataset* ser mais previsível em relação aos dados, ou seja, tem os dados com as qualidades parecidas. De certa forma, como as características extraídas tem relação com o tamanho das sequências, o fato de todo o dataset ter um tamanho fixo contribuiu para que o modelo apresentasse uma melhor precisão, pois suas características foram mais consistentes e melhor distinguíveis para o modelo.

Olhando para a acurácia alcançada com o *dataset DeepVacPred,* notamos que ela ficou abaixo do que foi alcançado com o treinamento com o *dataset* Viral, dessa maneira, como a precisão foi maior para o primeiro *dataset* mencionado, concluímos que a classificação da classe negativa apresentou uma performance levemente abaixo do que se era esperado. Possivelmente, a classe negativa do *dataset DeepVacPred* tem suas características muito próximas aos da classe positiva, fazendo com que o modelo apresente uma leve tendência de classificação negativa para as amostras apresentadas.

6 CONSIDERAÇÕES FINAIS

Este trabalho buscou analisar o uso de um modelo SVM para a classificação de epítopos ligantes e não ligantes. Abordamos o problema de diversas maneiras, primeiramente, variando os parâmetros de entrada em um *kernel* RBF e definindo quais obtiveram o melhor desempenho. Em seguida, combinamos as características, agrupando todas com todas, buscando obter a melhor combinação para a classificação dos epítopos e, em um terceiro e último momento, realizamos a troca do *dataset* utilizado inicialmente para um com qualidades diferentes em suas sequências, buscando entender os impactos que o espaço amostral utilizado para treinamento tem sobre o modelo de aprendizado de máquina.

Como resultados, nos testes de variação dos parâmetros identificamos que para os *datasets* utilizados os parâmetros *gamma* que apresentaram melhor resultado ficaram entre 0.001 e 0.0001. Já para o parâmetro C, identificamos que os valores variam entre 250 e 500. Após a identificação dos melhores parâmetros, avançamos para a melhor combinação de características, sendo que, AAP + AAT foram as que tiveram melhores resultados, quando levadas em consideração a acurácia e a área sob a curva ROC, sendo de 84% e 91% respectivamente.

Por fim, decidimos testar o espaço amostral do modelo, o treinando com um *dataset* que apresentava classes extremamente balanceadas e com a quantidade de aminoácidos em sua composição fixo, diferente do primeiro utilizado. Buscando assim, analisar o impacto que os dados têm sobre os resultados que já tínhamos obtido com a variação de parâmetros e com a combinação de características. Neste testes entendemos que no caso deste projeto o espaço amostral teve impacto principalmente na precisão, o que seria desejável, visto que que acertar os epitopos ligantes é o foco do trabalho, porém, não poderíamos considerar um modelo que classifica um não epitopo como um epitopo positivo, uma vez que este, não poderia ser usado no desenvolvimento de uma vacina eficaz.

Durante a execução deste projeto, notamos uma dificuldade em encontrar *datasets* com um tamanho superior ao utilizado. Uma vez que o SVM é um modelo supervisionado, precisamos de um conjunto de dados já rotulados para executar os treinamentos desejados. Como já mencionado, a identificação de um epítopo *in-vitro*

é algo custoso e que necessita de vários experimentos para ser confirmado. Dessa maneira, a variedade de *datasets* disponíveis não é alta o que restringe os resultados que podem ser capturados, pois com um espaço amostral maior, os modelos de aprendizado de máquina têm a capacidade de entender diferentes padrões nos dados apresentados.

6.1 Trabalhos Futuros

Identificamos ao longo do trabalho que o uso de características que descrevem as relações do aminoácidos presentes nas sequências apresentaram resultados melhores frente àquelas características que apenas descreviam a quantidade de aminoácidos que a sequência tinha em sua composição. Dessa maneira, o uso de características que descrevem as propriedades físico-químicas das sequências, como é o caso do ângulo de torção de cada aminoácido, poderiam incrementar ainda mais o desempenho já obtido.

