Universidade Estadual de Campinas
Instituto de Computação

Gabriel Bianchin de Oliveira

Predição de Estruturas Secundárias de Proteínas usando Aprendizado de Máquina e BLAST

CAMPINAS
2021

Gabriel Bianchin de Oliveira

Predição de Estruturas Secundárias de Proteínas usando Aprendizado de Máquina e BLAST

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Zanoni Dias
Coorientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Dissertação defendida por Gabriel Bianchin de Oliveira e orientada pelo Prof. Dr. Zanoni Dias.

CAMPINAS
2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

OL4p

Oliveira, Gabriel Bianchin de, 1997-
Predição de estruturas secundárias de proteínas usando aprendizado de máquina e BLAST / Gabriel Bianchin de Oliveira. – Campinas, SP : [s.n.], 2021.

Orientador: Zanoni Dias.
Coorientador: Hélio Pedrini.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina. 2. Aprendizado profundo. 3. Proteínas - Estrutura. I. Dias, Zanoni, 1975-. II. Pedrini, Hélio, 1963-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Protein secondary structure prediction using machine learning and BLAST
Palavras-chave em inglês:
Machine learning
Deep learning
Proteins - Structure
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Zanoni Dias [Orientador]
Ricardo Cerri
Guilherme Pimentel Telles
Data de defesa: 11-03-2021
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-1238-4860
- Currículo Lattes do autor: http://lattes.cnpq.br/8002321277160438

Universidade Estadual de Campinas
Instituto de Computação

Gabriel Bianchin de Oliveira

Predição de Estruturas Secundárias de Proteínas usando Aprendizado de Máquina e BLAST

Banca Examinadora:

- Prof. Dr. Zanoni Dias
  Instituto de Computação - Unicamp
- Prof. Dr. Ricardo Cerri
  Departamento de Computação - UFSCar
- Prof. Dr. Guilherme Pimental Telles
  Instituto de Computação - Unicamp

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 11 de março de 2021

Agradecimentos

Agradeço à minha família, pelo apoio durante todo o percurso até a Unicamp. Agradeço principalmente aos meus pais, Elizabeth e João, aos meus avós, Madalena e Pedro, e aos meus tios, Eliana, Elaine e Renato, por todo suporte durante esses dois últimos anos. Infelizmente, minha tia Eliana não pode acompanhar meu trabalho finalizado, mas acredito que ela ficaria feliz com a conclusão desta etapa em minha vida.

Agradeço aos meus amigos, em especial ao Vicente, que sempre me ajudou quando precisei.

Agradeço à Mirian, por todo apoio e carinho.

Agradeço aos meus orientadores, Zanoni Dias e Hélio Pedrini, pela orientação, dedicação e confiança nesses dois anos de mestrado. Aprendi muito com vocês.

Agradeço aos membros do LIV, em especial ao Vinicius, Leodécio, Tiago, Daiane e Marianna, pelos comentários que auxiliaram no desenvolvimento da pesquisa.

Agradeço aos professores Ricardo Martins, Gustavo Silva e Carlos Renato, que me apoiaram e me incentivaram a entrar na pós-graduação da Unicamp.

Agradeço também a todos que direta ou indiretamente me fizeram chegar até aqui.

Resumo

Proteínas, que são sequências de aminoácidos, são fundamentais em diversos processos biológicos dos seres vivos. Devido às interações físicas e químicas entre os aminoácidos que formam as proteínas, estruturas tridimensionais locais e globais são formadas. Com os avanços tecnológicos na área biológica, o sequenciamento de proteínas se tornou simples e rápido de ser feito. Por outro lado, a definição de estruturas tridimensionais locais, chamadas de estruturas secundárias, e globais, chamadas de estruturas terciárias, continua custosa. Estruturas tridimensionais têm alto impacto na definição de funções de proteínas e no auxílio ao desenvolvimento de aplicações, como remédios e biossensores.

Como opção para a definição de estruturas globais das proteínas a partir da sequência de aminoácidos, a análise de estruturas secundárias se tornou o principal método intermediário na literatura. Para realizar a predição de estruturas secundárias, duas abordagens são mais comumente utilizadas, sendo elas métodos baseados em modelo, que usam ferramentas que encontram proteínas similares, e métodos livres de modelo, que usam classificadores de aprendizado de máquina. Nos trabalhos recentes, diversas metodologias foram propostas para predizer estruturas secundárias, porém este problema continua em aberto. Outro ponto importante nos métodos atuais é que a maioria das abordagens utiliza informações evolutivas além da sequência de aminoácidos que formam as proteínas, sendo incapazes de predizer estruturas secundárias utilizando apenas a cadeia de aminoácidos.

Nesta pesquisa, propomos diversos classificadores baseados em modelo e livres de modelo para realizar a classificação de estruturas secundárias das proteínas. Além da análise individual dos classificadores, investigamos a fusão entre os preditores baseados em modelo e preditores livres de modelo, assim como a fusão entre todos os classificadores. Nossos preditores são capazes de classificar estruturas secundárias a partir de sequências de aminoácidos com ou sem informações evolutivas, o que não é possível para a maioria dos métodos disponíveis na literatura. Os resultados obtidos em três bases de dados diferentes mostram que nossos classificadores são competitivos comparados com as abordagens da literatura.

Abstract

Proteins, which are sequences of amino acids, are fundamental in several biological processes of living beings. Due to physical and chemical interactions between the amino acids that form proteins, local and global three-dimensional structures are formed. With technological advances in the biological area, protein sequencing has become simple and quick to be done. On the other hand, the definition of local three-dimensional structures, called secondary structures, and global three-dimensional structures, called tertiary structures, remains costly. Three-dimensional structures have a high impact on the definition of protein functions and the aid of application development, such as medicines and biosensors.

As an option for the definition of global protein structures from the amino acid sequence, the analysis of secondary structures has become the main intermediate method in the literature. To perform the prediction of secondary structures, two approaches are most commonly used, namely template-based methods, which use tools that find similar proteins, and template-free methods, which use machine learning classifiers. In recent works, several methodologies have been proposed to predict secondary structures, but this problem remains open. Another important point in current methods is that most approaches use evolutionary information in addition to the sequence of amino acids that form proteins, being unable to predict secondary structures using only the chain of amino acids.

In this research, we propose several template-based and template-free models to classify secondary structures of proteins. In addition to the individual classifier analysis, we investigated the fusion between template-based predictors and template-free predictors, as well as the fusion between all classifiers. Our predictors can classify secondary structures from amino acid sequences with or without evolutionary information, which is not possible for most methods available in the literature. The results obtained in three different databases show that our classifiers are competitive compared to the approaches in the literature.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Neste capítulo, descrevemos o problema sob investigação nesta dissertação, assim como a motivação, as questões de pesquisa, os objetivos e a organização do texto.

1.1 Contexto e Motivação

Proteínas são importantes em diversos processos biológicos dos seres vivos. Elas possuem diversas funções nas células, como regulação de reações, resposta imunológica e transporte. As proteínas são formadas por uma sequência de unidades menores, os aminoácidos [66]. A sequência de aminoácidos constitui a estrutura primária da proteína [45].

Existem 20 diferentes tipos de aminoácidos que podem formar uma proteína. A quantidade de configurações de sequências de aminoácidos é potencialmente infinita, considerando o tamanho da sequência e a frequência de cada aminoácido. Entretanto, o genoma humano possui apenas 35.000 proteínas diferentes [88].

Devido às interações físicas e químicas entre os aminoácidos, as proteínas formam estruturas tridimensionais [96]. Estas estruturas são divididas em enovelamentos que cada um dos aminoácidos forma, chamados de estruturas secundárias, e o enovelamento da proteína como um todo, chamado de estrutura terciária.

As estruturas tridimensionais impactam nas funções exercidas pelas proteínas, já que cada função depende de um enovelamento específico [13], além de que cada proteína pode exercer mais que uma função simultaneamente [12]. Algumas doenças, como fibrose cística, Alzheimer e outras doenças neurodegenerativas, são associadas ao enovelamento incorreto da proteína [57].

Entender as estruturas tridimensionais pode auxiliar no desenvolvimento de novas aplicações, como criação de medicamentos, biossensores e estudo de enzimas [39, 95, 96].

A análise do enovelamento global das proteínas pode ser feita a partir da estrutura primária, como foi mostrado em um trabalho recente [73], porém este problema ainda continua em aberto. O método mais comum usado na literatura consiste em primeiro entender as estruturas secundárias e depois predizer a estrutura terciária das proteínas.

Para determinar as estruturas secundárias, métodos laboratoriais são necessários, como cristalografia de radiografia, espectroscopia de ressonância magnética nuclear e microscopia eletrônica [45], porém estes métodos são caros e lentos.

Diferente das estruturas secundárias, o sequenciamento de aminoácidos que constituem as proteínas se tornou mais barato recentemente. A diferença entre o número de proteínas sequenciadas e proteínas com estruturas secundárias definidas cresce a cada ano [54]. A Figura 1.1 demonstra a diferença ao longo dos últimos 20 anos no número de proteínas sequenciadas na base UniProtKB, que é o principal repositório de proteínas sequenciadas, com o número de proteínas com estruturas secundárias definidas na base *Protein Data Bank* (PDB), que é o principal repositório de proteínas com estruturas secundárias catalogadas por métodos laboratoriais.

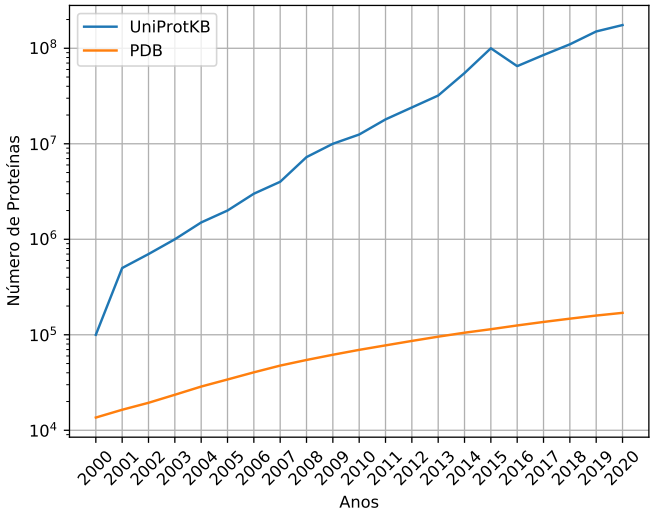


Figura 1.1: Número de proteínas sequenciadas na base UniProtKB e proteínas com estruturas secundárias definidas na base PDB ao longo dos últimos 20 anos.

Com a diferença entre o número de proteínas sequenciadas e proteínas com estruturas secundárias definidas, a classificação experimental de estruturas secundárias não se mostra viável. Com isso, outros métodos para predizer estruturas secundárias se tornam cada vez mais importantes, como métodos computacionais [24, 28].

Os métodos para a predição de estruturas secundárias disponíveis na literatura são divididos em métodos baseados em modelo e métodos livres de modelo. Os métodos baseados em modelo usam ferramentas que verificam a similaridade entre proteínas no treino e teste [74]. Uma destas ferramentas que podem ser utilizadas para métodos baseados em modelo é o BLAST [3], que é uma ferramenta que analisa, busca e encontra alinhamentos locais similares de sequências de proteínas.

Os métodos livres de modelo usam, principalmente, aprendizado de máquina e são capazes de predizer estruturas secundárias de proteínas no teste a partir de proteínas no treino, sendo que estes dois conjuntos possuem dados com baixa similaridade. Como entrada de dados, a maioria dos classificadores utilizam a sequência de aminoácidos em um vetor *one-hot encoding*, onde os dados categóricos representando aminoácidos são transformados em dados numéricos binários, que são esparsos devido a dimensionalidade, e informações evolutivas, como matriz de pontuação de posição específica.

Os classificadores de aprendizado de máquina aplicados para predizer estruturas secundárias de proteínas são divididos entre classificadores locais e globais. Os classificadores locais utilizam a análise local das interações entre aminoácidos, enquanto classificadores

globais verificam a sequência inteira da proteína [54]. As duas metodologias possuem desvantagens, como a dependência de interações próximas para os classificadores locais e o custo computacional para os classificadores globais.

A principal desvantagem dos métodos livres de modelo mais recentes disponíveis na literatura é que a maioria deles não pode predizer estruturas secundárias utilizando apenas a sequência de aminoácidos. A utilização de apenas a sequência de aminoácidos para predizer estruturas secundárias é importante para grandes bases de dados, principalmente pelo tempo necessário para gerar informações evolutivas [21].

1.2 Questões de Pesquisa

Para alcançar e melhorar os resultados do problema sob investigação, propomos algumas questões para guiar a pesquisa:

- Utilizar apenas a sequência de aminoácidos pode produzir resultados próximos aos resultados utilizando sequência de aminoácidos e informações evolutivas?
- A transformação do vetor *one-hot encoding* esparso em um vetor denso pode ajudar na classificação de estruturas secundárias das proteínas?
- Qual é o impacto da fusão de classificadores locais e globais na classificação?
- A fusão entre métodos baseados em modelo e livres de modelo pode melhorar os resultados?

1.3 Objetivos e Contribuições

O principal objetivo deste trabalho é propor, implementar e avaliar métodos usando classificadores de aprendizado de máquina e BLAST para predizer estruturas secundárias de proteínas. Para alcançar o objetivo, os seguintes pontos foram definidos:

- Definição de métodos livres de modelos usando diferentes algoritmos com classificação local, global e local-global.
- Avaliação da fusão de diferentes classificadores livres de modelo.
- Investigação do BLAST como um método baseado em modelo.
- Avaliação da fusão entre métodos livres de modelo e baseados em modelo.

As principais contribuições deste trabalho são:

- Investigação da importância da matriz de pontuação de posição específica na predição de estruturas secundárias de proteínas.
- Desenvolvimento e avaliação de um método de fusão que utiliza algoritmos de otimização, que chamamos de sacola de otimizadores.

- Avaliação de *Transformers* no problema de predição de estruturas secundárias de proteínas.
- Avaliação de fusões entre métodos livres de modelo e baseados em modelo.

1.4 Publicações

Os seguintes artigos foram gerados a partir do desenvolvimento desta pesquisa:

- G. B. de Oliveira, H. Pedrini e Z. Dias [60]. Ensemble of Bidirectional Recurrent Networks and Random Forests for Protein Secondary Structure Prediction. 27th International Conference on Systems, Signals and Image Processing (IWSSIP). Niterói, Rio de Janeiro, Brasil. 2020. pp. 311-316.
- G. B. de Oliveira, H. Pedrini e Z. Dias [61]. Fusion of BLAST and Ensemble of Classifiers for Protein Secondary Structure Prediction. 33rd Conference on Graphics, Patterns and Images (SIBGRAPI). Porto de Galinhas, Pernambuco, Brasil. 2020. pp. 308-315.
- G. B. de Oliveira, H. Pedrini e Z. Dias [62]. Protein Secondary Structure Prediction Based on Fusion of Machine Learning Classifiers. 36th ACM/SIGAPP Symposium On Applied Computing - Bioinformatics Track (ACM SAC BIO). Gwangju, Jeolla do Sul, Coreia do Sul. 2021. pp. 26-29.

1.5 Organização do Texto

O restante do texto está organizado da seguinte forma. No Capítulo 2, detalhamos alguns conceitos relevantes relatados com o tópico sob investigação. No Capítulo 3, apresentamos os trabalhos relacionados com a predição de estruturas secundárias das proteínas. No Capítulo 4, apresentamos as bases de dados utilizadas e as métricas de avaliação. No Capítulo 5, descrevemos o método proposto para a predição de estruturas secundárias. No Capítulo 6, apresentamos e analisamos os resultados experimentais utilizando sequência de aminoácidos e no Capítulo 7, apresentamos e analisamos os resultados experimentais com sequência de aminoácidos e matriz de pontuação de posição específica. No Capítulo 8, descrevemos as conclusões e possíveis linhas de pesquisa para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Neste capítulo, apresentamos alguns conceitos biológicos e computacionais utilizados na pesquisa, que servem como base para o entendimento deste trabalho.

2.1 Conceitos Biológicos

Nesta seção, descrevemos sucintamente alguns conceitos biológicos relacionados ao tema sob investigação.

2.1.1 Proteínas

Proteínas são macromoléculas que estão presentes em todos os organismos vivos, responsáveis por diversos processos biológicos e funções, como proteção, regulação de reações químicas e transporte [57]. Elas são formadas por sequências de aminoácidos conectados por ligações peptídicas de hidrogênio.

As proteínas possuem quatro diferentes estruturas. A estrutura primária da proteína consiste na sequência linear dos aminoácidos [45]. As estruturas secundárias da proteína ocorrem devido às interações físicas e químicas entre as ligações de hidrogênio dos aminoácidos [57], formando diversas estruturas tridimensionais em cada um dos polipeptídeos que formam a sequência [53]. A estrutura terciária da proteína é representada pelo conjunto de todas os enovelamentos formados na cadeia polipeptídica da proteína [1]. Algumas proteínas possuem uma estrutura quaternária, que é formada por um complexo de duas ou mais cadeias polipeptídicas [68].

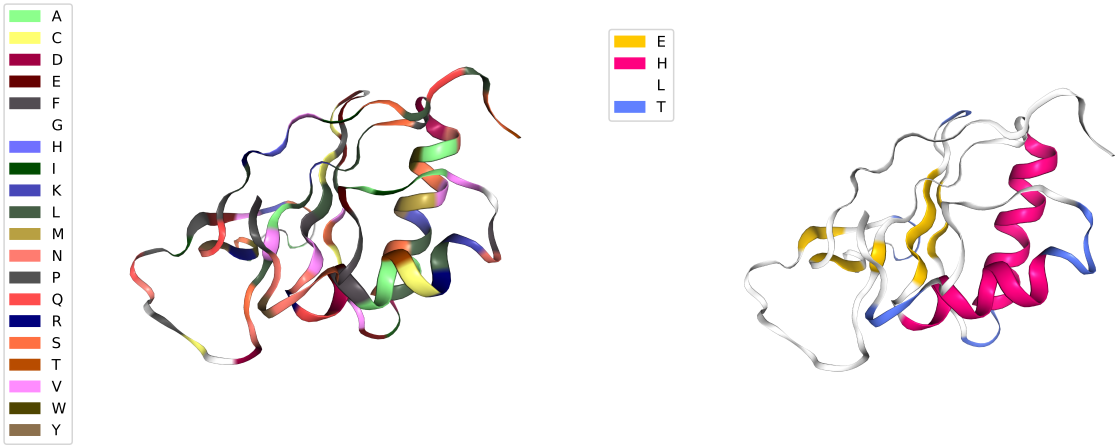
Nas bases de dados, como o *Protein Data Bank* (PDB) [7], as proteínas são depositadas em arquivos do tipo FASTA. Este tipo de arquivo possui informações sobre o nome da proteína e a sequência de aminoácidos. A Figura 2.1a apresenta o arquivo FASTA da proteína PDB ID: 6BI6.

2.1.2 Aminoácidos

Os aminoácidos são as unidades básicas que formam as sequências das proteínas. Ao todo, existem 20 diferentes aminoácidos que podem formar a sequência de cada proteína. A

```
>6BI6:A|PDBID|CHAIN|SEQUENCE
GPTSLQLSIVHRLPQNYRWSAGFAGSKVEPIPQNGPCGDNSLVALKLLSPDGDNAWSVMYKLSQALSDIEV
PCSVLECEGEPCLFVNRQDEFAATCRLKNFGVAIAEPFSNYNPF
```

(a) Arquivo FASTA contendo a sequência de aminoácidos.



(b) Aminoácidos na forma tridimensional.

(c) Estruturas secundárias.

Figura 2.1: Sequência de aminoácidos e estruturas secundárias da proteína PDB ID: 6BI6.

Tabela 2.1 apresenta todos os 20 aminoácidos e o código padrão correspondente a cada um.

Em algumas bases de dados, alguns aminoácidos possuem mais do que uma letra como representação, como Alanina, que é representada pelas letras “A” e “X”, Asparagina, que é representada pelas letras “N” e “B” e Glutamina, que é representada pelas letras “Q” e “Z”.

Aminoácido	**Código**	**Aminoácido**	**Código**
Alanina	A	Isoleucina	I
Arginina	R	Leucina	L
Asparagina	N	Lisina	K
Aspartato	D	Metionina	M
Cisteina	C	Prolina	P
Fenilalanina	F	Serina	S
Glicina	G	Tirosina	Y
Glutamato	E	Treonina	T
Glutamina	Q	Triptofano	W
Histidina	H	Valina	V

Tabela 2.1: Aminoácidos e os códigos correspondentes.