Em nossas análises, comprovamos que a característica ProtVec não apresentou um desempenho satisfatório. Entendemos que ela pode ter grande relevância, se porventura em sua concepção, o treinamento da NLP fizesse uso de um *dataset* de epítopos. Outra proposta, nesta mesma característica, é variar o tamanho dos vizinhos (*k-mers*) utilizados, pois neste trabalho utilizamos apenas *4-mers.*

REFERÊNCIAS

ABBAS, Abul K. *et al*. **Imunologia Básica: Funções e distúrbios do sistema imunológico.** 4. ed. rev. Rio de Janeiro: Elsevier Editora Ltda, 2013. 336 p. ISBN 978-85-352-7682-4.

ABBAS, Abul K. *et al*. **Imunologia Celular e Molecular.** 8. ed. rev. Rio de Janeiro: Elsevier Editora Ltda, 2015. 552 p. ISBN 978-85-352-8164-4.

ASGARI, Ehsaneddin; MOFRAD , Mohammad R. K. **Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics.** PLOS ONE , [*S. l.*], p. 1-15, 10 nov. 2015. DOI https://doi.org/10.1371/journal.pone.0141287. Disponível em: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0141287. Acesso em: 9 out. 2021.

BAHAI, Akash; ASGARI, Ehsaneddin; MOFRAD, Mohammad R K; KLOETGEN, Andreas; MCHARDY, Alice C. **EpitopeVec: linear epitope prediction using deep protein sequence embeddings.** Bioinformatics, [*S. l.*], ano 2021, p. 1-9, 28 jun. 2021. DOI https://doi.org/10.1093/bioinformatics/btab467. Disponível em: https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btab467/6310734#283768450. Acesso em: 4 out. 2021.

BOSER, Bernhard E. *et al*. **A training algorithm for optimal margin classifiers.** Association for Computing Machinery, Pittsburgh, Pennsylvania, USA, p. 144–152, 1 jul. 1992. DOI https://doi.org/10.1145/130385.130401. Disponível em: https://dl.acm.org/doi/10.1145/130385.130401. Acesso em: 1 dez. 2021.

CAO, Dong-Sheng *et al*. **PyDPI: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies.** J Chem Inf Model., [*S. l.*], v. 53, n. 11, p. 3086-96, 25 nov. 2013. DOI https://doi.org/10.1021/ci400127q. Disponível em: https://pubs.acs.org/doi/10.1021/ci400127q. Acesso em: 27 nov. 2021.

CHEN, J. *et al*. **Prediction of linear B-cell epitopes using amino acid pair antigenicity scale.** Amino Acids, [*S. l.*], n. 33, p. 423–428, 17 nov. 2021. DOI https://doi.org/10.1007/s00726-006-0485-9. Disponível em: https://doi.org/10.1007/s00726-006-0485-9. Acesso em: 30 out. 2021.

DENG, Xiaoxiang; STORZ, Ulrich; DORANZ, Benjamin J. **Enhancing antibody patent protection using epitope mapping information.** MAbs, Philadelphia, PA, USA, v. 10, n. 2, p. 204-209, 7 dez. 2017. DOI 10.1080/19420862.2017.1402998. Disponível em: https://doi.org/10.1080/19420862.2017.1402998. Acesso em: 30 set. 2021.

GÉRON, Aurélien. **Mãos à Obra:** Aprendizado de Máquina com Scikit-Learn & TensorFlow. Rio de Janeiro (RJ): Alta Book, 2019. 576 p. ISBN 978-85-508-0902-1.

GRIFONI, Alba *et al*. **A Sequence Homology and Bioinformatic Approach Can Predict Candidate Targets for Immune Responses to SARS-CoV-2.** Cell Host & Microbe, [*S. l.*], v. 27, n. 4, p. 671-680, 8 abr. 2020. DOI https://doi.org/10.1016/j.chom.2020.03.002. Disponível em: https://www.sciencedirect.com/science/article/pii/S1931312820301669. Acesso em: 22 nov. 2021.

MUTNEJA, Manpreet *et al*. **An introduction to Antibodies and their applications.** 3. ed. [*S. l.*]: EMD Millipore, 2014. 100 p. Disponível em: https://www.researchgate.net/publication/280494428_An_introduction_to_Antibodies_and_their_applications>. Acesso em: 15 set. 2021.

NOBLE, William S. **What is a support vector machine?**. NATURE BIOTECHNOLOGY , [*S. l.*], v. 12, n. 24, p. 1565-1567, 1 dez. 2006. DOI https://doi.org/10.1038/nbt1206-1565. Disponível em: https://www.nature.com/articles/nbt1206-1565#citeas. Acesso em: 1 dez. 2021.

PARVIZPOUR , Sepideh *et al*. **Epitope-based vaccine design: a comprehensive overview of bioinformatics approaches.** Drug Discovery Today, [*S. l.*], v. 25, n. 6, p. 1034-1042, 20 mar. 2020. DOI 10.1016/j.drudis.2020.03.006. Disponível em: 10.1016/j.drudis.2020.03.006. Acesso em: 30 set. 2021.

PEDREGOSA, Fabian *et al*. **RBF SVM parameters.** Journal of Machine Learning Research, 2011a. Disponível em: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html. Acesso em: 01 dez. 2021.

PEDREGOSA, Fabian *et al*. **Scikit-learn: Machine Learning in Python**. Journal of Machine Learning Research, [*S. l.*], v. 12, p. 2825--2830, 2011b. Disponível em: https://scikit-learn.org/stable/getting_started.html#getting-started. Acesso em: 26 nov. 2021.

PEDREGOSA, Fabian *et al*. **Tuning the hyper-parameters of an estimator: Exhaustive Grid Search.** Journal of Machine Learning Research, 2011c. Disponível em: https://scikit-learn.org/stable/modules/grid_search.html. Acesso em: 26 nov. 2021.

SAHA, Sudipto; RAGHAVA, G P S. **Prediction of Continuous B-Cell Epitopes in an Antigen Using Recurrent Neural Network.** Wiley InterScience, Chandigarh, India, v. 65, n. 1, 1 out. 2006. Proteins, p. 40-48. DOI 10.1002/prot.21078. Disponível em: https://webs.iiitd.edu.in/raghava/reprints/abcpred.pdf. Acesso em: 21 out. 2021.

SALEM, Dina Ahmed; SEOUD, Rania Ahmed Abul; KADAH, Yasser. **Conformational B-cell epitopes classification using machine learning techniques.** Journal of Engineering and Applied Science, [*S. l.*], v. 60, n. 3, p. 343-359, 1 jun. 2013. Disponível em: https://www.researchgate.net/publication/272071925_Conformational_B-cell_epitopes_classification_using_machine_learning_techniques. Acesso em: 21 out. 2021.

SELA-CULANG, Inbal; KUNIK, Vered; OFRAN, Yanay. **The Structural Basis of Antibody-Antigen Recognition.** Frontiers in Immunology , [*s. l.*], v. 4, p. 302, 8 out. 2013. DOI https://doi.org/10.3389/fimmu.2013.00302. Disponível em: https://www.frontiersin.org/articles/10.3389/fimmu.2013.00302/full. Acesso em: 13 out. 2021.

TURING, A. M. **COMPUTING MACHINERY AND INTELLIGENCE**. Mind, [*S. l.*], v. LIX, n. 236, p. 433–460, 1 out. 1950. DOI https://doi.org/10.1093/mind/LIX.236.433. Disponível em: https://academic.oup.com/mind/article/LIX/236/433/986238. Acesso em: 30 nov. 2021.

TYPES of Unsupervised Learning. *In*: **Supervised Machine Learning, Unsupervised Machine Learning, and Deep Learning**. [*S. l.*]: AnalystPrep, 6 mar. 2021. Disponível em: https://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning/. Acesso em: 27 nov. 2021.