2.1.3 Estruturas Secundárias

As estruturas secundárias das proteínas são as estruturas tridimensionais que cada um dos aminoácidos forma, que ocorrem devido às interações físicas e químicas entre os aminoá-

cidos que compõem a proteína. As estruturas formadas são energeticamente eficientes [1]. A Figura 2.1b mostra a estrutura terciária da proteína PDB ID: 6BI6, destacando os aminoácidos, enquanto a Figura 2.1c apresenta as estruturas secundárias que cada um dos aminoácidos forma.

Para a categorização de estruturas secundárias, existem duas diferentes classificações. Na classificação Q3, cada um dos aminoácidos pode pertencer a uma de três possíveis classes, enquanto na classificação Q8 existem oito diferentes classes, sendo a categorização Q8 uma subclassificação da Q3 [49]. Na classificação Q3, as classes são H (hélice), E (folha) e C (espiral) [22]. Na classificação Q8, as classes são H (hélice alfa), G (3-hélice), B (resíduo em folha beta isolada), E (folha estendida), I (5-hélice), T (torção de ligação de hidrogênio), S (torção) e L (espiral) [49].

Existem cinco métodos para dividir a classificação Q3 em classificação Q8. A Tabela 2.2 descreve as cinco variações. O mapeamento de Q3 para Q8 não é padronizado devido a não possuir uma clara fronteira biológica entre estruturas secundárias.

	Classes Q8				
Classes Q3	**Método 1**	**Método 2**	**Método 3**	**Método 4**	**Método 5**
C	I, L, S, T	B, G, I, L, S, T	L, S, T	B, I, L, S, T	B, L, S, T
E	B, E	E	B, E	E	E
H	G, H	H	G, H, I	G, H	G, H, I

Tabela 2.2: Métodos para agrupamento de classes.

Entre as oito classes da categoria Q8, algumas estruturas são mais frequentes do que outras, o que torna a classificação Q8 em um problema desbalanceado. As classes mais comuns são H, L, E e T, enquanto as classes restantes são menos frequentes.

2.1.4 Matriz de Pontuação de Posição Específica

Além da sequência de aminoácidos, a matriz de pontuação de posição específica é utilizada para realizar a predição de estruturas secundárias na maioria dos trabalhos recentes da literatura, muito por conta dos resultados atingidos por Jones [37], que demonstrou que essa característica auxilia na predição de estruturas secundárias.

A matriz de pontuação de posição específica é usada para determinar o alinhamento de sequências de proteínas e pode ser utilizada para avaliar proteínas que possuem sequências similares e que são distintas evolutivamente [14].

Para fazer o cálculo da matriz de pontuação de posição específica, uma base de dados com grande quantidade de proteínas é necessária, onde seja possível encontrar múltiplas proteínas com alinhamentos próximos [27]. A principal base de dados para fazer a consulta é a UniRef [76]. A ferramenta PSI-BLAST [3] é responsável por buscar as proteínas com sequências similares e gerar a matriz de pontuação de posição específica.

A partir das proteínas homólogas encontradas, o cálculo da matriz de posição específica atribui um vetor de tamanho 20 para cada um dos aminoácidos da proteína, levando em conta a quantidade de proteínas homólogas, a quantidade de aminoácidos na mesma posição específica e o número de aminoácidos nas proteínas homólogas [14].

Primeiro, é necessário calcular o número de aminoácidos j na posição i entre todas as N proteínas homólogas. Depois, calcula-se o *score* de cada aminoácido j na posição específica i da sequência, levando em conta o número $F_{i,j}$ de aminoácidos j na posição i e a quantidade de proteínas homólogas, conforme a Equação 2.1. Depois, deve-se calcular a taxa de frequência P_j de cada aminoácido j estar presente nas proteínas homólogas. Por fim, a matriz de pontuação de posição específica para o aminoácido j na posição i é dada pelo *score* do aminoácido j na posição i e pela taxa de frequência P_j de cada aminoácido j estar presente nas proteínas homólogas, conforme apresentado na Equação 2.2.

$$score_{i,j} = \frac{F_{i,j}}{N} \tag{2.1}$$

$$M_{i,j} = \log(\frac{score_{i,j}}{P_j}) \tag{2.2}$$

2.2 Conceitos Computacionais

Nesta seção, descrevemos os conceitos computacionais que servem como base para o entendimento desta pesquisa.

2.2.1 Algoritmos de Classificação

Nesta subseção, apresentamos os algoritmos de classificação relacionados ao tema sob investigação.

Floresta Aleatória

Árvores de decisão são classificadores baseados em tomadas de decisão. A principal ideia deste tipo de classificador é que a decisão final é dividida em decisões menores mais simples, auxiliando o classificador a atingir a resposta esperada [71].

A árvore de decisão é composta por nós, que são responsáveis pelas tomadas de decisão do tipo "se e senão". Os nós do tipo folha realizam a predição das classes disponíveis para a classificação.

O algoritmo de classificação floresta aleatória (do inglês *Random Forest*, RF) consiste em um conjunto de árvores de decisão que votam nas classes que cada uma das árvores acredita ser a correta, sendo que, ao final, a classe com mais votos é escolhida [9].

Cada uma das árvores da floresta aleatória possui uma certa quantidade de amostras para treinamento, sendo que, para cada amostra, é utilizada uma certa quantidade de características, selecionadas de forma aleatória. O algoritmo mais utilizado para o treinamento das árvores é o *bagging* [8], onde são selecionadas amostras com reposição, ou seja, podem existir amostras duplicadas em uma mesma árvore.

Redes Neurais

O cérebro dos seres vivos é composto por unidades básicas chamadas neurônios. O neurônio é composto por dendritos, responsáveis pela recepção de informações de outros neurô-

nios, o corpo celular e os axônios, responsáveis pelo envio de informações para os próximos neurônios. A conexão de neurônios forma uma rede neural biológica.

Baseado no neurônio biológico, Rosemblatt [69] propôs o perceptron, um neurônio artificial. Perceptrons possuem uma composição similar com os neurônios biológicos, com dendritos, que recebem a entrada de dados, corpo celular, que calcula e verifica a ativação do neurônio, e axônios, que apresentam a saída de dados.

Redes neurais são compostas por perceptrons empilhados em camadas, conectando cada neurônio de cada camada com os neurônios da camada seguinte. A primeira camada, chamada de camada de entrada, recebe os valores dos dados em questão e passa a saída dos neurônios para a entrada dos neurônios da próxima camada. As camadas intermediárias da rede são chamadas de camadas ocultas. A última parte da rede é chamada de camada de saída. A Figura 2.2 mostra uma rede neural com 5 neurônios na camada de entrada, 7 neurônios na camada oculta e 3 neurônios na camada de saída.

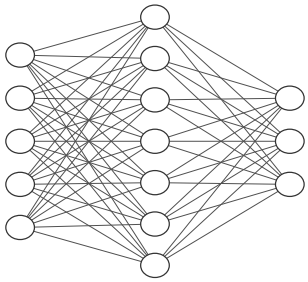


Figura 2.2: Rede neural com 3 camadas.

Durante o treinamento da rede neural, os pesos das conexões entre neurônios de diferentes camadas precisam receber atualizações para reduzir o erro da rede, avaliada por uma função de perda. A atualização é feita pelo algoritmo *backpropagation* (retropropagação), alterando os pesos da camada de saída até a camada de entrada. O método de otimização usa descida do gradiente [77].

Redes Neurais Recorrentes

Redes neurais possuem grande capacidade como classificadores e podem atingir bons resultados em muitos problemas de aprendizado de máquina. Entretanto, existem algumas limitações na configuração padrão de redes neurais, como dados em formato de sequência [50], como quadros de vídeo, sequência de textos e música.

A principal dificuldade de redes neurais em lidar com dados sequenciais e temporais ocorre devido à necessidade de separar os dados em trechos e usar a separação como janelas deslizantes. Com isso, esse tipo de rede perde informações sequenciais, o que na maioria dos casos é importante para analisar a sequência inteira.

Para resolver os problemas apresentados de redes neurais, as redes neurais recorrentes (do inglês *Recurrent Neural Networks*, RNN) foram criadas. A construção das RNN foi baseada no trabalho de Rumelhart et al. [70], usando o conceito de uma rede que pode aprender dados internos a partir de uma sequência. Para isso, é necessária uma memória interna em cada nó da rede para ter acesso às informações históricas da rede.

Na arquitetura *vanilla* da RNN, além das conexões entre camadas, existem ligações entre os neurônios da mesma camada, fazendo com que a informação da rede em um momento anterior continue naquela camada. A Figura 2.3 mostra um neurônio da arquitetura *vanilla*, que recebe informações de um momento anterior e do momento atual da sequência, processa e entrega essa informação para um neurônio da mesma camada e para a próxima camada.

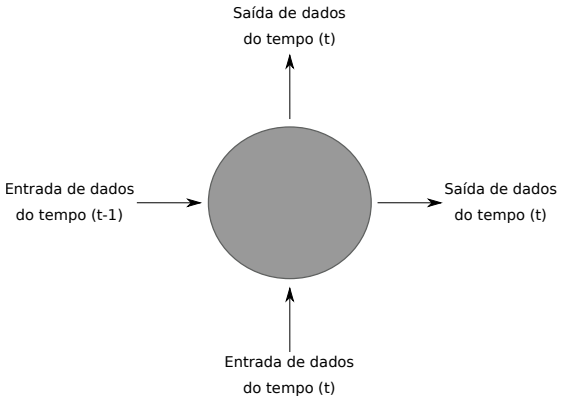


Figura 2.3: Neurônio da arquitetura *vanilla* da RNN.

A principal desvantagem da arquitetura *vanilla* é o problema de desaparecimento ou explosão de gradiente, que ocorre quando o gradiente da rede diminui consideravelmente (problema de desaparecimento) ou aumenta exponencialmente (problema de explosão), principalmente nas primeiras camadas da rede. Com isso, muitos módulos de memórias foram propostos para substituir os neurônios da arquitetura *vanilla* e cada um deles possui diferentes tipos de mecanismos para lidar com o problema de gradiente.

Os dois principais módulos de memória são LSTM (*Long Short-Term Memory*) [32] e GRU (*Gated Recurrent Unit*) [15]. O módulo GRU pode alcançar resultados similares comparado com o módulo LSTM, porém usando menos parâmetros [48].

Em alguns casos, a análise sequencial dos dados precisa de informações passadas e futuras. Portanto, redes bidirecionais recorrentes podem ser utilizadas para esse tipo de tarefa [72], como na predição de estruturas secundárias, em que é necessário analisar aminoácidos predecessores e sucessores ao aminoácido analisado [29]. A Figura 2.4 apresenta o funcionamento de uma camada deste tipo de rede recorrente, onde as informações futuras vêm dos aminoácidos sucessores e as informações passadas passadas vêm dos aminoácidos predecessores do aminoácido analisado, com a concatenação das duas características para a sequência do fluxo da rede.

Redes Neurais Convolucionais

Redes neurais convolucionais (do inglês *Convolutional Neural Networks*, CNN) é um tipo de rede neural inspirada no córtex visual dos animais. Esse tipo de rede neural é invariante à translação, escala e distorção [46]. Nas primeiras camadas, a CNN é capaz de obter características locais e em camadas mais profundas, características globais.

As CNN são compostas principalmente por três diferentes tipos de camadas. Camadas convolucionais são responsáveis por aplicar diferentes tipos de filtros nas imagens. Utilizar

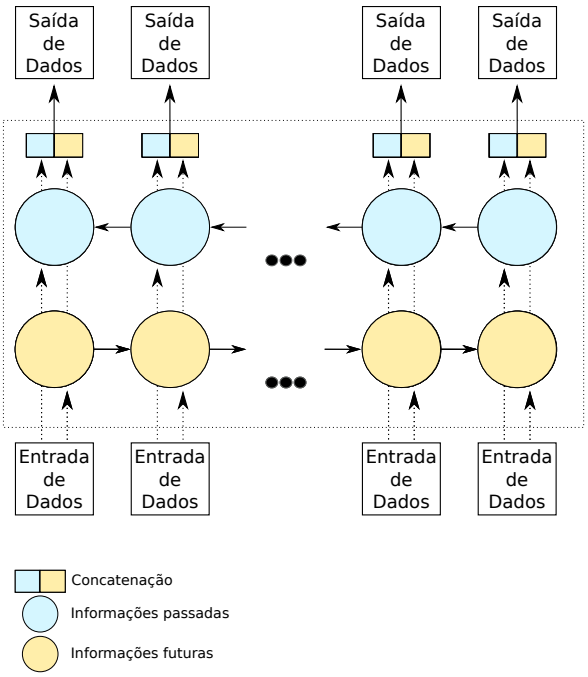


Figura 2.4: Camada bidirecional recorrente.

apenas camadas convolucionais implica um alto custo computacional, portanto, camadas de *pooling* são usadas para reduzir as dimensões das imagens. Ao final da rede, camadas totalmente conectadas com neurônios do tipo perceptron são utilizadas para fazer as predições.

Desde a década de 2010, as CNNs se tornaram classificadores populares de imagens após os resultados atingidos pela AlexNet [43] na base de dados ImageNet [18]. Algumas arquiteturas que recebem atenção na literatura são ResNet [30], GoogLeNet [79], EfficientNet [80] e Inception-v4 e Inception-ResNet-v2 [78].

A arquitetura Inception-v4 é baseada em blocos *inceptions*, que consiste em um bloco com convoluções e *poolings* em paralelo que são concatenados ao final do bloco. Este tipo de arquitetura é capaz de concatenar informações de diversas janelas de diversos tamanhos em relação a cada pixel da imagem.

As CNNs mostraram que podem atingir bons resultados em outras áreas, como na predição de estruturas secundárias [49, 51, 95].

Transformers

Mecanismos de atenção começaram a ser utilizados em aprendizado de máquina em diversas tarefas. A ideia principal destes mecanismos é fazer com que a rede aprenda o que ela deve focar e o que não deve focar para realizar alguma atividade.

Os *Transformers* [83] são arquiteturas baseadas em redes neurais do tipo *encoder-decoder*, capazes de lidar com dados textuais e sequenciais. Estas arquiteturas utilizam mecanismos de auto-atenção (do inglês *self-attention*), em que os mecanismos de atenção aprendem quais *tokens* de uma frase estes devem prestar atenção. Esta técnica se tornou o estado da arte em diversos problemas de processamento de linguagem natural, atingindo

resultados melhores em relação aos mecanismos básicos de atenção [4, 55].

Alguns métodos disponíveis realizam as análises de processamento de linguagem natural apenas em um sentido, ou seja, da esquerda para a direita ou da direita para a esquerda [63, 65]. Com isso, grande parte das informações pode ser perdida.

O BERT (*Bidirectional Encoder Representations from Transformers*) [19] é um modelo de representação que utiliza *Transformers* e que utiliza camadas bidirecionais, tornando-se o estado da arte em diversos problemas de processamento de linguagem natural.

Para realizar o treinamento e avaliação, etapas de pré-processamento dos dados são necessárias, como manter as sentenças em um mesmo tamanho adicionando *padding* e caracteres para indicar o início. Durante o treinamento, uma certa quantidade de palavras é mascarada para treinar a representação contextual do modelo. Além disso, o modelo foi treinado com a predição da próxima sentença dada uma sentença inicial.

O modelo treinado do BERT pode ser utilizado como ponto inicial para diversas tarefas. Além disso, o treinamento fino para determinadas tarefas é possível, transformando o BERT em um modelo muito utilizado na literatura. Outras arquiteturas inspiradas no BERT, como o RoBERTa (*Robustly Optimized BERT Pretraining Approach*) [52], atingiram e ultrapassaram os resultados do BERT em diversas tarefas.

2.2.2 BLAST

O BLAST (*Basic Local Alignment Search Tool*) [3] é utilizado para buscar proteínas com sequências de aminoácidos similares através da comparação dos melhores alinhamentos locais. Para realizar os alinhamentos, a ferramenta precisa de uma proteína como consulta e uma base de dados para busca, contendo proteínas que serão utilizadas para a verificação dos alinhamentos.

Durante a geração dos alinhamentos, podem ocorrer três casos de alinhamentos entre cada um dos aminoácidos da proteína da consulta com cada um dos aminoácidos das proteínas da base de dados. No primeiro caso, chamado de *match*, ocorre o alinhamento perfeito entre os dois aminoácidos. No segundo caso, chamado de *mismatch*, ocorre o alinhamento entre os dois aminoácidos distintos. No terceiro caso, chamado de *gap*, ocorre o alinhamento de um aminoácido com um trecho vazio. A Figura 2.5 mostra exemplos de *match*, *mismatch* e *gap*.

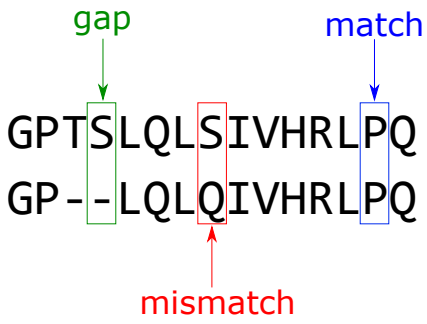


Figura 2.5: Exemplo de *match*, *mismatch* e *gap*.

Os pesos dos trechos alinhados são dados por uma matriz de substituição. A matriz

mais utilizada é a BLOSUM62 [31], que dá pesos para *matches* e *mismatches*, enquanto o peso dos *gaps* é dado por valores fixos. Ao final, o alinhamento possui valores de *bit score*, que corresponde à pontuação do trecho alinhado, e *E-value*, que corresponde à quantidade de alinhamentos esperada com alta similaridade. Quanto maior o *bit score* e menor o *E-value*, melhor é o alinhamento.

Para a predição de estruturas secundárias de proteínas, o BLAST pode ser utilizado para buscar trechos similares de proteínas e apontar que aqueles trechos possuem as mesmas estruturas secundárias. A principal desvantagem dessa abordagem vem da necessidade de existirem proteínas similares na consulta e na base de dados.

2.2.3 Algoritmos de Otimização

Nesta subseção, apresentamos os algoritmos de otimização relacionados com o tópico sob investigação.

Algoritmo Genético

O algoritmo genético (do inglês *Genetic Algorithm*, GA) é um algoritmo de otimização baseado na teoria da evolução [33]. Na teoria da evolução, os indivíduos mais adaptados ao ambiente carregam o seu material genético para as próximas gerações através de cruzamentos de códigos genéticos. Além disso, mutações podem ocorrer, alterando trechos do material genético passado para as novas gerações.

Neste algoritmo, os processos biológicos de evolução são aplicados em uma população inicial. A partir dos primeiros indivíduos, os mais adaptados ao ambiente, que neste caso são avaliados por uma função, são selecionados para serem os pais da próxima geração, gerando novos indivíduos através de cruzamentos e mutações. O Algoritmo 1 descreve o processo genérico do algoritmo genético.

Algoritmo 1: Algoritmo Genético.

```
Entrada: População com n indivíduos
Saída: Melhor indivíduo
início
    Defina o máximo de iterações do algoritmo
    Inicialize t
    enquanto t < máximo de iterações faça
        para cada Indivíduo faça
            Calcule o valor de cada indivíduo em relação a uma função de
             qualidade
        fim
        Selecione os melhores indivíduos para serem os pais da próxima geração
        Gere novos indivíduos por meio de cruzamentos entre os pais
        Aplique mutações aleatórias nos novos indivíduos
        Atualize t
    fim
    retorna melhor indivíduo
fim
```

Busca Cuco

A busca cuco (do inglês *Cuckoo Search*, CS) é um algoritmo de otimização baseado nas estratégias de reprodução dos pássaros cucos [91]. Os pássaros cucos colocam seus ovos nos ninhos de outros pássaros em uma relação parasita. Os ovos podem ser identificados pelo dono do ninho, podendo jogar os ovos dos pássaros cuco fora ou abandonar o ninho com os ovos invasores. Entretanto, os filhotes de cucos aprendem a imitar o som dos filhotes do pássaro dono do ninho para sobreviver.

O algoritmo de otimização CS usa as seguintes regras:

- Cada pássaro cuco possui um ovo e coloca este ovo em um ninho aleatório;
- Os melhores ninhos continuam para as próximas gerações;
- O número de ninhos é fixo;
- O dono do ninho pode descobrir o ovo do cuco com uma certa taxa de probabilidade.