YANG, Zikun *et al*. **An in silico deep learning approach to multi-epitope vaccine design: a SARS-CoV-2 case study.** Scientific Reports, [*S. l.*], v. 11, n. 3238, p. 1-21, 5 fev. 2021. DOI 10.1038/s41598-021-81749-9. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7865008/. Acesso em: 9 out. 2021.

YAO, Bo *et al*. SVMTriP: **A Method to Predict Antigenic Epitopes Using Support Vector Machine to Integrate Tri-Peptide Similarity and Propensity.** PLOS ONE, [*S. l.*], v. 7, n. 9, 12 set. 2012. e45152, p. 1-5. DOI https://doi.org/10.1371/journal.pone.0045152. Disponível em: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0045152. Acesso em: 29 out. 2021.

ZHANG, Wen *et al*. **Accurate Prediction of Immunogenic T-Cell Epitopes from Epitope Sequences Using the Genetic Algorithm-Based Ensemble Learning.** PLoS One, [*s. l.*], ano e0128194, v. 10, ed. 5, 28 maio 2015. DOI https://doi.org/10.1371/journal.pone.0128194. Disponível em https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0128194. Acesso em: 30 set. 2021.

APÊNDICE A

A.1 Os 10 melhores resultados para cada característica.

A seguir apresentamos os 10 melhores resultados de área sob a curva ROC e Acurácia das variações dos parâmetros do modelo para cada combinação de características.

A.1.1 Característica AAC

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC	0.1	0.1	0,59102	0,65363
		0.01	0,59101	0,65627
		0.001	0,58443	0,65627
		1	0,59100	0,64138
	0.0001	0.1	0,59183	0,65627
		1	0,59532	0,65627
		25	0,58571	0,65627
		50	0,58224	0,65627
		100	0,58275	0,65627
		250	0,58227	0,65627

A.1.2 Característica AAP

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAP	0.01	0.001	0,659771	0,656272
		0.0001	0,659977	0,656272
	0.001	0.01	0,659977	0,656272
		0.001	0,659977	0,656272
		0.0001	0,659977	0,656272
	0.0001	0.1	0,659977	0,656272
		0.01	0,659977	0,656272
		0.001	0,659977	0,656272
		0.0001	0,659977	0,656272
		1	0,659977	0,656272

A.1.3 Característica AAT

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAT	0.01	1	0,89916	0,82709
	0.001	25	0,89916	0,82686
		50	0,89916	0,82693
		100	0,89916	0,82693
		250	0,89916	0,82678
	0.0001	50	0,89916	0,82701
		100	0,89916	0,82717
		250	0,89916	0,82693
		500	0,89916	0,82725
		1000	0,89916	0,82732

A.1.4 Característica ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
ProtVec	0.01	1	0,61648	0,64479
	0.001	1	0,61371	0,67605
		25	0,61504	0,63393
	0.0001	0.001	0,61511	0,65627
		25	0,61166	0,67419
		50	0,61398	0,67349
		100	0,61589	0,67217
		250	0,61669	0,67116
		500	0,61370	0,66496
		1000	0,60975	0,65317

A.1.5 Características AAC + AAP

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC + AAP	0.01	0.1	0,64104	0,65627
		0.01	0,63967	0,65627
		1	0,64162	0,65906
	0.001	25	0,63951	0,65503
		50	0,64047	0,65643
	0.0001	1	0,64077	0,65627
		25	0,64057	0,65627
		50	0,64000	0,65627
		100	0,63957	0,65627
		250	0,63940	0,65627

A.1.6 Características AAC + AAT

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC + AAT	0.01	1	0,90465	0,83337
	0.001	25	0,90495	0,83322
		50	0,90447	0,83252
		100	0,90374	0,83291
	0.0001	25	0,90361	0,83066
		50	0,90433	0,83190
		100	0,90464	0,83275
		250	0,90493	0,83384
		500	0,90507	0,83368
		1000	0,90510	0,83353

A.1.7 Características AAC + ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC + ProtVec	0.01	1	0,61543	0,64595
		1000	0,60700	0,62470
	0.001	1	0,61231	0,67559
		25	0,60979	0,61725
	0.0001	0.001	0,61623	0,65627
		25	0,60774	0,67403
		50	0,61015	0,67248
		100	0,61198	0,67124
		250	0,61323	0,67000
		500	0,60967	0,65906

A.1.8 Características AAP + AAT

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAP + AAT	0.01	1	0,91328	0,84214
	0.001	25	0,91336	0,84253
		50	0,91336	0,84245
		100	0,91336	0,84245
		250	0,91338	0,84198
		500	0,91336	0,84167
		1000	0,91332	0,84222
	0.0001	250	0,91336	0,84268
		500	0,91337	0,84245
		1000	0,91335	0,84230

A.1.9 Características AAP + ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAP + ProtVec	0.01	1	0,63250	0,64812
	0.001	1	0,64401	0,67559
		25	0,63723	0,64541
	0.0001	1	0,63030	0,67279
		25	0,64265	0,67295
		50	0,64546	0,67396
		100	0,64673	0,67644
		250	0,64679	0,67667
		500	0,64428	0,67086
		1000	0,63718	0,65705

A.1.10 Características AAT + ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAT + ProtVec	0.001	1	0,8848	0,8162
		25	0,8676	0,7949
		50	0,8518	0,7788
	0.0001	1	0,8536	0,7451
		25	0,8936	0,8242
		50	0,8938	0,8231
		100	0,8937	0,8233
		250	0,8919	0,8227
		500	0,8875	0,8177
		1000	0,8785	0,8085

A.1.11 Características AAC + AAP + AAT

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC + AAP + AAT	0.01	1	0,90981	0,83997
	0.001	25	0,91073	0,84121
		50	0,91022	0,84028
		100	0,90919	0,83958
	0.0001	25	0,90925	0,83818
		50	0,91042	0,83981
		100	0,91102	0,84082
		250	0,91132	0,84098
		500	0,91137	0,84121
		1000	0,91125	0,84229

A.1.12 Características AAC + AAP + ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC + AAP + ProtVec	0.01	1	0,63049	0,65037
	0.001	1	0,64093	0,67574
		25	0,62903	0,62866
	0.0001	1	0,62835	0,67264
		25	0,64046	0,67365
		50	0,64237	0,67566
		100	0,64358	0,67566
		250	0,64186	0,67202
		500	0,63742	0,66465
		1000	0,62972	0,65069

A.1.13 Características AAC + AAT + ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC + AAT + ProtVec	0.001	1	0,8838	0,8172
		25	0,8625	0,7901
		50	0,8471	0,7748
	0.0001	1	0,8516	0,7441
		25	0,8932	0,8226
		50	0,8939	0,8219
		100	0,8935	0,8234
		250	0,8908	0,8204
		500	0,8855	0,8151
		1000	0,8750	0,8044

A.1.14 Características AAP + AAT + ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAP + AAT + ProtVec	0.001	1	0,8865	0,8186
		25	0,8716	0,8010
		50	0,8562	0,7847
	0.0001	1	0,8516	0,7470
		25	0,8983	0,8305
		50	0,8997	0,8308
		100	0,8998	0,8299
		250	0,8973	0,8286
		500	0,8926	0,8232
		1000	0,8834	0,8117

A.1.15 Características AAC + AAP + AAT + ProtVec

Combinação de Características	Gamma	C	Roc Auc	Accuracy
AAC + AAP + AAT + ProtVec	0.001	1	0,8851	0,8173
		25	0,8672	0,7954
		50	0,8517	0,7808
	0.0001	1	0,8498	0,7467
		25	0,8974	0,8278
		50	0,8990	0,8308
		100	0,8992	0,8310
		250	0,8963	0,8282
		500	0,8904	0,8217
		1000	0,8799	0,8092