No algoritmo, cada ovo representa uma solução e as melhores soluções substituem as piores soluções. A atualização dos valores de cada ovo é feito pelo método voos de Lévy. O Algoritmo 2 apresenta o algoritmo CS.

Algoritmo 2: Busca Cuco.

```
Entrada: População com n indivíduos
Saída: Melhor indivíduo
início
    Crie n ninhos
    Designe cada indivíduo para um ninho
    Defina o máximo de iterações do algoritmo
    Inicialize t
    enquanto t < máximo de iterações faça
        para cada Indivíduo faça
            Gere o correspondente cuco usando voos de Lévy
            Calcule o valor de cada indivíduo em relação a uma função de
             qualidade
            Escolha um ninho aleatório
            se cuco possui melhor valor que o cuco do ninho então
                Descarte o pior cuco e faça uma cópia do melhor cuco no ninho
            fim
        fim
        Abandone os piores ninhos (redefina o indivíduo do ninho)
        Ordene pelos melhores indivíduos
        Atualize t
    fim
    retorna melhor indivíduo
fim
```

Otimização por Enxame de Partículas

O algoritmo otimização por enxame de partículas (do inglês *Particle Swarm Optimization*, PSO) é um algoritmo baseado no comportamento social em grupos de animais em movimento [40]. Assim como um bando de pássaros e um cardume de peixes, movimento é importante para decidir qual nova posição cada um dos indivíduos do grupo irá.

Neste algoritmo de otimização, cada indivíduo tem informação sobre o melhor valor atingido individualmente, o melhor resultado do grupo e a atual direção do movimento. A atualização de posição leva em conta essas informações de modo ponderado, ou seja, algumas informações são mais relevantes para a atualização da posição. O Algoritmo 3 apresenta o algoritmo PSO.

Algoritmo 3: Otimização por Enxame de Partículas.

```
Entrada: População com n indivíduos
Saída: Melhor indivíduo
início
    para cada Indivíduo faça
        Defina o valor de movimento igual a 0
    fim
    Defina o máximo de iterações do algoritmo
    Inicialize t enquanto t < máximo de iterações faça
        para cada Indivíduo faça
            Calcule o valor de cada indivíduo em relação a uma função de
             qualidade
            Atualize a melhor posição individual
            se melhor posição individual > melhor posição global então
                Atualize a melhor posição global
            fim
            Atualize o movimento
            Atualize o indivíduo usando movimento, melhor posição individual e
             melhor posição global
        fim
        Atualize t
    fim
    retorna melhor indivíduo
fim
```

Capítulo 3

Trabalhos Relacionados

Neste capítulo, apresentamos diversos trabalhos disponíveis na literatura para a predição de estruturas secundárias, que podem ser divididos em três grandes fases.

3.1 Primeira Fase

A primeira fase deu início na década de 1970, muito por conta do trabalho de Chou e Fasman [16], que desenvolveram um método para a predição de hélices e folhas de estruturas secundárias baseado em regras manualmente encontradas que utilizavam informações de vizinhos próximos, médias e distâncias curtas e informações de início e final de áreas de hélices e folhas.

A partir do método criado por Chou e Fasman, Garnier *et al.* [23] propuseram um método estatístico baseado em uma janela de tamanho 17, ou seja, 8 aminoácidos antes e 8 aminoácidos seguintes do aminoácido analisado, de forma quantitativa, diferentemente do modelo proposto por Chou e Fasman, que utilizava regras qualitativas.

No mesmo período, alguns trabalhos começaram a utilizar proteínas homólogas para predizer estruturas secundárias. Levin *et al.* [47] utilizaram a matriz de características de Kabsch e Sander [38] para classificar estruturas secundárias de proteínas homólogas encontradas a partir de uma base de dados maior. Nishikawa e Ooi [59] propuseram um método para classificar proteínas homólogas utilizando a matriz de características de Kubota *et al.* [44].

A principal característica da primeira fase de trabalhos de classificação de estruturas secundárias é a utilização de regras criadas e medidas estatísticas para realizar a classificação, assim como os primeiros estudos da utilização de métodos baseados em modelos para a predição de estruturas secundárias. Devido ao pouco poder computacional disponível na época, os métodos foram testados com poucas proteínas, o que provavelmente fez com que os resultados ficassem piores em relação aos métodos das fases seguintes. A Tabela 3.1 sumariza alguns métodos da primeira fase.

Trabalho	Ano	Método
Chou e Fasman [16]	1974	Regras criadas manualmente utilizando vizinhos próximos, médias e informações de regiões
Garnier *et al.* [23]	1978	Regras quantitativas utilizando janelas de vizinhos próximos
Levin *et al.* [47]	1986	Verificação de características da matriz de Kabsch e Sander [38] para analisar proteínas homólogas
Nishikawa e Ooi [59]	1986	Verificação de características da matriz de Kubota *et al.* [44] para analisar proteínas homólogas

Tabela 3.1: Trabalhos de predição de estruturas secundárias da primeira fase.

3.2 Segunda Fase

Com o desenvolvimento do poder computacional a partir do final da década de 1980, além da criação e evolução de bases de dados de proteínas, como o desafio bianual CASP [58], PISCES [84], PDB [7] e CB513 [17], diversos trabalhos começaram a utilizar algoritmos classificadores de aprendizado de máquina, dando início à segunda fase da predição de estruturas secundárias de proteínas.

Dentre os métodos da segunda fase, Holley e Karplus [34] propuseram o primeiro método utilizando redes neurais para a classificação de estruturas secundárias. Devido ao pouco poder computacional, os autores conseguiram criar uma rede com apenas três camadas (uma camada de entrada, uma camada oculta e uma camada de saída), além de treinar e testar o método com apenas 62 proteínas do PDB. O método aplica janela deslizante de tamanho igual a 17.

Outros trabalhos na literatura também utilizaram redes neurais, como Kneller *et al.* [42], que empregam uma rede sem camadas ocultas, com janela de tamanho 13 e realizaram o treino e teste em 105 proteínas do PDB, e Jones [37], que usaram uma rede com uma camada oculta, janela de tamanho 15 e 187 proteínas da base CASP3 para treinamento e teste.

Ainda com redes neurais, porém variações da rede padrão, existem outros trabalhos que pertencem a segunda fase, como Zhang e Jing [93], que construíram uma rede neural com função radial, e Ceroni *et al.* [11], que criaram redes neurais recorrentes com grafos para realizar a representação das interações entre os aminoácidos. Ceroni *et al.* [11] foi um dos primeiros trabalhos a relatar a importância de interações de longa distância na predição de estruturas secundárias.

Além de redes neurais, outros classificadores foram utilizados para fazer a predição de estruturas secundárias. Dentre eles, a máquina de vetores de suporte (do inglês *Support Vector Machines*, SVM) foi utilizada tanto como classificador um-contra-o-resto [35] quanto combinada com técnicas de agrupamento devido à alta complexidade do SVM [94].

Alguns outros métodos livres de modelo utilizados na literatura são baseados em regras de associação [97] e modelos com votação de classificadores [85, 90]. Vale destacar o

método utilizado por Wang *et al.* [85], que realiza pós-processamento na classe "H" da classificação Q3, deixando o método mais eficiente em relação aos classificadores redes neurais e SVM.

Em relação aos métodos que são baseados em modelos, o BLAST e PSI-BLAST foram criados na década de 1990 [2, 3], o que permitiu o uso destas ferramentas para a predição de estruturas secundárias, como foi utilizado em Przybylski e Rost [64], que avaliaram PSI-BLAST e BLAST para predizer estruturas secundárias a partir de alinhamentos de proteínas do conjunto de teste (cerca de 1.600 proteínas do PDB) com uma base de dados construída, com a fusão das proteínas do PDB, SWISS-PROT e TrEMBL.

Os métodos da segunda fase apresentaram classificadores com métodos de janela deslizante para realizar a classificação de estruturas secundárias das proteínas. Entre os métodos, não houve um consenso entre o tamanho ótimo da janela analisada, já que cada trabalho encontrou um tamanho de janela diferente. Com isso, a principal desvantagem desses métodos é a limitação de análise pela janela estabelecida, pois a estrutura secundária que um aminoácido forma pode depender de aminoácidos distantes. A Tabela 3.2 apresenta alguns métodos da segunda fase de predição de trabalhos sobre estruturas secundárias.

Trabalho	**Ano**	**Método**	**Bases Utilizadas**
Holley e Karplus [34]	1989	Redes neurais	62 proteínas do PDB
Kneller *et al.* [42]	1990	Redes neurais	105 proteínas do PDB
Jones [37]	1999	Redes neurais	187 proteínas do CASP3
Hua e Sun [35]	2001	SVM	RS126 e CB513
Przybylski e Rost [64]	2002	BLAST e PSI-BLAST	PDB, SWISS-PROT e TrEMBL
Ceroni *et al.* [11]	2005	RNN + grafos	800 proteínas do PDB
Zhong *et al.* [94]	2007	SVM + agrupamento	4.000 proteínas do PISCES
Wang *et al.* [85]	2008	Classificadores binários	RS126 e CB513
Zhang e Jing [93]	2008	Redes neurais com função radial	126 proteínas do PDB
Zhou *et al.* [97]	2010	Regras de associação	RS126 e CB513
Yang *et al.* [90]	2011	Votação piramidal	RS126, CB513 e CASP8

Tabela 3.2: Trabalhos de predição de estruturas secundárias da segunda fase.

Ainda na segunda fase, alguns trabalhos começaram a avaliar a utilização de características adicionais além da sequência de aminoácidos na predição de estruturas secundárias, como matriz de pontuação de posição específica, características biológicas dos aminoácidos e da sequência das proteínas.

A utilização destas características, principalmente da matriz de pontuação de posição específica, fizeram com que os métodos atingissem altos valores de acurácia Q3. Com isso, os trabalhos da literatura começaram a realizar a predição na classificação Q8, que é uma subclassificação da categoria Q3. Além da nova classificação, a matriz de pontuação de posição específica se tornou comum nos métodos da terceira fase, com bases de dados já disponibilizando essa característica juntamente com a sequência de aminoácidos.

3.3 Terceira Fase

A terceira fase de métodos de predição de estruturas secundárias teve início na década de 2010, com o desenvolvimento de classificadores que utilizam aprendizado profundo, como redes neurais convolucionais e recorrentes. Nesta fase, novas bases foram e continuam sendo utilizadas, como CB6133 [95] e novas bases do CASP.

As redes convolucionais possuem filtros capazes de analisar as janelas de interações entre aminoácidos próximos. Como cada camada pode possuir filtros de tamanhos variados, a escolha da janela de tamanho ótimo não se torna um problema para esse método, já que é possível variar o tamanho a cada camada. Diversos trabalhos utilizaram redes convolucionais para a predição de estruturas secundárias [20, 49, 51].

Com redes convolucionais mais profundas, informações mais globais da sequência de aminoácidos podem ser obtidas, porém as informações locais se perdem. A Figura 3.1 apresenta essa perda de informações locais das sequências de proteínas na predição de estruturas secundárias.

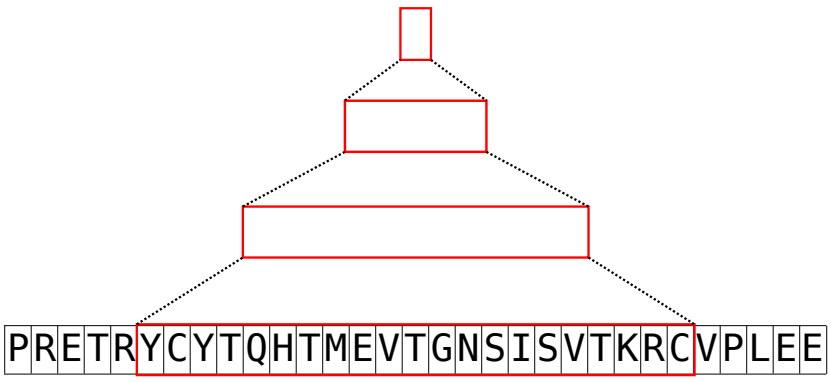


Figura 3.1: Redes neurais convolucionais com informações globais após algumas camadas de convolução.

Para manter as informações locais e concatenar com as informações globais, diversos métodos foram propostos, como redes convolucionais com blocos *inception* [22, 66, 82], redes convolucionais estocásticas generativas supervisionadas [95], redes convolucionais condicionais [10, 86], redes convolucionais com conexões com camadas anteriores [96] e redes convolucionais com módulos de contexto [54].

Dentre as abordagens propostas para manter informações locais e globais, os blocos *inception* podem concatenar janelas de tamanhos diferentes em uma mesma camada,

tornando-se uma abordagem interessante para analisar e concatenar interações de tamanhos variados para um mesmo aminoácido em uma única camada de convolução. Fang *et al.* [22] foram os primeiros autores a utilizar blocos *inception* da arquitetura GoogLeNet [79] em paralelo. Uddin *et al.* [82] utilizaram os mesmos blocos da arquitetura GoogLeNet com mecanismos de atenção. Ratul *et al.* [66] criaram uma arquitetura com blocos *inception* com convoluções unidimensionais e blocos residuais para obter informações locais e globais.

Além das redes convolucionais, redes neurais bidirecionais recorrentes com módulos de memória LSTM e GRU foram utilizadas na literatura [20, 29, 36, 75, 87]. Esse tipo de rede recebe a sequência inteira como entrada e consegue analisar, de forma global, as interações entre os aminoácidos. O grande ponto negativo nas arquiteturas com RNNs é o custo computacional, já que as arquiteturas possuem grandes quantidades de neurônios.

Com os resultados obtidos por RNNs e CNNs, abordagens utilizando ambas em uma mesma arquitetura para a classificação começaram a atingir resultados melhores em relação aos métodos utilizando apenas um tipo de rede neural [25, 26, 45, 48, 81, 92], assim como métodos que fazem fusões entre redes começaram a ganhar espaço [10, 13, 20, 28, 36].

Outro aspecto importante utilizado nos métodos da terceira fase de predição de estruturas secundárias de proteínas consiste em transformar o vetor de características da sequência de aminoácidos no formato *one-hot encoding* em um vetor de característica denso, mostrando melhoras no resultado da predição de estruturas secundárias [26, 66]. Na representação *one-hot encoding*, os dados, que são tratados inicialmente como categóricos, são transformados em dados numéricos, com diversos valores iguais a 0 e apenas a classe correspondente igual a 1, tornando-se, naturalmente, um vetor esparso.

Em relação aos métodos baseados em modelos, a maioria dos preditores da terceira fase não usa esta abordagem. O principal trabalho da terceira fase que utiliza esta técnica é Magnan e Baldi [56], que criaram uma ferramenta para realizar a predição utilizando 100 redes bidirecionais e BLAST, sendo que a classificação feita pelas 100 redes é trocada pela predição de estruturas com bom alinhamento no BLAST.

A Tabela 3.3 apresenta um panorama dos métodos da terceira fase para predição de estruturas secundárias de proteínas.

Trabalho	**Ano**	**Método**	**Bases Utilizadas**
Magnan e Baldi [56]	2014	RNN + BLAST	PDB
Sønderby e Winther [75]	2014	RNN	CB6133 e CB513
Zhou e Troyanskaya [95]	2014	CNN estocásticas generativas	CB6133 e CB513
Li e Yu [48]	2016	CNN + RNN	CB6133, CB513, CASP10 e CASP11
Lin *et al.* [49]	2016	CNN + *shift and stitch*	CB6133, 4prot e CB513

Wang *et al.* [86]	2016	CNN condicional	CullPDB, CAMEO, CASP10, CASP11 e CB5133
Busia e Jaitly [10]	2017	CNN condicional	CB6133 e CB513
Hasic *et al.* [28]	2017	Redes neurais + janela deslizante	RS121 e FC699
Hattori *et al.* [29]	2017	RNN	CB6133 e CB513
Johansen *et al.* [36]	2017	RNN + campos condicionais	CB6133, CB513, CASP10, CASP11 e CASP12
Liu *et al.* [51]	2017	CNN	CullPDB, CB513, 25PDB, CASP9, CASP10 e CASP11
Wang *et al.* [87]	2017	RNN + *autoencoder*	CullPDB e CB513
Drori *et al.* [20]	2018	Fusão de redes	CB6133 e CB513
Fang *et al.* [22]	2018	*Inception*	CB6133, JPRED, CASP10, CASP11, CASP12, CB513 e algumas proteínas do PDB
Guo *et al.* [26]	2018	CNN + RNN	CB6133, CB513, CASP10 e CASP11
Zhang *et al.* [92]	2018	CNN + RNN	TR12148, CB513, CASP10, CASP11 e CASP12
Zhou *et al.* [96]	2018	CNN + conexões	CB6133 e CB513
Uddin *et al.* [82]	2019	*Inception* + atenção	CB6133, CB513, CASP10 e CASP11
Guo *et al.* [25]	2019	CNN + RNN	CB6133, CB513, CASP10 e CASP11
Long e Tian [54]	2019	CNN + módulos de contexto	CullPDB, CB513, CASP12 CASP13
Torrisi *et al.* [81]	2019	CNN + RNN	PDB
Cheng *et al.* [13]	2020	Fusão CNN com RNN + RF	25pdb
Kumar *et al.* [45]	2020	CNN + RNN	CB6133, CB513, CASP10 e CASP11

Ratul *et al.* [66]	2020	*Inception* + blocos residuais	CB6133, CB513, CASP10 e CASP11

Tabela 3.3: Trabalhos de predição de estruturas secundárias da terceira fase.

Capítulo 4

Bases de Dados e Métricas de Avaliação

Neste capítulo, descrevemos as bases de dados e as métricas de avaliação utilizadas para avaliar o método proposto.

4.1 Bases de Dados

Nesta seção, apresentamos as bases de dados usadas no desenvolvimento do método proposto.

4.1.1 CB6133

A base de dados CB6133 [1] é um conjunto de 6.133 proteínas com sequências de tamanho de até 700 aminoácidos e que possuem menos que 30% de similaridade entre si [95]. A base foi coletada a partir do PISCES CullPDB [84].

Cada proteína da base possui um vetor de características de tamanho igual a 700 (proteínas com menos que 700 aminoácidos possuem um preenchimento). Para cada um dos aminoácidos das sequências que formam a proteína, existem 57 características, sendo 22 delas sobre a sequência de aminoácidos no formato *one-hot encoding* (o aminoácido "X" é diferente do aminoácido "A" e existe um indicador de preenchimento), 9 sobre a estrutura secundária da classificação Q8 (existe um indicador de preenchimento), 2 sobre C e N terminal, 2 sobre acessibilidade do solvente absoluto e relativo, e 22 sobre a matriz de pontuação de posição específica (o aminoácido "X" é diferente do aminoácido "A" e existe um indicador de preenchimento). O preenchimento possui todos os valores iguais a 0.

Para obter as estruturas secundárias das proteínas, a ferramenta DSSP [38] foi utilizada nos arquivos de cada proteína no PDB. A ferramenta DSSP recebe como entrada o arquivo no formato PDB e retorna as estruturas secundárias de cada um dos aminoácidos. A matriz de pontuação de posição específica foi obtida usando PSI-BLAST contra UniRef90 [76] com limiar igual a 0,001 e 3 iterações, sendo que os resultados foram normalizados para o intervalo 0 até 1 a partir da função sigmoide [95].

[1] `https://www.princeton.edu/~jzthree/datasets/ICML2014/`

Alguns aminoácidos são mais frequentes que outros nas proteínas da base, conforme mostra a Figura 4.1. Os aminoácidos mais presentes são Alanina (“A”) e Leucina (“L”), compondo cerca de 17% dos aminoácidos das proteínas da base CB6133.

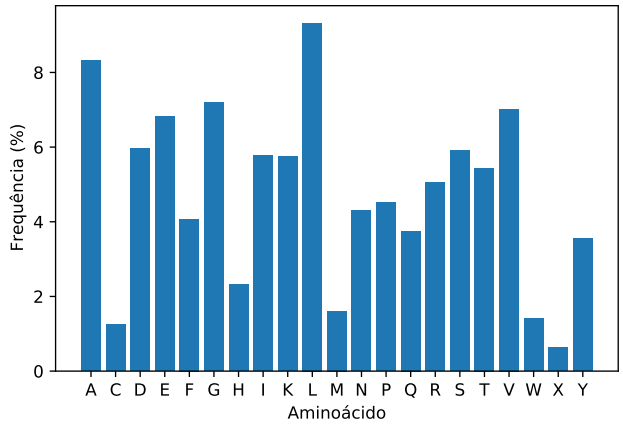


Figura 4.1: Frequência de aminoácidos na base de dados CB6133.

A frequência de estruturas secundárias na base CB6133 é apresentada na Figura 4.2. As classes mais frequentes são “H” (34,39%), “E” (21,74%), “L” (19,29%) e “T” (11,32%) e as classes menos frequentes são “I” (0,02%), “B” (1,06%), “G” (3,90%) e “S” (8,29%).

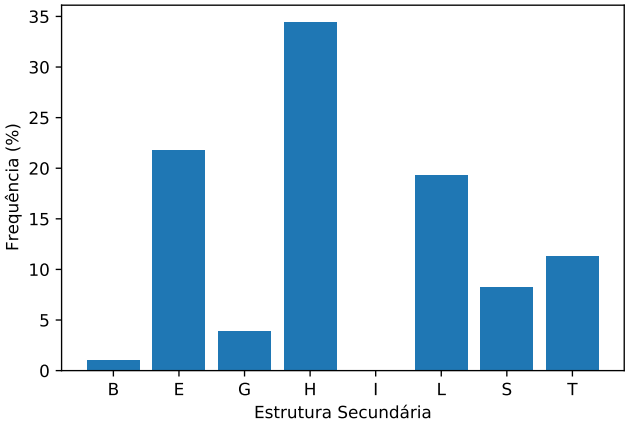


Figura 4.2: Frequência de estruturas secundárias na base de dados CB6133.

A Tabela 4.1 mostra a média, mediana e desvio padrão do tamanho das sequências de uma mesma estrutura na base CB6133. As classes “B”, “L”, “S” e “T” geralmente aparecem em sequências de até 2 estruturas consecutivas e a classe “I” aparece geralmente em sequências de 5 estruturas consecutivas. As classes “H” e “E” possuem os maiores desvios padrões, o que mostra que essas classes não seguem um padrão em relação à sequência de estruturas consecutivas.

A porcentagem de cada aminoácido formar cada estrutura secundária das proteínas da base CB6133 é apresentada na Figura 4.3. Com isso, é possível perceber que os

Classe	Média	Mediana	Desvio Padrão
B	1,02	1	0,13
E	5,52	5	2,79
G	3,38	3	0,89
H	11,61	10	6,74
I	5,39	5	0,78
L	1,84	1	1,31
S	1,52	1	0,81
T	2,05	2	0,80

Tabela 4.1: Estatísticas das sequências de estruturas de uma mesma classe na base de dados CB6133.

aminoácidos tendem a sempre formar estruturas secundárias próximas às distribuições das classes, com destaque para as classes maioritárias, como “E”, “H” e “L”.

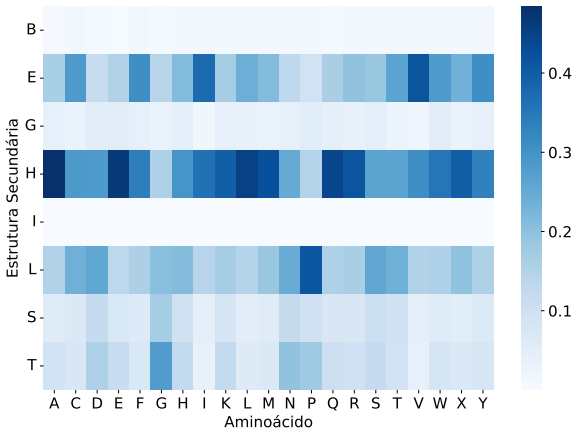


Figura 4.3: Mapa de calor da formação de estrutura secundária por aminoácido nas proteínas da base de dados CB6133.

Para fazer uma comparação justa com os trabalhos da literatura, nós utilizamos a mesma divisão da base, ou seja, 5.600 proteínas para treinamento, 256 proteínas para validação e 272 proteínas para teste. A Figura 4.4 apresenta a distribuição das classes no conjunto de treinamento, validação e teste. A distribuição é parecida nos três conjuntos principalmente devido que proteínas possuem uma distribuição muito próxima de estruturas secundárias. Nos conjuntos de treinamento e validação existem amostras da classe “I”, porém no conjunto de teste não existem dados da classe “I”.

O tamanho das proteínas nos conjuntos de treinamento, validação e teste é apresentado na Figura 4.5. Com isso, é possível perceber que o tamanho médio das proteínas nos três conjuntos é similar, com valores próximos a 200 aminácidos por sequência.

Nos experimentos envolvendo teste na base CB513, nós usamos uma versão filtrada da base CB6133 para treino e validação. A base filtrada foi criada com proteínas com menos de 25% de similaridade com as proteínas da base CB513. A base filtrada sofreu a mesma

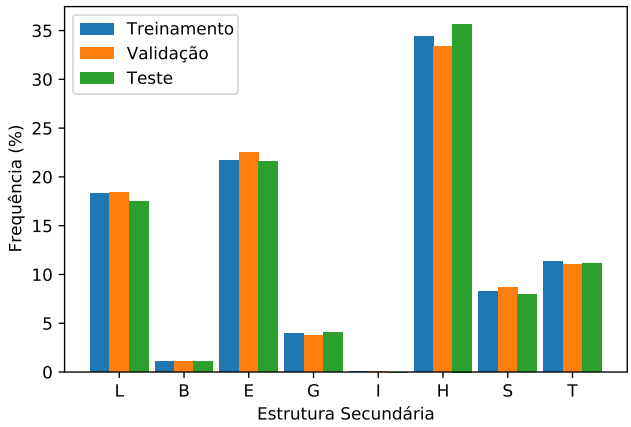


Figura 4.4: Distribuição das classes nos conjuntos de treinamento, validação e teste na base de dados CB6133.

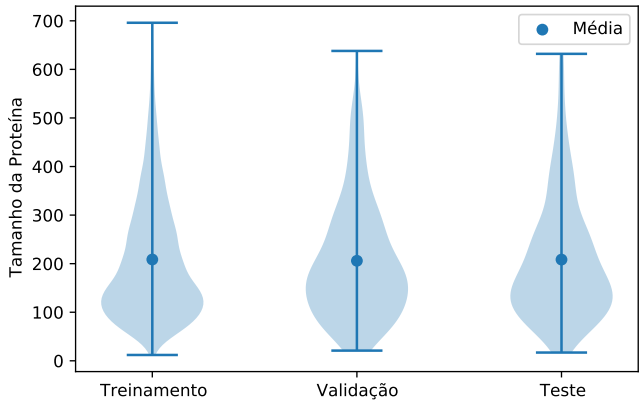


Figura 4.5: Tamanho das proteínas nos conjuntos de treinamento, validação e teste na base de dados CB6133.

divisão utilizada na literatura, ou seja, 5.278 proteínas para treinamento e 256 proteínas para validação.

4.1.2 CB513

A base de dados CB513 é um conjunto de 513 proteínas que foi originada da fusão das bases RS126 e CB396 [17]. As 9 proteínas homólogas destas duas bases foram removidas, sobrando 513 para a criação da base CB513. A base de dados CB513 é utilizada apenas para teste.

Assim como na base CB6133, a base de dados CB513 possui 57 características para cada um dos aminoácidos das proteínas. Nesta base de dados, existe uma proteína com mais de 700 aminoácidos, portanto, esta proteína foi dividida em duas, sendo que uma parte possui 700 aminoácidos e a restante se tornou uma nova proteína. As proteínas com menos que 700 aminoácidos foram preenchidas com valores iguais a 0 para atingir o

tamanho de 700.

Alguns aminoácidos são mais frequentes que outros nas proteínas da base CB513, conforme mostra a Figura 4.6. Os aminoácidos mais frequentes nas proteínas da base CB513 são Alanina (“A”) e Leucina (“L”).

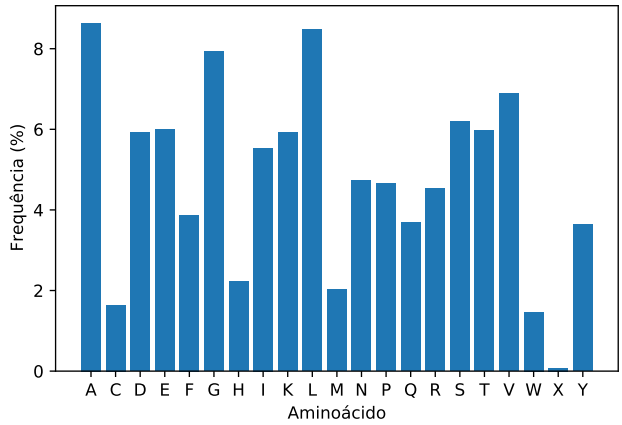


Figura 4.6: Frequência de aminoácidos na base de dados CB513.

A frequência das estruturas na base CB513 é apresentada na Figura 4.7. As classes mais frequentes são “H” (30,88%), “E” (21,25%), “L” (21,14%) e “T” (11,81%) e as classes menos frequentes são “I” (0,03%), “B” (1,39%), “G” (3,69%) e “S” (9,81%).

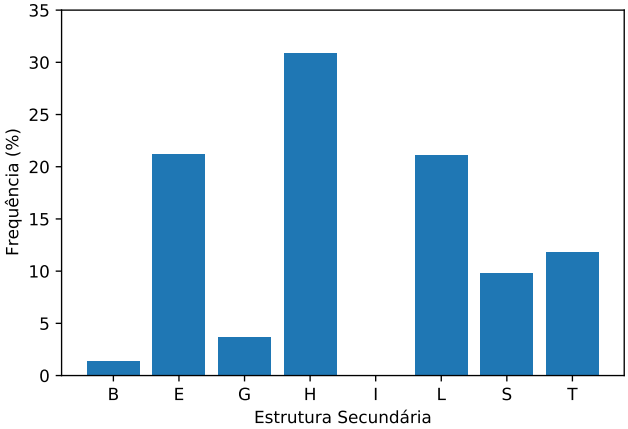


Figura 4.7: Frequência de estruturas secundárias na base de dados CB513.

A Tabela 4.2 mostra a média, mediana e desvio padrão do tamanho de sequências de uma mesma estrutura na base CB513. Os valores são próximos aos da base CB513, o que revela que existem padrões similares nas sequências de estruturas de uma mesma classe de modo geral.

A porcentagem de aminoácidos que formam cada estrutura secundária é apresentada na Figura 4.8, que mostra que os aminoácidos tendem a formar as estruturas com distribuição próxima à distribuição das classes, com a formação majoritariamente em classes mais frequentes.

Classe	Média	Mediana	Desvio Padrão
B	1,01	1	0,12
E	5,27	5	2,67
G	3,39	3	0,91
H	10,96	10	5,48
I	5,00	5	0,00
L	1,87	1	1,35
S	1,59	1	0,87
T	2,08	2	0,83

Tabela 4.2: Estatísticas das sequências de estruturas de uma mesma classe na base de dados CB513.

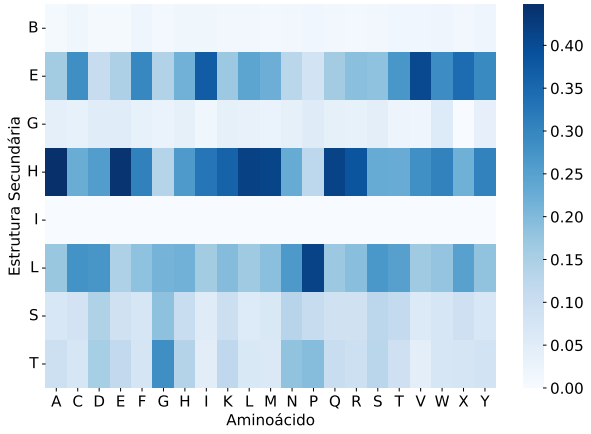


Figura 4.8: Mapa de calor da formação de estrutura secundária por aminoácido nas proteínas da base de dados CB513.

Para testar os métodos na base CB513, nós treinamos e validamos os modelos na versão filtrada da base CB6133. A Figura 4.9 apresenta a distribuição das classes no conjunto de treinamento, validação e teste.

O tamanho das proteínas nos conjuntos de treinamento, validação e teste seguem padrões próximos. A Figura 4.10 mostra o tamanho das proteínas nos três conjuntos, que possuem em média sequências de tamanho igual a 200 aminoácidos.

4.1.3 PDB

O PDB (*Protein Data Bank*) é um repositório criado em 1971 que reúne informações de estruturas 3D de proteínas, ácidos nucleicos e macromoléculas complexas. Inicialmente, o PDB foi hospedado no *Brookhaven National Laboratory* (BNL) [6].

Em Outubro de 1998, o PDB passou para a responsabilidade da *Research Collaboratory for Structural Bioinformatics* (RCSB) [7]. Em 2003, foi criada a fundação de colaboradores wwPDB (*WorldWide Protein Data Bank*)[2], que é responsável pela manutenção do

[2] `https://www.wwpdb.org`

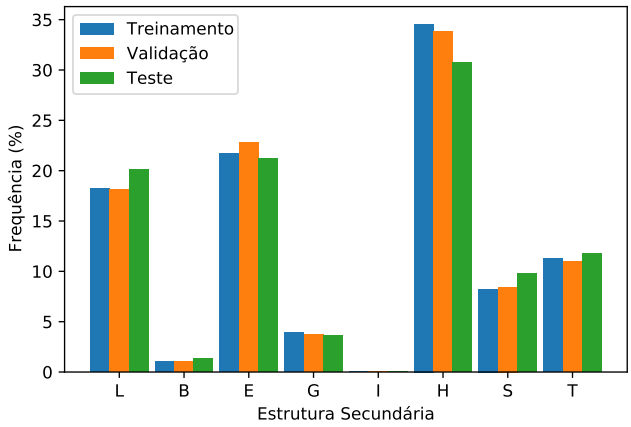


Figura 4.9: Distribuição das classes nos conjuntos de treinamento, validação e teste na base de dados CB513.

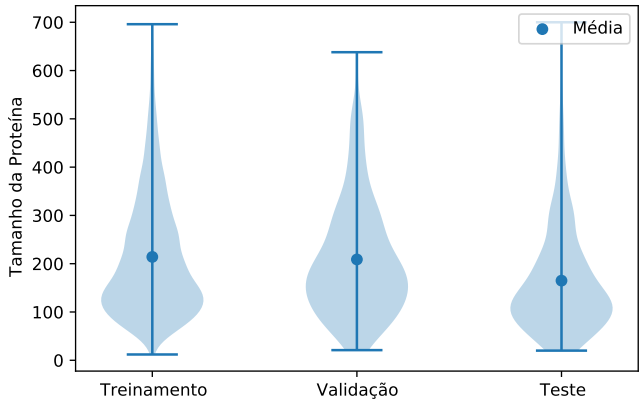


Figura 4.10: Tamanho das proteínas nos conjuntos de treinamento, validação e teste na base de dados CB513.

PDB atualmente [5].

O wwPDB possui parceiros na América (RCSB PDB)[3], na Europa (PDBe)[4], no Japão (PDBj)[5], além do BMRB (*Biological Magnetic Resonance Data Bank*)[6]. Todos os parceiros colaboram com questões de arquivamento, políticas de depósito e anotação, formatos, padrões e atualizações semanais. Cada um deles possui um repositório próprio, de modo que os dados são apresentados para diversas comunidades [6].

O início do PDB foi caracterizado por buscar interesse da comunidade em depositar as estruturas das proteínas [5]. A partir das décadas de 1980 e 1990, houve um enorme crescimento de depósitos de dados no PDB, conforme mostra a Figura 4.11. Atualmente, o PDB conta com mais de 150.000 estruturas, totalizando cerca de 1 TB de dados e continua crescendo a cada semana. Toda quarta-feira, os novos dados ficam disponíveis

[3] `https://www.rcsb.org`
[4] `https://www.ebi.ac.uk/pdbe`
[5] `https://pdbj.org`
[6] `http://www.bmrb.wisc.edu`

para *download*.

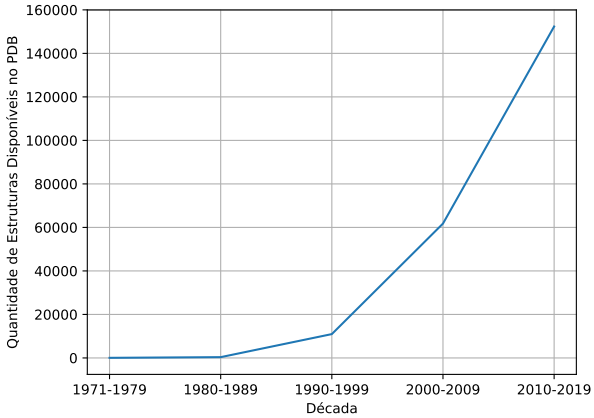


Figura 4.11: Evolução no número de proteínas depositadas no PDB ao longo do tempo.

Para a avaliação do método proposto, o conjunto de proteínas depositadas em 2018, que chamamos de PDB 2018, foi utilizado. Para obter as estruturas secundárias, todas as proteínas do conjunto foram aplicadas na ferramenta DSSP [38], que é responsável por obter as estruturas dos arquivos do PDB. Alguns aminoácidos possuem mais de uma letra como representação, portanto, tratamos o aminoácido "X" como "A", "B" como "N" e "Z" como "Q".

Além da classificação Q8, utilizamos a classificação Q3 na base PDB 2018. Para gerar as classes na categoria Q3, transformamos as classes "I", "L", "S" e "T" da classificação Q8 na classe "C" na classificação Q3, as classes "B" e "E" da classificação Q8 na classe "E" na classificação Q3 e as classes "G" e "H" da classificação Q8 na classe "H" da classificação Q3. O método utilizado corresponde ao método 1 da Tabela 2.2. Escolhemos este método, pois é o mais comum na literatura [22, 49, 87].

A Figura 4.12 apresenta a frequência de aminoácidos nas proteínas da base PDB 2018. A frequência dos aminoácidos não é balanceada, com destaque para a presença mais frequente dos aminoácidos Alanina ("A") e Leucina ("L") nas proteínas da base de dados.

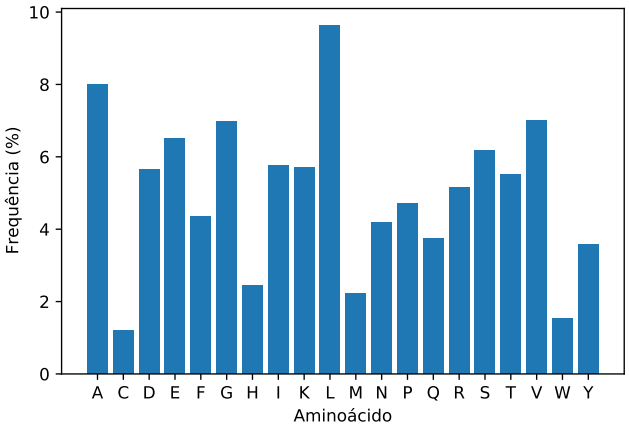


Figura 4.12: Frequência de aminoácidos na base de dados PDB 2018.

A frequência de estruturas secundárias da classificação Q3 é apresentada na Figura 4.13. A frequência das classes é 40,54% para a classe "C", 21,78% para a classe "E" e 37,68% para a classe "H".

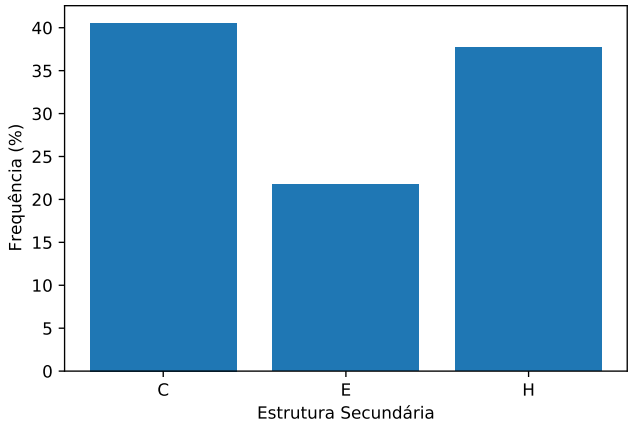


Figura 4.13: Frequência de estruturas secundárias para a classificação Q3 na base de dados PDB 2018.

A Tabela 4.3 mostra a média, mediana e desvio padrão do tamanho das sequências de uma mesma estrutura na base PDB 2018. As classes "C" e "E" aparecem na média em sequências menores em relação às sequências da estrutura "H". Como o desvio padrão é alto, todas as classes apresentam tamanhos de sequências mais dispersos.

Classe	**Média**	**Mediana**	**Desvio Padrão**
C	4,72	4	3,61
E	4,49	4	3,06
H	9,72	8	6,86

Tabela 4.3: Estatísticas das sequências de estruturas de uma mesma classe para a classificação Q3 na base de dados PDB 2018.

Cada um dos aminoácidos pode formar cada uma das estruturas secundárias da classificação Q3, conforme apresentado na Figura 4.14. O padrão de formação de estruturas secundárias segue próximo à distribuição das classes, com destaque para Glicina ("G") e Prolina ("P") para a formação da classe "C".

Para treinar, validar e testar o modelo, dividimos a base em 6.979 proteínas para treinamento, 500 proteínas para validação e 500 proteínas para teste. A distribuição das classes da classificação Q3 nos três conjuntos é retratada na Figura 4.15. Pode-se observar que a distribuição segue próxima nos três conjuntos. Na Figura 4.16 o tamanho das proteínas é apresentado nos três conjuntos. Os valores mostram que os conjuntos são bem similares, com tamanho médio das proteínas próximo a 300 aminoácidos.

A frequência de estruturas secundárias na classificação Q8 na base PDB 2018 é apresentada na Figura 4.17. As classes mais frequentes são "H" (33,82%), "E" (20,64%), "L"

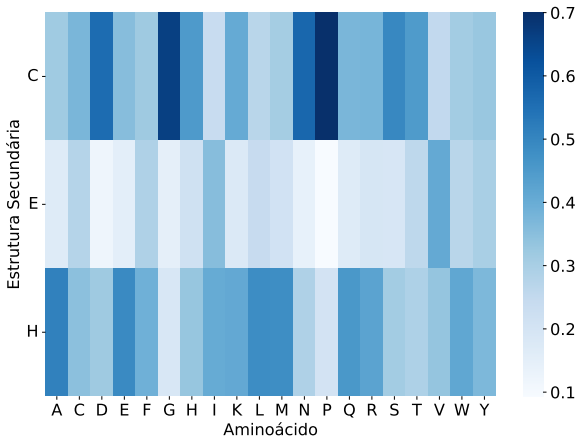


Figura 4.14: Mapa de calor da formação de estrutura secundária para a classificação Q3 por aminoácido nas proteínas da base de dados PDB 2018.

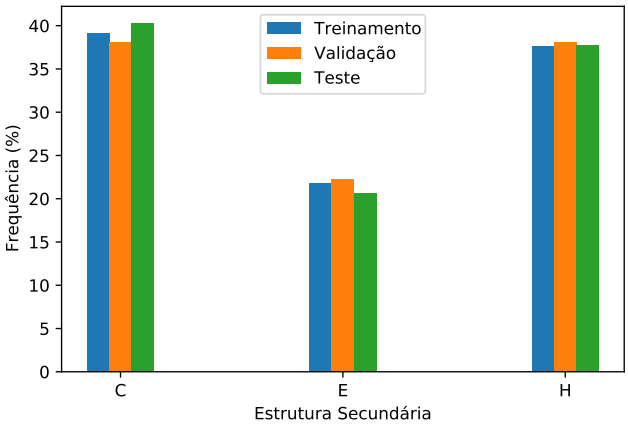


Figura 4.15: Distribuição das classes nos conjuntos de treinamento, validação e teste para a classificação Q3 na base de dados PDB 2018.

(20,04%) e "T" (11,09%) e as classes menos frequentes são "I" (0,62%), "B" (1,13%), "G" (3,88%) e "S" (8,78%).

Em relação ao tamanho de sequências de estruturas de uma mesma classe, algumas classes, como "B", "L" e "S", possuem tamanhos médios de 1 a 2 estruturas consecutivas de uma mesma classe. Outras classes, como "E" e "H", possuem alto desvio padrão, o que resulta em sequências com tamanhos variados. A Tabela 4.4 apresenta os valores de média, mediana e desvio padrão do tamanho das sequências de estruturas de uma mesma classe.

Cada aminoácido pode formar todas as estruturas secundárias com uma taxa próxima aos valores da distribuição das classes Q8. A Figura 4.18 apresenta o mapa de calor de aminoácido por estrutura secundária.

Para treinar, validar e testar o modelo, utilizamos a mesma divisão para a classificação Q8, ou seja, 6.979 proteínas para treinamento, 500 proteínas para validação e 500 proteínas para teste. A Figura 4.19 mostra a distribuição das estruturas secundárias nos

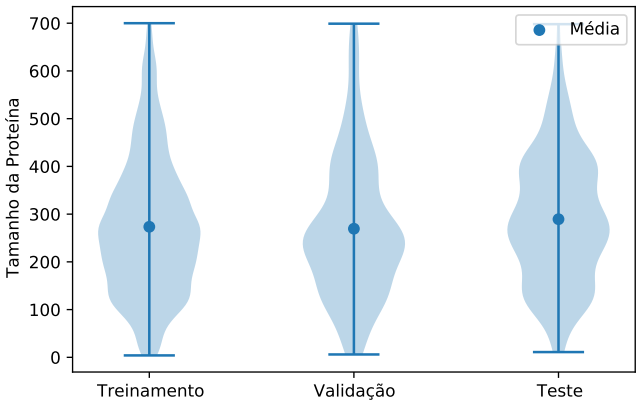


Figura 4.16: Tamanho das proteínas nos conjuntos de treinamento, validação e teste na base de dados PDB 2018.

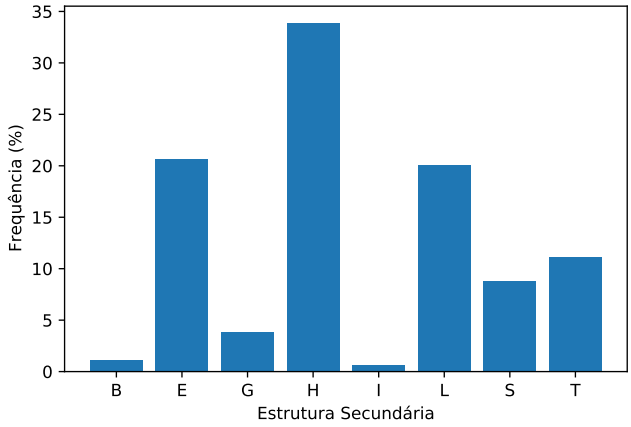


Figura 4.17: Frequência de estruturas secundárias para a classificação Q8 na base de dados PDB 2018.

Classe	Média	Mediana	Desvio Padrão
B	1,02	1	0,14
E	5,38	5	2,80
G	3,37	3	0,87
H	11,47	10	6,66
I	5,28	5	0,67
L	1,94	1	1,41
S	1,54	1	0,83
T	2,09	2	0,84

Tabela 4.4: Estatísticas das sequências de estruturas de uma mesma classe para a classificação Q8 na base de dados PDB 2018.

três conjuntos, onde possuem valores próximos.

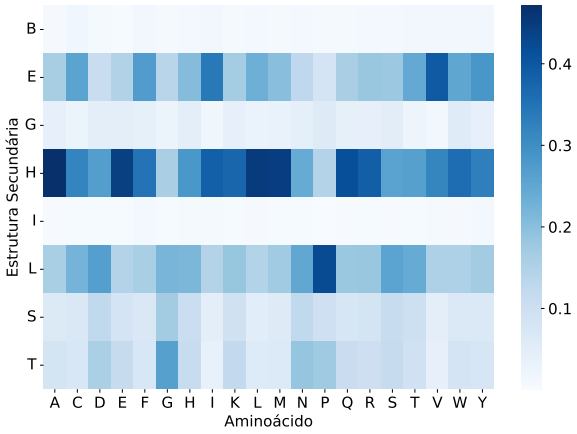


Figura 4.18: Mapa de calor da formação de estrutura secundária para a classificação Q8 por aminoácido nas proteínas da base de dados PDB 2018.

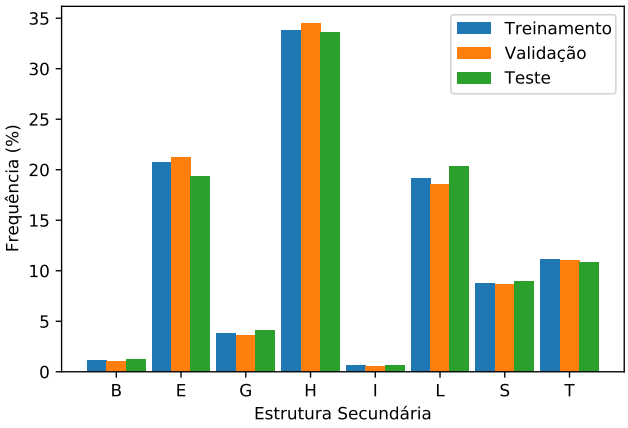


Figura 4.19: Distribuição das classes nos conjuntos de treinamento, validação e teste para a classificação Q8 na base de dados PDB 2018.

4.2 Métricas de Avaliação

Os desempenhos dos métodos propostos foram avaliados utilizando as métricas quantitativas indicadas a seguir.

A precisão é representada pela Equação 4.1, onde VP é o número de verdadeiros positivos e FP é o número de falsos positivos. A métrica da precisão verifica a capacidade de classificar como positivo um dado que realmente é positivo. Esta métrica foi utilizada para cada classe na classificação.

$$\text{Precisão} = \frac{\text{VP}}{\text{VP+FP}} \tag{4.1}$$

A taxa de revocação é dada pela Equação 4.2, onde VP é o número de verdadeiros positivos e FN é o número de falsos negativos. Essa medida verifica a capacidade do classificador de classificar corretamente dados positivos. Esta métrica foi utilizada para

cada classe na classificação.

$$\text{Revocação} = \frac{\text{VP}}{\text{VP+FN}} \tag{4.2}$$

A Equação 4.3 apresenta a Acurácia Q3. Essa medida é utilizada para avaliar as decisões corretas do método na classificação Q3.

$$\text{Acurácia Q3} = \frac{\sum_{i \in \{\text{C, E, H}\}} \text{predições corretas em } i}{\sum_{i \in \{\text{C, E, H}\}} \text{dados da classe } i} \tag{4.3}$$

A Equação 4.4 apresenta a Acurácia Q8. Essa medida é utilizada para avaliar as decisões corretas do método na classificação Q8.

$$\text{Acurácia Q8} = \frac{\sum_{i \in \{\text{B, E, G, H, I, L, S, T}\}} \text{predições corretas em } i}{\sum_{i \in \{\text{B, E, G, H, I, L, S, T}\}} \text{dados da classe } i} \tag{4.4}$$

Capítulo 5

Método para a Predição de Estruturas Secundárias

Neste capítulo, apresentamos os métodos livres de modelo e baseados em modelo que constituem nosso classificador de estruturas secundárias, assim como o método de fusão utilizado para fundir os classificadores.

5.1 Métodos Livres de Modelos

Nesta seção, descrevemos os métodos livres de modelo, divididos em redes neurais bidirecionais recorrentes, florestas aleatórias, blocos *Inception-v4*, redes *inception* recorrentes e *Transformers*.

5.1.1 Redes Neurais Bidirecionais Recorrentes

Redes neurais recorrentes são capazes de classificar sequências de dados a partir de informações obtidas em momentos anteriores. Para a predição de estruturas secundárias das proteínas, tanto aminoácidos predecessores quanto aminoácidos sucessores ao aminoácido analisado devem ser considerados, portanto, redes bidirecionais possuem grande capacidade nesta tarefa.

Os classificadores de estruturas secundárias podem ser divididos em classificadores globais, que analisam a sequência inteira da proteína, e classificadores locais, que analisam trechos da sequência. RNNs são consideradas classificadores globais, visto que conseguem obter informações a partir da sequência inteira da proteína.

Parte importante das RNNs consiste no módulo de memória utilizada. Neurônios do tipo *vanilla* possuem limitações em relação ao desaparecimento ou explosão do gradiente. Dentre as opções para superar esta limitação, dois módulos de memória são comumente utilizados, LSTM e GRU, sendo que o módulo GRU possui menos parâmetros.

Na metodologia proposta, utilizamos redes bidirecionais recorrentes com módulos de memória GRU para fazer a predição de estruturas secundárias. Avaliamos diversas configurações de redes no conjunto de validação da base CB6133, com os resultados apresentados nos Capítulos 6 e 7. As configurações avaliadas possuem diferentes quantidades de camadas, quantidade de neurônios por camadas e a utilização de camadas de *embedding*

para transformar o vetor esparso referente à sequência de aminoácidos em um vetor denso. Selecionamos os valores de 600 neurônios por camadas e entre 2 até 6 camadas como as melhores redes, além da utilização de uma camada de *embedding* para a sequência de aminoácidos. A camada de saída consiste em uma camada totalmente conectada (camada densa) com ativação *softmax*. A Figura 5.1 apresenta a configuração geral utilizada nas redes bidirecionais recorrentes.

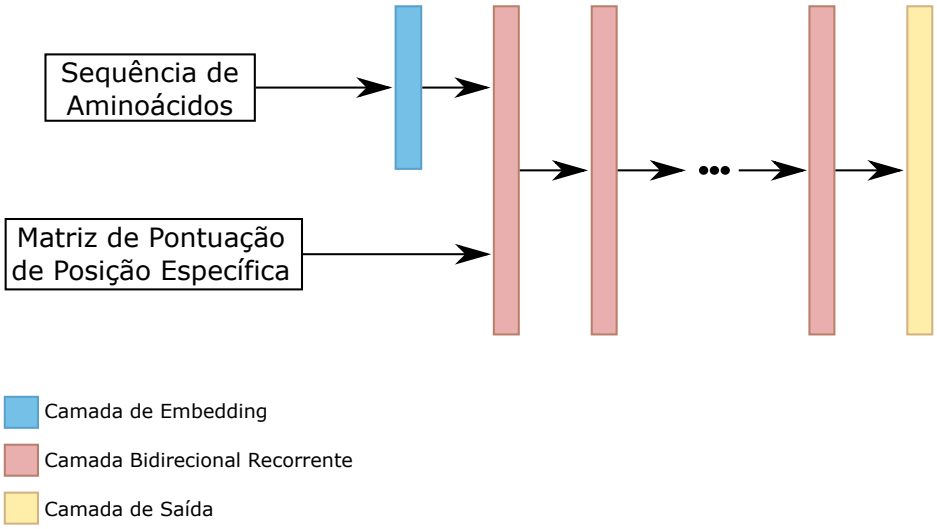


Figura 5.1: Arquitetura geral do classificador RNN.

Para cada configuração de rede, avaliamos a fusão de duas redes com a mesma arquitetura, sendo que uma rede analisa a sequência de aminoácidos no sentido encontrado nas bases de dados e a outra rede analisa no sentido inverso, ou seja, do final da sequência em direção ao começo. Ao final, as probabilidades das duas redes são concatenadas e normalizadas para que a soma total de cada predição seja igual a 1. A predição final, ou seja, depois da concatenação e normalização, é considerada a predição final daquela configuração de rede. Os resultados obtidos no conjunto de validação da base CB6133 (presentes nos Capítulos 6 e 7) mostraram que esta técnica é capaz de obter melhores resultados comparados com as redes separadas, sendo utilizada nos demais experimentos. A Figura 5.2 ilustra esta técnica para a rede com duas camadas recorrentes.

Como as redes mais profundas podem sofrer com taxas de aprendizados mais altas para encontrar mínimos locais, utilizamos o otimizador Adam [41] com taxa de aprendizado igual a 10^{-4}. Além disso, utilizamos as técnicas de parada precoce (*early stopping*) para evitar sobreajuste (*overfitting*) na rede e a diminuição da taxa de aprendizado em uma taxa igual a 10^{-1} caso não tenha melhorias após 5 épocas.

Após a predição de cinco diferentes configurações de redes bidirecionais recorrentes, com 600 neurônios por camada, com 2 até 6 camadas e com *embedding*, realizamos a fusão entre elas utilizando o método de fusão descrito na Seção 5.5. O resultado da fusão é considerado como a predição final das RNNs.

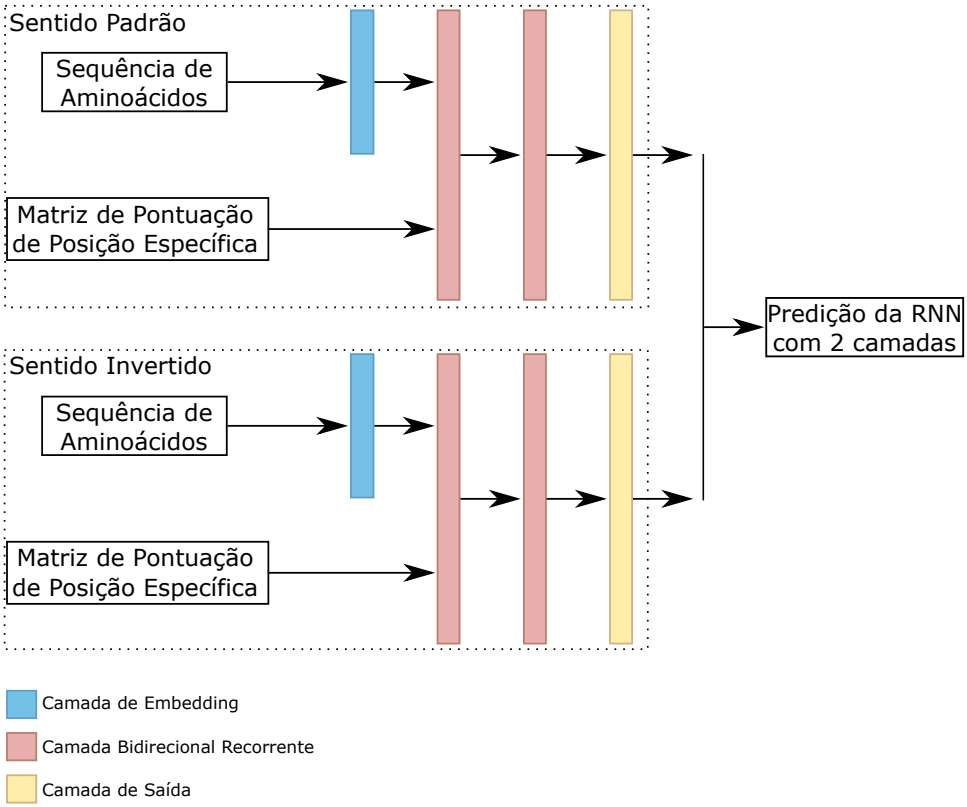


Figura 5.2: Fusão de RNNs com duas camadas bidirecionais recorrentes.

5.1.2 Florestas Aleatórias

Assim como a análise da sequência como um todo, ou seja, de modo global, a avaliação local possui importância para a predição de estruturas secundárias de proteínas, já que é capaz de encontrar padrões de interações entre aminoácidos próximos na sequência.

Dentre os métodos locais para a predição, os principais métodos abordados na literatura possuem janelas deslizantes para a análise de interações dentro de uma região pré-determinada. A principal divergência dos métodos da literatura baseados em janelas deslizantes consiste no tamanho ótimo para a janela, variando entre 13 [97] até 17 [34].

No método proposto para a predição local de estruturas secundárias de proteínas, utilizamos o algoritmo de florestas aleatórias com janelas deslizantes. Avaliamos diversos tamanhos de janelas deslizantes no conjunto de validação da base CB6133 (resultados apresentados nos Capítulos 6 e 7), variando entre 3 até 21, e os melhores resultados obtidos foram com janelas de tamanho 9 até 17.

Assim como a janela deslizante, o preenchimento ao início e término da sequências impactam na capacidade de classificação do método. Assim como nas bases originais, optamos por preencher com valores iguais a 0 nas janelas, tanto para valores à esquerda (trechos ao início da sequência), quanto para valores à direita (trechos ao final da sequência).

Como as florestas aleatórias possuem diferentes visões de uma mesma proteína (cada floresta com o tamanho de janela diferente), realizamos a fusão entre elas usando o método de fusão apresentado na Seção 5.5. O resultado final da fusão é considerado como a

predição final das florestas aleatórias. A Figura 5.3 mostra a fusão final entre as florestas aleatórias com diferentes janelas deslizantes.

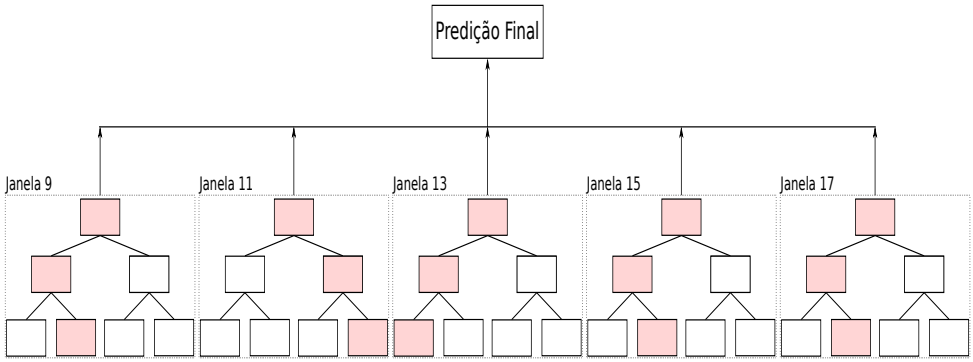


Figura 5.3: Fusão entre florestas aleatórias.

5.1.3 Blocos Inception-v4

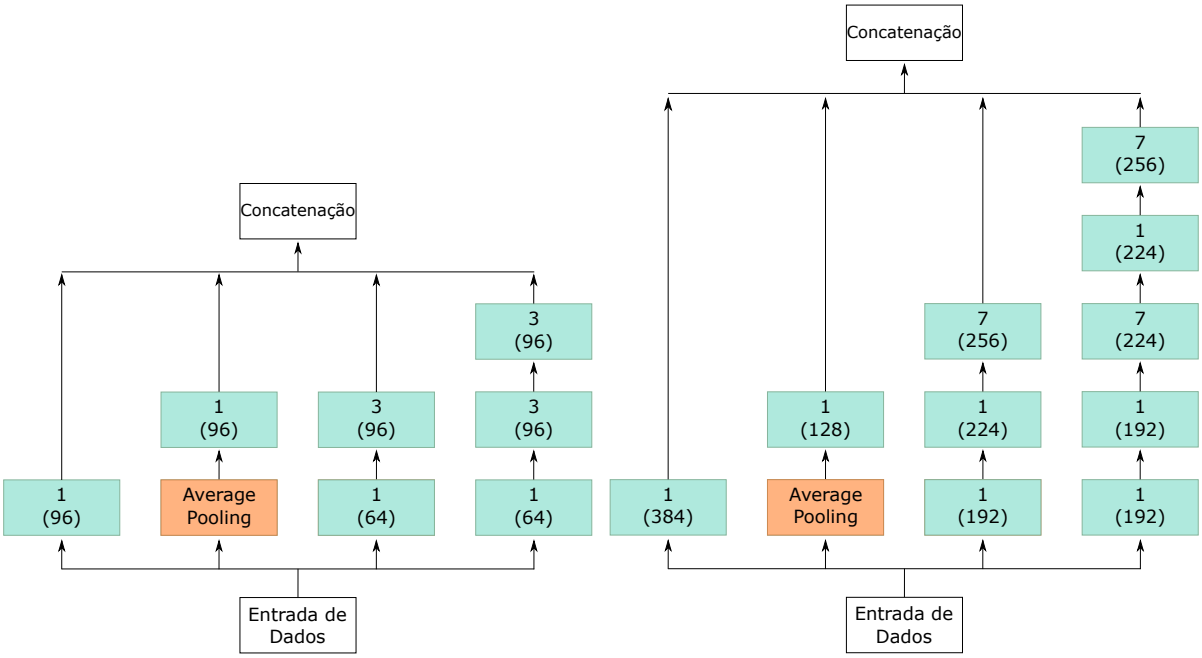
Redes neurais convolucionais podem extrair informações locais de imagens nas primeiras camadas e informações globais em camadas mais profundas. Porém, com informações mais globais, muitas informações locais se perdem.

Devido aos resultados recentes de redes convolucionais que utilizam informações locais juntamente com informações globais na predição de estruturas secundárias das proteínas [10, 22, 66], criamos um classificador baseado na arquitetura Inception-v4 [78], que chamamos de blocos Inception-v4 (BIv4). Escolhemos esta arquitetura, pois possui convoluções de diferentes tamanhos e com a concatenação das informações, dentre elas, a convolução de tamanho igual a 1, enviando informações locais para as próximas camadas da rede.

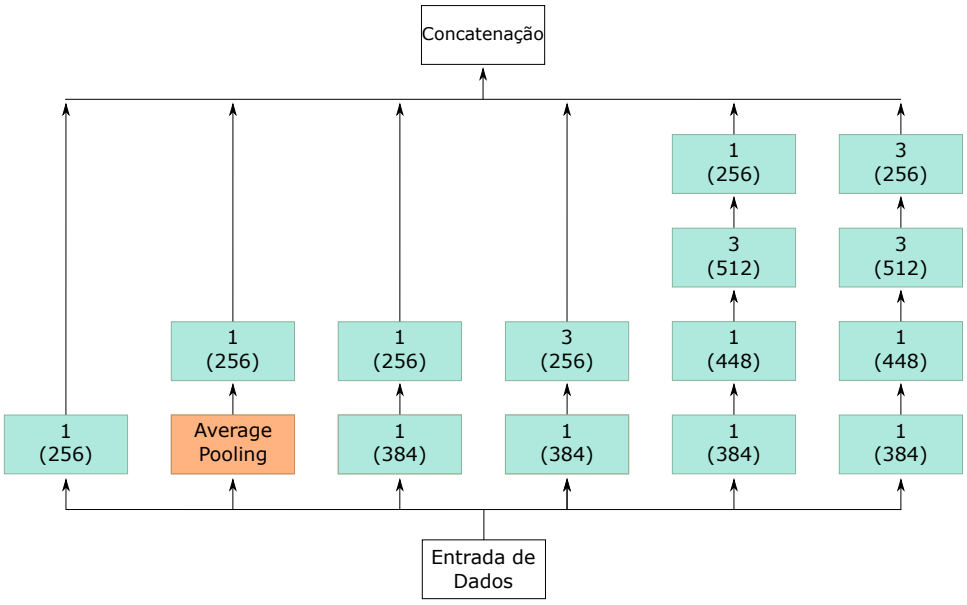
Como a arquitetura original da Inception-v4 realiza a classificação de imagens, retiramos as etapas de redução de tamanho das imagens e avaliamos apenas os três diferentes blocos que compõem a rede (bloco *inception* A, bloco *inception* B e bloco *inception* C), além de transformar as convoluções e *poolings* que constituem os blocos em unidimensionais, já que a tarefa consiste em analisar a sequência e não trechos de uma imagem. A Figura 5.4 apresenta a arquitetura das três variações de blocos *inceptions*, onde as camadas verdes indicam convoluções, com o tamanho do filtro acima do número de filtros, indicado em parênteses, e as camadas laranjas indicam *pooling*.

Realizamos experimentos no conjunto de validação da base CB6133 (conforme apresentado nos Capítulos 6 e 7) utilizando os três diferentes blocos, variando a quantidade de blocos entre 1 até 10 e adicionando ou retirando a camada de *embedding*. Selecionamos como melhores arquiteturas as redes com 3 até 7 blocos *inception* B e com *embedding* para a sequência de aminoácidos. Ao final, o classificador possui uma camada de saída totalmente conectada com ativação *softmax*.

Para o treinamento de cada rede, utilizamos a taxa de aprendizado igual a 10^{-3}, com otimizador Adam [41] e as técnicas de regularização *early stopping* e a diminuição da taxa de aprendizado em fator igual a 10^{-1} caso não tenha melhorias após 5 épocas.



(a) Bloco A. (b) Bloco B.



(c) Bloco C.

Figura 5.4: Blocos unidimensionais da arquitetura Inception-v4.

Ao término da predição das cinco arquiteturas de redes BIv4s, realizamos a fusão entre elas por meio da técnica de fusão apresentada na Seção 5.5. O resultado da fusão é considerado como a predição final dos BIv4s.

5.1.4 Redes Inception Recorrentes

Como o classificador utilizando blocos unidimensionais da arquitetura Inception-v4 consegue concatenar informações de janelas de tamanhos diferentes, ou seja, informações locais com convoluções de tamanho igual a 1 com informações com convoluções de janelas

maiores, avaliamos a possibilidade de entregar estas informações para um classificador puramente global.

O classificador Rede Inception Recorrente (RIR) possui nas camadas iniciais os blocos unidimensionais do tipo B da arquitetura Inception-v4, seguidas por camadas bidirecionais recorrentes com módulos de memória GRU.

Na primeira parte, usamos as mesmas configurações do classificador com blocos Inception-v4, ou seja, com 3 até 7 blocos do tipo B empilhados. Na segunda etapa, avaliamos diferentes configurações de camadas bidirecionais recorrentes, variando a quantidade de camadas entre 1 até 5 e a quantidade de neurônios, variando entre 100 até 500. Realizamos experimentos no conjunto de validação da base CB6133 (conforme apresentado nos Capítulos 6 e 7) e constatamos que a melhor configuração é com 3 camadas bidirecionais recorrentes e 100 neurônios por camada. Ao final, o classificador possui uma camada de saída totalmente conectada com ativação *softmax*. A Figura 5.5 apresenta o modelo geral da RIR.

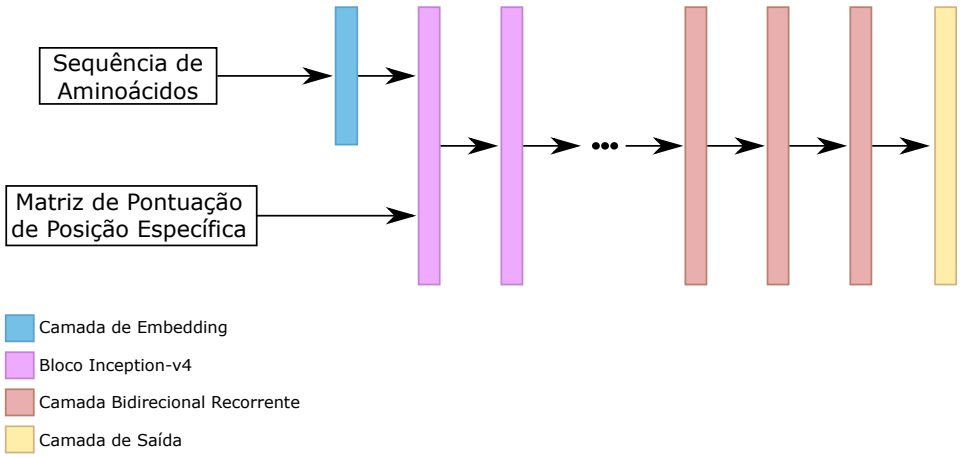


Figura 5.5: Arquitetura geral do classificador RIR.

Durante o treinamento, utilizamos a taxa de aprendizado igual a 10^{-3}, com otimizador Adam [41] e as técnicas de regularização *early stopping* e a diminuição da taxa de aprendizado em fator igual a 10^{-1} caso não tenha melhorias após 5 épocas.

Ao final da predição das cinco RIRs, realizamos a fusão entre elas utilizando a técnica de fusão descrita na Seção 5.5. O resultado da fusão é considerado como a predição final da arquitetura RIR.

5.1.5 Transformers

As arquiteturas baseadas em *Transformers* atingiram resultados promissores em diversas tarefas, como mecanismos de tradução (sequência para sequência). Os métodos baseados em *Transformers*, como o BERT [19] e o RoBERTa [52], tornaram-se o estado da arte em tarefas como análise de sentimento e reconhecimento de nome/entidade, ganhando atenção da comunidade científica.

Com os bons resultados dos métodos baseados em *Transformers* em diferentes tarefas, avaliamos três configurações distintas de classificadores baseados nesta técnica como

preditores de estruturas secundárias de proteínas.

O primeiro classificador consiste em blocos de *Transformers* para a realização de tradução de sequência de aminoácidos para estruturas secundárias, em um processo similar à tradução de um texto em uma linguagem para outra. Avaliamos diferentes quantidades de blocos *Transformers* no codificador (*encoder*) e no decodificador (*decoder*), assim como o tamanho do vetor latente e de *embedding*. A configuração utilizada possui uma camada no *encoder* e uma camada no *decoder*, com vetor latente com tamanho igual a 128 e com *embedding* com tamanho igual a 64. Treinamos o modelo com taxa de aprendizado igual a 10^{-3} por 50 épocas e com *early stopping* com valor igual a 5.

Como segundo classificador, avaliamos o BERT e o RoBERTa tratando a tarefa de predição de estruturas secundárias como análise de sentimento. Para isso, utilizamos janelas deslizantes e classificamos o aminoácido central da janela. Para o preenchimento, adicionamos um caractere especial tanto no início, no caso de aminoácidos do começo da sequência, quanto no final, no caso de aminoácidos do final da sequência, para manter os dados com tamanho igual ao da janela deslizante. Avaliamos janelas de diversos tamanhos, desde tamanho igual a 21 até 201. Em ambos os preditores, BERT e RoBERTa, realizamos o ajuste fino para a tarefa em questão, treinando os modelos com a taxa de aprendizado igual a 10^{-5} por 50 épocas com *early stopping* com valor igual a 5.

O último preditor usando *Transformers* consiste em reconhecedores de nome/entidade. Neste classificador, apresentamos a sequência completa da proteína e o preditor classifica cada aminoácido (ou *token*) da sequência em uma entidade. Avaliamos os métodos pré-treinados BERT e RoBERTa. Em ambos os preditores, utilizamos a técnica de ajuste fino, treinando o modelo com taxa de aprendizado igual a 10^{-5} por 50 épocas e com *early stopping* com valor igual a 5.

Para os três classificadores, utilizamos apenas a informação da sequência de aminoácidos e não usamos as informações de matriz de pontuação de posição específica, já que os métodos criam *tokens* para as palavras e esta tarefa seria dificultada pelos valores reais das características da matriz de pontuação de posição específica. A Figura 5.6 exemplifica como cada arquitetura foi utilizada para a predição de estruturas secundárias.

Avaliamos as três preditores no conjunto de validação da base CB6133 (conforme apresentado no Capítulo 6) e o melhor resultado foi utilizando BERT com janelas deslizantes. Considerando a melhor configuração, os melhores resultados na validação indicaram que a fusão de janelas de tamanho 101, 121, 141, 161 e 181 apresentava os melhores resultados, portanto, consideramos essa fusão como a predição final dos *Transformers*. Para realizar a fusão, usamos o método descrito na Seção 5.5.

5.2 Fusão dos Métodos Livres de Modelo

Após a realização da fusão de cada um dos cinco métodos livres de modelo separadamente, realizamos a fusão das predições finais de cada um dos métodos. Para realizar este procedimento, utilizamos o método de fusão apresentado na Seção 5.5. O resultado final da fusão dos cinco métodos livres de modelo é considerado o resultado final dos métodos livres de modelo. A Figura 5.7 ilustra a fusão dos métodos livres de modelo.

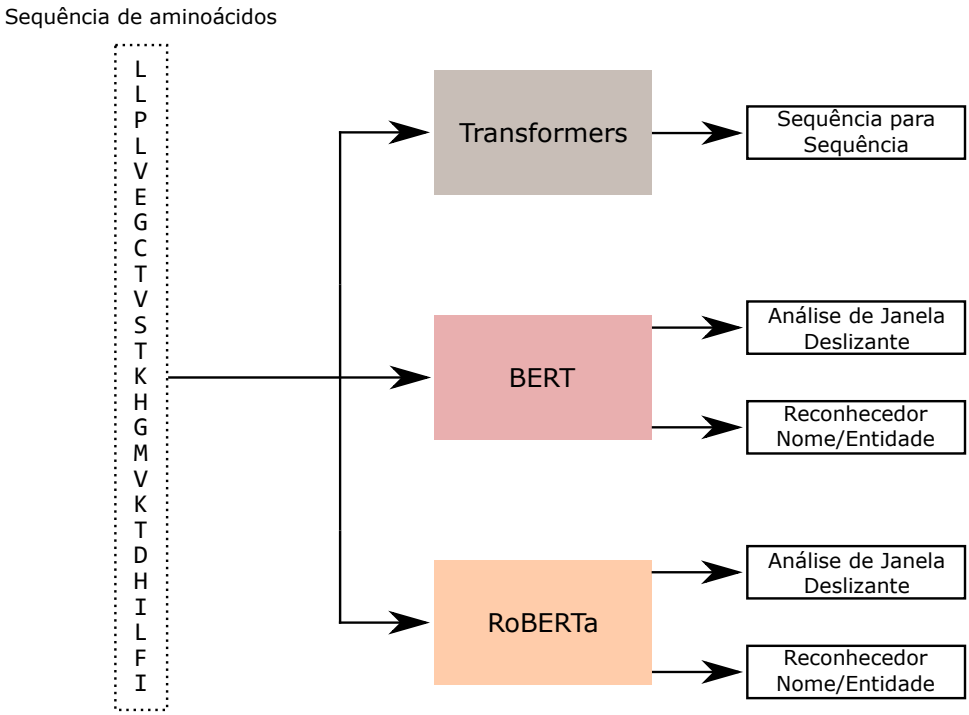


Figura 5.6: Arquiteturas baseadas em *Transformers* utilizados para a predição de estruturas secundárias.

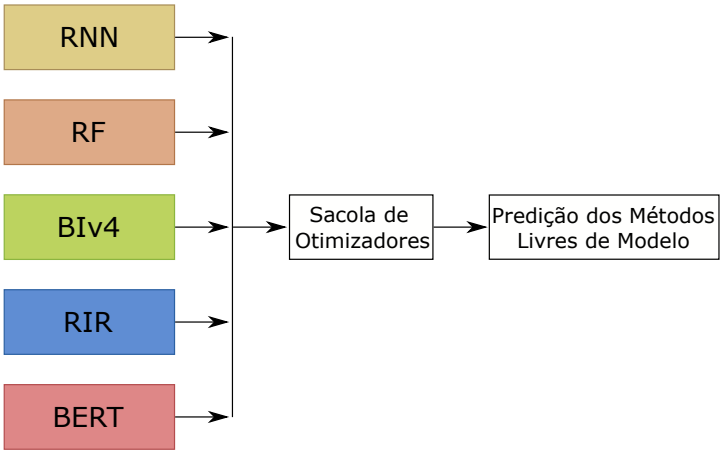


Figura 5.7: Fusão dos métodos livres de modelo.

5.3 Métodos Baseados em Modelo

Métodos baseados em modelo utilizam ferramentas que analisam proteínas com alta similaridade de sequências ou outras informações no conjunto de treinamento e no conjunto de teste para predizer as estruturas secundárias. Esta abordagem parte do princípio que proteínas similares tendem a possuir estruturas secundárias similares.

Como classificador baseado em modelo, utilizamos a ferramenta BLAST como preditor de estruturas secundárias. Para base de consulta, utilizamos todas as proteínas disponíveis no PDB até o ano de 2018, retirando a proteína da consulta das respostas obtidas. Utilizamos duas configuração de classificadores, sendo um mais restritivo, sempre garan-

tindo bons alinhamentos locais, e o outro mais abrangente, obtendo alinhamentos com menor restrição.

Para o primeiro classificador, avaliamos configurações restritivas em relação à quantidade de alinhamentos selecionadas, restrições por *E-value* e a utilização de pesos maiores para os melhores alinhamentos. Dentre as configurações, escolhemos para o primeiro classificador os 5 melhores alinhamentos, restritos pelo *E-value* menor ou igual a 10^{-10} e com pesos decrescentes de 10 até 1, isto é, o peso do melhor alinhamento corresponde a 10, do segundo melhor alinhamento corresponde a 9, até o peso igual a 1 para o décimo alinhamento. Esta configuração foi encontrada no conjunto de validação da base de dados CB6133. A probabilidade de cada estrutura é dada pela votação ponderada das estruturas similares encontradas no PDB.

Como segundo classificador, avaliamos configurações mais abrangentes, permitindo que aminoácidos que não sejam analisados pelo primeiro classificador tenham a estrutura secundária predita. Exploramos classificadores com *E-value* maiores, com buscas em uma janela de vizinhos próximos, que funcionam com o intuito de atribuir uma estrutura baseada em votação de vizinhos próximos, com a busca extrapolada caso não existam informações na janela analisada e com pesos para alinhamentos melhores, isto é, caso a busca retorne 100 alinhamentos, o melhor alinhamento terá peso igual a 100, o segundo melhor alinhamento terá peso igual a 99, até que o centésimo alinhamento tenha peso igual a 1. A melhor configuração encontrada no conjunto de validação da base CB6133 foi com pesos crescentes para melhores alinhamentos, alinhamentos restritos pelo *E-value* igual a 10, janela de vizinhança no tamanho igual a 201 e com busca extrapolada caso nenhuma estrutura seja predita na vizinhança. A probabilidade de cada estrutura é dada pela votação ponderada das estruturas similares encontradas no PDB.

Neste caso, consideramos apenas a sequência de aminoácidos para este classificador, ou seja, não utilizamos as informações da matriz de pontuação de posição específica.

Como os métodos podem não predizer todas as estruturas secundárias das proteínas, indicamos para as estruturas sem predição um vetor de probabilidades com todos os valores iguais a 0. Utilizamos essa abordagem para realizar a fusão dos dois classificadores, assim como a fusão com os métodos livres de modelo, já que a técnica de fusão utilizada, que é apresentada na Seção 5.5, necessita de valores para todas as estruturas (aceitando valores nulos). O resultado da fusão dos dois métodos é considerado a predição final dos métodos baseados em modelo. A Figura 5.8 ilustra a fusão dos dois métodos e a predição final dos métodos baseados em modelo.

5.4 Fusão dos Métodos Livres de Modelo e Baseados em Modelo

Ao final da fusão dos métodos livres de modelo e da fusão dos métodos baseados em modelo, realizamos a fusão das duas abordagens. Para realizar este procedimento, utilizamos o método de fusão apresentado na Seção 5.5.

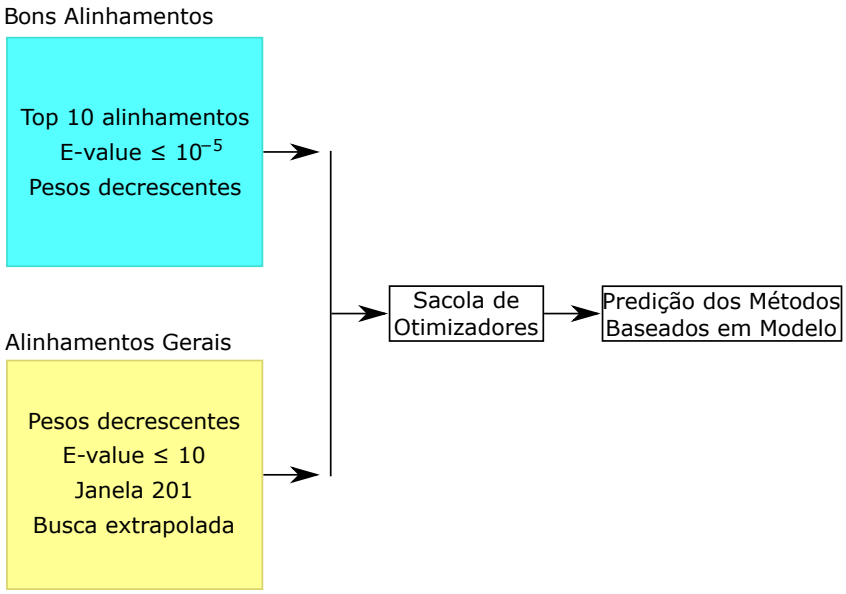


Figura 5.8: Fusão entre classificadores baseados em modelo.

5.5 Método de Fusão

Como os classificadores podem possuir visões diferentes e complementares para uma mesma tarefa de classificação, a fusão entre eles pode resultar em melhorias na predição final.

Para realizar a fusão entre os classificadores de um mesmo modelo, assim como a fusão entre todos os métodos livres de modelo, todos os métodos baseados em modelo e a fusão entre métodos livres de modelo e baseados em modelo, nós desenvolvemos um método, chamado de sacola de otimizadores, para realizar a fusão.

O objetivo da sacola de otimizadores é encontrar pesos para cada classe de cada um dos classificadores, de modo que seja feita uma média ponderada entre as predições dos classificadores presentes na fusão. Para realizar o processo de otimização, as saídas de cada um dos classificadores são utilizadas, sendo que as predições consistem em probabilidades de cada uma das classes para cada aminoácido.

A sacola de otimizadores é composta por três algoritmos de otimização distintos, sendo que ao final o melhor resultado entre eles é escolhido. Dentre os algoritmos de otimização, o algoritmo genético obteve os melhores resultados em 75% dos processos de fusão, seguido pela busca cuco, com 20%, e a otimização por enxame de partículas, com 5%. Os pesos das fusões encontrados foram obtidos a partir das predições dos classificadores no conjunto de validação de cada base de dados. O Algoritmo 4 descreve o processo da sacola de otimizadores.

5.5.1 Algoritmo Genético

O primeiro otimizador da sacola de otimizadores é o algoritmo genético. Nós dividimos o processo do otimizador em duas etapas, busca global e busca local, sendo iniciado pela busca global.

Algoritmo 4: Sacola de Otimizadores.

Entrada: Predições dos classificadores que irão participar da fusão e as classes reais dos aminoácidos das proteínas no conjunto de validação
Saída: Pesos para cada classe de cada classificador
início
  Encontra os pesos usando o Algoritmo Genético
  Encontra os pesos usando a Busca Cuco
  Encontra os pesos usando a Otimização por Enxame de Partículas
  se *pesos encontrados no Algoritmo Genético atingirem melhor acurácia no conjunto de validação do que os pesos encontrados na Busca Cuco e na Otimização por Enxame de Partículas* **então**
    retorna *pesos encontrados no Algoritmo Genético*
  fim
  senão se *pesos encontrados na Busca Cuco atingirem melhor acurácia no conjunto de validação do que os pesos encontrados no Algoritmo Genético e na Otimização por Enxame de Partículas* **então**
    retorna *pesos encontrados na Busca Cuco*
  fim
  senão
    retorna *pesos encontrados na Otimização por Enxame de Partículas*
  fim
fim

Na etapa de busca global, geramos uma população inicial de tamanho igual a 2.000 com pesos variando entre 0 e 10 e definimos o número máximo de iterações como 100. Para cada geração, realizamos a verificação de cada indivíduo em relação a função de custo, que definimos como sendo a acurácia Q3 ou Q8 mediante o cálculo de probabilidade do classificador multiplicado pelos pesos do indivíduo em questão.

Após todos os indivíduos serem avaliados, selecionamos os 100 melhores para serem os pais da próxima geração. Os pais geram 900 novos indivíduos através de cruzamentos. Para os cruzamentos, dois pais diferentes são selecionados aleatoriamente e, para cada um dos pesos, é selecionado o peso equivalente de um dos pais. Ao final, os filhos se juntam aos pais para formar os 1.000 primeiros indivíduos da próxima geração.

A partir dos 1.000 primeiros indivíduos da próxima geração, novos 1.000 indivíduos são gerados por meio de mutações. Nesta etapa, para cada um dos indivíduos, selecionamos um peso aleatoriamente e alteramos a partir de uma multiplicação do peso atual por 1 mais um valor aleatório, caso o número seja positivo, ou a divisão de 1 menos um valor aleatório, caso o número seja negativo. Selecionamos o número randômico por uma função normal, com média igual a 0 e desvio padrão igual a 1. Os parâmetros de média e desvio padrão foram selecionados a partir de experimentação. Ao final de cada geração, normalizamos os pesos de cada indivíduo com a função de normalização pela média.

Como o algoritmo pode não encontrar soluções melhores caso encontre um mínimo local, utilizamos o conceito de *early stopping* para diminuir o processamento desnecessário. Como marcador de iterações para aguardar melhorias, utilizamos o valor igual a 5.

Após a etapa de busca global, realizamos a busca local. Inicialmente, selecionamos

os 100 melhores indivíduos vindos da busca global. A partir dos 100 indivíduos, geramos novos 900 indivíduos através de mutações, com o mesmo método utilizado na etapa de mutação da busca global. Ao final da geração, normalizamos os pesos dos indivíduos pela normalização pela média e terminamos a geração. Executamos este processo por 100 gerações e, ao final, o melhor indivíduo é escolhido.

Assim como na busca global, utilizamos o conceito de *early stopping* para diminuir o processamento desnecessário, com o marcador de iterações com o valor igual a 5.

5.5.2 Busca Cuco

O segundo otimizador da sacola de otimizadores é a busca cuco. O algoritmo é iniciado com uma população de tamanho igual a 1.000, com pesos variando entre 0 e 10. Após a inicialização da população, ordenamos os melhores indivíduos por uma função de custo, que definimos sendo a acurácia Q3 ou Q8.

Ao início de cada geração, calculamos o ovo de cada indivíduo por meio dos voos de Lévy e avaliamos o ovo pela função de custo. Para cada ovo, selecionamos um indivíduo aleatório que contém outro ovo para a comparação e colocamos uma cópia do melhor ovo da comparação naquele indivíduo selecionado.

Após todos os ovos serem comparados com outro ovo qualquer, ordenamos os indivíduos por ordem decrescente de acurácia Q3 ou Q8. Ao final da geração, reinicializamos os 25% piores indivíduos com novos pesos gerados aleatoriamente com valores entre 0 e 10.

Assim como no algoritmo genético, utilizamos o número máximo de iterações igual a 100 e o mecanismo de *early stopping* igual a 5.

5.5.3 Otimização por Enxame de Partículas

O último otimizador da sacola de otimizadores é a otimização por enxame de partículas. O algoritmo é iniciado com uma população de tamanho igual a 1.000, com pesos variando entre 0 e 10 e com movimento igual a 0.

No começo de cada geração, avaliamos a posição de cada indivíduo por uma função de custo, que definimos como sendo a acurácia Q3 ou Q8. Caso a posição atual do indivíduo seja superior que a melhor posição anterior, atualizamos a melhor posição individual. Além da melhor posição individual, a melhor posição global é atualizada caso a posição atual do indivíduo seja superior ao melhor resultado global.

Ao final de cada geração, atualizamos a velocidade e a posição de todos os indivíduos. Para a atualização da velocidade, realizamos a soma ponderada entre a velocidade atual, a diferença entre a melhor posição individual e a posição atual e a diferença entre a melhor posição global e a posição atual. Como pesos, utilizamos o valor de 0,5 para a velocidade, a multiplicação de 0,8 por um número real aleatório entre 0 e 1 para a diferença entre a melhor posição individual pela posição atual e a multiplicação de 0,9 por um número real aleatório entre 0 e 1 para a diferença entre a melhor posição global e a posição atual. Fizemos avaliações de diferentes valores para os parâmetros e o melhor resultado obtido foi utilizando os valores usados no artigo que propõe o método [40].

A atualização da posição é feita pela soma da posição atual com a velocidade atualizada do indivíduo. Assim como no algoritmo genético e na busca cuco, utilizamos o número máximo de iterações igual a 100 e o mecanismo de *early stopping* igual a 5.

Capítulo 6

Resultados Experimentais utilizando Sequência de Aminoácidos

Neste capítulo, apresentamos os resultados experimentais utilizando sequência de aminoácidos para os métodos livres de modelo e baseados em modelo, assim como a fusão entre eles.

6.1 Métodos Livres de Modelos

Nesta seção, apresentamos e discutimos os resultados experimentais utilizando sequência de aminoácidos para os classificadores RNN, RF, BIv4, RIR e *Transformers*.

6.1.1 Redes Neurais Bidirecionais Recorrentes

Nesta subseção, apresentamos os resultados do classificador RNN, iniciando da escolha da arquitetura no conjunto de validação da base CB6133 e finalizando na avaliação da arquitetura selecionada para a classificação Q8 nos conjuntos de teste das bases CB6133, CB513 e PDB e para a classificação Q3 no conjunto de teste da base PDB.

Escolha da Arquitetura

Para a escolha das arquiteturas das redes bidirecionais recorrentes, optamos por treinar e validar as variações dos modelos no conjunto CB6133 pois a distribuição de classes possui valores próximos com outras bases de dados. Para a avaliação dos modelos, utilizamos a acurácia Q8.

Inicialmente, avaliamos a rede com uma camada bidirecional recorrente e 100 neurônios, analisando a sequência de aminoácidos no sentido padrão, ou seja, no sentido em que os aminoácidos eram encontrados na proteína, e a fusão de duas redes com a mesma arquitetura, porém uma rede analisando a sequência no sentido padrão e outra rede analisando no sentido invertido. Para a segunda abordagem, calculamos a média da predição das duas redes. A Tabela 6.1 mostra os resultados das duas abordagens, sendo que a fusão de duas redes apresentou melhores resultados. Com isso, utilizamos esta configuração como base para os próximos experimentos.

Rede	Acurácia Q8 (%)
Rede analisando sentido original	58,10
Fusão de redes analisando ambos os sentidos	**58,77**

Tabela 6.1: Comparação dos resultados das RNNs analisando os dados no sentido original e a fusão das redes analisando ambos os sentidos no conjunto de validação da base CB6133.

Após selecionarmos a fusão das duas redes com a mesma arquitetura, investigamos variações na quantidade de camadas na rede, entre 1 a 10 camadas, mantendo 100 neurônios em cada camada. A Tabela 6.2 descreve os resultados da arquitetura com diferentes quantidades de camadas bidirecionais recorrentes. As redes com 2 até 6 camadas obtiveram os melhores resultados.

Camadas	Acurácia Q8 (%)
1	58,77
2	**60,10**
3	**60,42**
4	**60,26**
5	**60,31**
6	**60,27**
7	60,07
8	59,74
9	59,90
10	60,02

Tabela 6.2: Comparação dos resultados das RNNs com diferentes números de camadas no conjunto de validação da base CB6133.

Na sequência, verificamos a quantidade de neurônios por camada. Para isto, realizamos a fusão das cinco configurações (redes com 2 até 6 camadas) e avaliamos a quantidade de neurônios por camada, variando valores entre 100 e 1000, de 100 em 100. Para a fusão, utilizamos a sacola de otimizadores. A Tabela 6.3 apresenta os resultados da fusão para cada configuração de neurônios por camada, com o melhor resultado em destaque. Pelos resultados obtidos, escolhemos a rede com 600 neurônios por camada.

Por fim, avaliamos a utilização da camada de *embedding* para a sequência de aminoácidos. A Tabela 6.4 mostra os resultados da rede com e sem *embedding*, com melhor resultado em destaque. Com os resultados obtidos, optamos por utilizar as redes com 2 até 6 camadas, 600 neurônios por camada e uma camada de *embedding* para a sequência de aminoácidos.

Base CB6133

Após selecionarmos as cinco variações de arquiteturas de redes bidirecionais recorrentes, treinamos e testamos cada uma delas no conjunto de dados CB6133. A Tabela 6.5 apresenta os resultados de cada uma das redes. A rede que atingiu o maior valor de acurácia Q8 foi com 2 camadas bidirecionais recorrentes, com 60,18% de acurácia Q8.

Neurônios	Acurácia Q8 (%)
100	61,44
200	62,21
300	62,44
400	62,91
500	62,78
600	**63,10**
700	62,62
800	62,72
900	62,48
1000	62,63

Tabela 6.3: Comparação dos resultados das RNNs com diferentes números de neurônios por camada no conjunto de validação da base CB6133.

Rede	Acurácia Q8 (%)
Com *embedding*	**63,42**
Sem *embedding*	63,10

Tabela 6.4: Comparação dos resultados da fusão de RNNs com e sem *embedding* no conjunto de validação da base CB6133.

Camadas	Acurácia Q8 (%)
2	60,18
3	60,13
4	59,78
5	59,57
6	58,62
Fusão	**61,63**

Tabela 6.5: Resultados das RNNs no conjunto de teste da base CB6133.

Após a análise das redes separadas, fizemos a fusão entre elas utilizando a sacola de otimizadores. Como resultado, a fusão atingiu 61,63% de acurácia Q8.

Base CB513

Com as cinco arquiteturas selecionadas, treinamos cada uma na base CB6133 filtrada e testamos na base CB513. A Tabela 6.6 apresenta os resultados individuais de cada rede. A arquitetura com 4 camadas atingiu o melhor resultado, com 57,49% de acurácia Q8, enquanto a pior rede, com 6 camadas, atingiu 54,41%.

Na sequência, após avaliarmos as redes individualmente, utilizamos a sacola de otimizadores para realizar a fusão das cinco redes. Como resultado, a fusão atingiu 57,58% de acurácia Q8.

Camadas	Acurácia Q8 (%)
2	57,43
3	57,31
4	57,49
5	56,91
6	54,41
Fusão	**57,58**

Tabela 6.6: Resultados das RNNs no conjunto de teste da base CB513.

Base PDB

Além das bases CB6133 e CB513, treinamos e testamos as cinco variações de modelos na base PDB. Para esta base, verificamos as arquiteturas sozinhas e as fusões em relação às métricas acurácia Q8 e acurácia Q3.

Inicialmente, avaliamos as redes individualmente utilizando a classificação Q8. A Tabela 6.7 apresenta os resultados obtidos, onde a melhor rede, com 3 camadas, atingiu 67,53% de acurácia Q8.

Na sequência, realizamos a fusão entre as cinco variações de arquitetura. Como resultado, a fusão obteve 68,48% de acurácia Q8, superando as redes individuais.

Em seguida, utilizamos a classificação Q3 para avaliar as redes. A Tabela 6.7 apresenta os resultados obtidos. O melhor resultado foi obtido pela rede com 2 camadas, que atingiu 78,61% de acurácia Q3.

Por último, aplicamos a fusão das redes com a sacola de otimizadores. Após a fusão, alcançamos o resultado de 79,22% de acurácia Q3.

Camadas	Acurácia Q8 (%)	Acurácia Q3 (%)
2	67,32	78,61
3	67,53	77,69
4	67,14	77,85
5	66,86	78,58
6	66,91	77,48
Fusão	**68,48**	**79,22**

Tabela 6.7: Resultados das RNNs no conjunto de teste da base PDB.

6.1.2 Florestas Aleatórias

Nesta subseção, apresentamos os resultados do classificador RF, partindo da escolha das janelas deslizantes no conjunto de validação da base CB6133, seguida da avaliação dos classificadores nos conjuntos de teste das base CB6133, CB513 e PDB. Para todas as bases, utilizamos a acurácia Q8, além da acurácia Q3 para o PDB.

Escolha das Janelas

Para a escolha das janelas de análise das florestas aleatórias, treinamos e validamos cada uma das configurações na base CB6133. Para avaliar os resultados, utilizamos a acurácia Q8.

Para este tipo de classificador, deve-se utilizar janelas deslizantes para analisar trechos de sequências. Como não existe consenso entre valores ótimos de janela nos trabalhos da literatura, analisamos diversos tamanhos e escolhemos as cinco melhores para compor a fusão de RFs. Nós avaliamos tamanhos de janelas entre 3 a 21, variando de 2 em 2, conforme apresentado na Tabela 6.8.

Janela	**Acurácia Q8 (%)**
3	44,53
5	44,65
7	51,63
9	**53,22**
11	**53,57**
13	**53,43**
15	**53,53**
17	**53,31**
19	53,18
21	53,22

Tabela 6.8: Comparação dos resultados das RFs com diferentes janelas deslizantes no conjunto de validação da base CB6133.

Com os resultados obtidos, optamos por utilizar florestas aleatórias com janelas de tamanho entre 9 e 17 aminoácidos.

Base CB6133

Após selecionarmos as cinco variações de classificadores, treinamos e testamos cada um deles na base CB6133. A Tabela 6.9 apresenta o resultado de cada um dos classificadores no conjunto de teste. Dentre os classificadores, o melhor resultado foi obtido pelo modelo com janela deslizante de tamanho igual a 15, que atingiu a acurácia Q8 igual a 53,41%.

Janela	**Acurácia Q8 (%)**
9	48,21
11	48,65
13	53,32
15	53,41
17	53,30
Fusão	**54,20**

Tabela 6.9: Resultados das RFs no conjunto de teste da base CB6133.

Na sequência, utilizamos o método de fusão apresentado na Seção 5.5 para fundir os cinco classificadores. Como resultado da fusão, obtivemos 54,20% de acurácia Q8.

Base CB513

Como segundo experimento, treinamos os cinco classificadores na base CB6133 filtrada e testamos na base CB513. A Tabela 6.10 apresenta os resultados dos classificadores. Assim como no conjunto de teste da base CB6133, o classificador que obteve a melhor acurácia Q8 foi o modelo com janela deslizante de tamanho igual a 15, com acurácia Q8 igual a 48,87%.

Janela	**Acurácia Q8 (%)**
9	43,71
11	44,09
13	48,85
15	48,87
17	48,63
Fusão	**49,64**

Tabela 6.10: Resultados das RFs no conjunto de teste da base CB513.

Em seguida, realizamos a fusão entre os cinco classificadores utilizando o método sacola de otimizadores. A fusão resultou em uma acurácia Q8 igual a 49,64%, superando os classificadores individuais.

Base PDB

Para a base PDB, treinamos e testamos os cinco classificadores e avaliamos cada um deles e as fusões em relação a acurácia Q8 e acurácia Q3.

Inicialmente, avaliamos os classificadores separadamente utilizando a classificação Q8. A Tabela 6.11 apresenta os resultados obtidos por cada um dos classificadores com diferentes tamanhos de janelas deslizantes. Como melhor resultado, o classificador com janela deslizante de tamanho 13 obteve 68,97% de acurácia Q8.

Em seguida, fundimos os cinco classificadores usando a sacola de otimizadores. Com a fusão, atingimos 69,75% de acurácia Q8, superando os classificadores individuais.

Na sequência, utilizamos a classificação Q3 para avaliar cada um dos modelos. A Tabela 6.11 apresenta os resultados de acurácia Q3 obtidos de cada um dos classificadores.

Por último, realizamos a fusão entre os cinco modelos utilizando a sacola de otimizadores. Após a fusão, alcançamos o resultado de 78,65% de acurácia Q3.

6.1.3 Blocos Inception-v4

Nesta subseção, apresentamos os resultados do classificador blocos Inception-v4 (BIv4), iniciando da escolha dos modelos no conjunto de validação da base CB6133 e finalizando na avaliação dos modelos selecionados nos conjuntos de teste das bases CB6133, CB513

Janela	Acurácia Q8 (%)	Acurácia Q3 (%)
9	68,51	77,57
11	68,92	77,90
13	68,97	78,00
15	68,93	78,17
17	68,95	77,97
Fusão	**69,75**	**78,65**

Tabela 6.11: Resultados das RFs no conjunto de teste da base PDB.

e PDB, sendo que o último foi avaliado tanto para a classificação Q8 quanto para a classificação Q3.

Escolha dos Modelos

Para a escolha dos modelos para a predição de estruturas secundárias, testamos os três diferentes blocos que compõem a arquitetura Inception-v4, chamados de blocos A, B e C, além de variações na quantidade de blocos de um mesmo tipo, entre 1 e 10. Com os resultados obtidos com RNNs com camadas de *embedding*, adotamos inicialmente a mesma estratégia para este classificador.

A Tabela 6.12 apresenta os resultados de acurácia Q8 para diferentes quantidades de blocos no conjunto de validação da base CB6133. A partir dos resultados, escolhemos os modelos com 3 a 7 blocos do tipo B para experimentos adicionais.

	Acurácia Q8 (%)		
Quantidade	**Blocos A**	**Blocos B**	**Blocos C**
1	49,12	56,28	49,21
2	53,87	58,43	54,19
3	55,99	**59,03**	56,61
4	57,15	**58,93**	57,78
5	57,37	**58,59**	58,25
6	57,63	**58,93**	57,58
7	57,89	**58,60**	58,76
8	57,20	57,55	57,48
9	57,80	58,70	58,56
10	57,99	58,75	57,90

Tabela 6.12: Comparação dos resultados com diferentes blocos Inception-v4 e quantidades de blocos no conjunto de validação da base CB6133.

Na sequência, verificamos a fusão desses cinco modelos com camada de *embedding* para a sequência de aminoácidos e a fusão de modelos sem esta configuração. A Tabela 6.13 ilustra que a camada de *embedding* aumentou o valor de acurácia Q8. Sendo assim, optamos por manter a camada de *embedding*.

Fusão	Acurácia Q8 (%)
Com *embedding*	**60,67**
Sem *embedding*	57,13

Tabela 6.13: Comparação dos resultados da fusão de BIv4 com e sem *embedding* no conjunto de validação da base CB6133.

Base CB6133

Com os cinco modelos selecionados, treinamos e testamos cada um deles na base CB6133. A Tabela 6.14 apresenta a acurácia Q8 de cada um dos modelos. Dentre as redes, o modelo com 4 blocos do tipo B atingiu 58,98% de acurácia Q8.

Blocos	Acurácia Q8 (%)
3	58,92
4	58,98
5	58,02
6	58,86
7	57,96
Fusão	**60,40**

Tabela 6.14: Resultados dos BIv4 no conjunto de teste da base CB6133.

Após analisarmos os modelos individualmente, realizamos a fusão entre eles utilizando a sacola de otimizadores. Como resultado, a fusão obteve 60,40% de acurácia Q8.

Base CB513

Além do treinamento e teste na base CB6133, realizamos o treinamento na base CB6133 filtrada e testamos os modelos na base CB513. A Tabela 6.15 ilustra os resultados obtidos por cada um dos modelos. Diferentemente dos experimentos na base CB6133, a rede com 3 blocos obteve o melhor resultado.

Blocos	Acurácia Q8 (%)
3	55,68
4	55,04
5	54,70
6	54,90
7	55,52
Fusão	**56,73**

Tabela 6.15: Resultados dos BIv4 no conjunto de teste da base CB513.

Na sequência, fundimos as cinco diferentes configurações através da sacola de otimizadores. A fusão alcançou 56,73% de acurácia Q8, superando os classificadores individuais.

Base PDB

Nos experimentos realizados na base PDB, avaliamos cada um dos modelos e as fusões entre eles em relação às métricas acurácia Q8 e acurácia Q3.

Na etapa inicial, executamos a classificação Q8 para cada um dos classificadores. Os resultados obtidos individualmente por cada um dos modelos são apresentados na Tabela 6.16.

Em seguida, realizamos a fusão entre os cinco diferentes classificadores usando a sacola de otimizadores. Com a fusão, obtivemos 71,64% de acurácia Q8, superando os modelos analisados individualmente.

Após a classificação Q8, analisamos cada um dos cinco classificadores utilizando a classificação Q3. A Tabela 6.16 apresenta os valores de acurácia Q3 atingidos por cada um dos classificadores.

Por fim, fizemos a fusão entre os cinco modelos usando a sacola de otimizadores. Após a fusão, alcançamos o resultado de 78,26% de acurácia Q3.

Blocos	**Acurácia Q8 (%)**	**Acurácia Q3 (%)**
3	63,48	77,29
4	69,53	74,73
5	65,89	75,95
6	69,49	78,26
7	65,07	73,78
Fusão	**71,64**	**78,26**

Tabela 6.16: Resultados dos BIv4 no conjunto de teste da base PDB.

6.1.4 Redes Inception Recorrentes

Nesta subseção, apresentamos os resultados dos classificadores Rede Inception Recorrente (RIR), inciando da escolha das arquiteturas no conjunto de validação da base CB6133, seguida da realização de experimentos nas bases CB6133, CB513 e PDB.

Escolha da Arquitetura

Para os classificadores RIR, utilizamos o conjunto de validação da base CB6133 para avaliar variações de camadas bidirecionais recorrentes nas cinco variações de classificadores blocos Inception-v4. Para cada um dos classificadores, analisamos configurações com 1 até 5 camadas recorrentes e de 100 até 500 neurônios por camada. A Tabela 6.17 apresenta os cinco melhores resultados da fusão das redes testadas.

Como configuração para as redes RIR, escolhemos as arquiteturas com 3 camadas bidirecionais recorrentes e com 100 neurônios por camada.

Base CB6133

Com as cinco arquiteturas selecionadas, treinamos e testamos cada uma delas na base de dados CB6133. A Tabela 6.18 mostra os resultados de cada uma das arquiteturas, sendo

Camadas	Neurônios	Acurácia Q8 (%)
3	**100**	**59,92**
1	100	59,85
5	200	59,73
3	400	59,69
2	300	59,62

Tabela 6.17: Comparação das variações da arquitetura RIR no conjunto de validação da base CB6133.

que a rede com 5 blocos obteve o melhor resultado, 60,40% de acurácia Q8.

Blocos	Acurácia Q8 (%)
3	59,92
4	60,18
5	60,40
6	60,13
7	60,04
Fusão	**60,99**

Tabela 6.18: Resultados das redes RIR no conjunto de teste da base CB6133.

Após avaliarmos as arquiteturas individualmente, fizemos a fusão entre elas com a sacola de otimizadores. A fusão obteve 60,99% de acurácia, superando as redes individuais.

Base CB513

Na sequência, treinamos as cinco diferentes arquiteturas na base CB6133 filtrada e testamos na base CB513. A Tabela 6.19 apresenta a acurácia Q8 de cada uma das arquiteturas no conjunto de teste da base CB513. Diferentemente da base CB6133, a rede com 3 blocos obteve o melhor resultado de acurácia Q8.

Blocos	Acurácia Q8 (%)
3	57,12
4	56,78
5	56,74
6	56,85
7	56,62
Fusão	**57,68**

Tabela 6.19: Resultados das redes RIR no conjunto de teste da base CB513.

Em seguida, fundimos as cinco arquiteturas utilizando a sacola de otimizadores. O resultado da fusão atingiu 57,68% de acurácia Q8.

Base PDB

Além da avaliação das arquiteturas nas base CB6133 e CB513, realizamos experimentos na base PDB, tanto para a classificação Q8 quanto para classificação Q3, treinando os modelos no conjunto de treinamento e medindo os resultado no conjunto de teste.

No primeiro experimento, avaliamos cada uma das arquiteturas em relação à classificação Q8. A Tabela 6.20 ilustra os resultados obtidos individualmente por cada um dos modelos, sendo que a rede com 7 blocos alcançou o melhor resultado, com 69,55% de acurácia Q8.

Na sequência, fundimos as cinco arquiteturas utilizando o método descrito na Seção 5.5. Com isto, a fusão atingiu 71,98% de acurácia Q8.

Em seguida, avaliamos os modelos em relação à classificação Q3. A Tabela 6.20 apresenta os resultados individuais de cada arquitetura, onde a rede com 6 blocos obteve o melhor resultado.

Por fim, realizamos a fusão das arquiteturas através da sacola de otimizadores. Como resultado, a fusão conseguiu 81,91% de acurácia Q3, superando as arquiteturas individuais.

Blocos	**Acurácia Q8 (%)**	**Acurácia Q3 (%)**
3	69,45	78,75
4	66,03	78,81
5	68,54	77,92
6	68,02	80,35
7	69,55	79,02
Fusão	**71,98**	**81,91**

Tabela 6.20: Resultados das redes RIR no conjunto de teste da base PDB.

6.1.5 Transformers

Nesta subseção, apresentamos os resultados de classificadores baseados em *Transformers*. Inicialmente, selecionamos o classificador no conjunto de validação da base CB6133, e em seguida, avaliamos o método escolhido nos conjuntos de teste das bases CB6133, CB513 e PDB.

Escolha do Classificador

Para a escolha dos classificadores baseados em *Transformers*, verificamos três adaptações de tarefas de processamento de linguagem natural para a predição de estruturas secundárias.

Como primeiro classificador, utilizamos a arquitetura *Transformers* para realizar a tarefa de tradução de aminoácidos para estruturas secundárias. Com isto, avaliamos diferentes quantidade de blocos *encoder* e *decoder* para a tradução, variando entre 1 e 5 blocos, além de variações no vetor latente e no *embedding*, com tamanhos 64, 128, 256 e 512. A Tabela 6.21 apresenta os resultados das cinco melhores configurações.

Encoder	*Decoder*	Vetor Latente	*Embedding*	Acurácia Q8 (%)
1	**1**	**128**	**64**	**44,63**
1	1	128	256	44,45
1	1	256	64	44,27
1	1	128	128	44,08
1	1	256	128	43,92

Tabela 6.21: Comparação dos resultados com diferentes valores para *encoder*, *decoder*, vetor latente e *embedding* para classificadores *Transformers* no conjunto de validação da base CB6133.

Como segundo classificador, avaliamos o BERT e o RoBERTa para análise de trechos, ou janelas, em uma tarefa próxima à análise de sentimento de frases. Para cada janela, analisamos o aminoácido central. Dentre os tamanhos de janelas, utilizamos valores entre 21 até 201, variando de 20 em 20. A Tabela 6.22 apresenta os resultados das janelas utilizando o BERT e o RoBERTa.

	Acurácia Q8 (%)	
Janela	**BERT**	**RoBERTa**
21	57,12	56,72
41	57,11	56,63
61	57,12	57,01
81	57,11	57,37
101	**58,27**	57,05
121	**58,25**	56,53
141	**58,38**	57,44
161	**58,33**	58,14
181	**58,65**	58,15
201	58,12	57,99

Tabela 6.22: Comparação dos resultados com diferentes valores de janelas para classificadores BERT e RoBERTa no conjunto de validação da base CB6133.

Por último, avaliamos o BERT e o RoBERTa para análise de *tokens*. Nesta tarefa, cada um dos *tokens*, ou aminoácidos, é classificado conforme a classe, neste caso, estruturas secundárias. A Tabela 6.23 mostra os resultados do BERT e o RoBERTa para análise de *tokens*.

Arquitetura	Acurácia Q8 (%)
BERT	**55,82**
RoBERTa	54,37

Tabela 6.23: Comparação dos resultados dos classificadores BERT e RoBERTa para classificação de *tokens* no conjunto de validação da base CB6133.

Com os resultados, optamos pelo BERT com janelas deslizantes de tamanho 101, 121,

141, 161 e 181 para experimentos adicionais.

Base CB6133

A partir dos classificadores selecionados, avaliamos cada um deles na base de dados CB6133. A Tabela 6.24 ilustra os resultados obtidos por cada um dos classificadores em relação à acurácia Q8. Dentre os classificadores, o BERT com janela igual a 161 obteve o melhor resultado, com 58,82% de acurácia Q8.

Janela	**Acurácia Q8 (%)**
101	58,32
121	58,73
141	58,49
161	58,82
181	58,64
Fusão	**59,60**

Tabela 6.24: Resultados dos classificadores BERT no conjunto de teste da base CB6133.

Na sequência, avaliamos a fusão dos classificadores BERT. Para a fusão, utilizamos a sacola de otimizadores. Como resultado, a fusão obteve 59,60% de acurácia Q8, superando as redes individuais.

Base CB513

Após a avaliação dos classificadores na base CB6133, treinamos cada um deles na base CB6133 filtrada e testamos na base CB513. A Tabela 6.25 mostra a acurácia Q8 de cada um dos classificadores no conjunto de teste da base CB513.

Janela	**Acurácia Q8 (%)**
101	54,99
121	55,12
141	55,09
161	55,29
181	55,36
Fusão	**55,86**

Tabela 6.25: Resultados dos classificadores BERT no conjunto de teste da base CB513.

Em seguida, realizamos a fusão entre os cinco classificadores BERT utilizando a sacola de otimizadores. Com a fusão, atingimos 55,86% de acurácia Q8.

Base PDB

Após a avaliação dos classificadores nas bases CB6133 e CB513, analisamos os resultados de acurácia Q8 e acurácia Q3 na base de dados PDB.

Inicialmente, avaliamos cada um dos classificadores BERT pela acurácia Q8. A Tabela 6.26 apresenta os resultados obtidos individualmente por cada um dos classificadores.

Na sequência, realizamos a fusão dos cinco classificadores BERT utilizando o método descrito na Seção 5.5. Com isto, a fusão atingiu 63,04% de acurácia Q8.

Após avaliarmos os classificadores BERT na acurácia Q8, realizamos a categorização no formato Q3. A Tabela 6.26 apresenta os resultados atingidos por cada um dos classificadores em relação à acurácia Q3.

Por fim, fundimos os cinco classificadores BERT através da sacola de otimizadores. A fusão obteve 75,68% de acurácia Q3, superando as arquiteturas individuais.

Janela	**Acurácia Q8 (%)**	**Acurácia Q3 (%)**
101	60,05	73,43
121	61,13	74,26
141	61,48	74,54
161	61,15	74,13
181	60,75	74,04
Fusão	**63,04**	**75,68**

Tabela 6.26: Resultados dos classificadores BERT no conjunto de teste da base PDB.

6.1.6 Fusão de Métodos Livres de Modelo

Após a análise individual dos métodos livres de modelo, realizamos experimentos para a fusão entre os cinco diferentes classificadores para as bases CB6133, CB513 e PDB.

Base CB6133

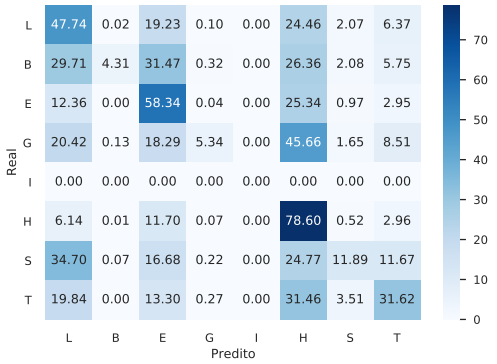
Antes de fundirmos os classificadores, comparamos as matrizes de confusão dos métodos para avaliarmos quais classes cada um deles possui maior taxa de verdadeiros positivos. A Figura 6.1 apresenta as matrizes de confusão dos cinco métodos no conjunto de teste da base CB6133. Em todos os modelos, as classes majoritárias "H", "L", "E" e "T" foram mais vezes preditas corretamente. Em relação às classes minoritárias, o classificador RF obteve o melhor resultado na classe "B", o classificador RNN atingiu o melhor resultado na classe "S" e o classificador RIR obteve o melhor resultado na classe "G".

Após a análise qualitativa dos resultados de cada classificador, fizemos a fusão entre os cinco métodos. A Tabela 6.27 mostra os resultados dos métodos individuais e da fusão, sendo que a fusão ultrapassou os resultados dos classificadores individuais. Como não existem resultados para esta base, não podemos comparar os resultados com a literatura.

A Tabela 6.28 ilustra os pesos encontrados na fusão. Pelos valores dos pesos, o classificador que é mais importante, ou seja, que possui maiores pesos na maioria das classes, é o RF, mesmo não sendo o método que obteve o maior valor de acurácia Q8.

Além da acurácia Q8, apresentamos os valores de precisão e revocação para cada classe na classificação da fusão dos métodos livres de modelo na Tabela 6.29. A avaliação de precisão e revocação por classe é uma prática comum dos trabalhos da literatura.

(a) RNN.



(b) RF.

(c) BIv4.

(d) RIR.

(e) BERT.

Figura 6.1: Matrizes de confusão dos métodos livres de modelo no conjunto de teste da base CB6133.

A Figura 6.2 ilustra a matriz de confusão da fusão dos métodos livres de modelo. Pelos valores na matriz de confusão, fica evidente a classificação dos dados de classes minoritárias, como “B” e “G”, como classes majoritárias, como “H”, “E”, “L” e “T”.

Método	Acurácia Q8 (%)
Fusão	**62,69**
RNN	61,63
RIR	60,99
BIv4	60,40
BERT	59,60
RF	54,20

Tabela 6.27: Resultados dos métodos individuais e da fusão dos métodos livres de modelo no conjunto de teste da base CB6133.

Classe	RNN	RF	BIv4	RIR	BERT
B	0,582	3,690	0,438	1,768	1,197
E	1,458	1,058	0,026	0,442	0,175
G	1,168	2,072	0,249	0,149	0,991
H	3,316	0,512	0,193	1,569	0,020
I	0,401	0,089	0,215	2,775	3,883
L	0,851	1,769	0,180	0,449	0,091
S	0,758	2,557	0,175	0,676	0,107
T	0,892	2,204	0,426	0,398	0,017

Tabela 6.28: Pesos de cada classe de cada classificador na fusão dos métodos livres de modelo na base CB6133.

Classe	Precisão	Revocação	Frequência (%)
H	0,75	0,85	35,67
E	0,60	0,75	21,56
L	0,53	0,51	18,57
T	0,47	0,42	11,11
S	0,43	0,19	7,92
G	0,46	0,12	4,06
B	0,93	0,04	1,10
I	—	—	0,00

Tabela 6.29: Precisão e revocação da fusão dos métodos livres de modelo no conjunto de teste da base CB6133.

Base CB513

Assim como foi feito nos experimentos na base CB6133, realizamos a comparação das matrizes de confusão dos métodos na Figura 6.3. As predições das classes majoritárias, "H", "L" e "E", tiveram altos valores de verdadeiros positivos.

Na sequência da análise qualitativa das matrizes de confusão de cada classificador, realizamos a fusão entre os métodos utilizando da sacola de otimizadores. A Tabela 6.30 apresenta os resultados dos classificadores individuais e da fusão.

A Tabela 6.31 mostra os pesos encontrados para cada classe de cada classificador na

Figura 6.2: Matriz de confusão da fusão dos métodos livres de modelo no conjunto de teste da base CB6133.

Método	**Acurácia Q8 (%)**
Fusão	**57,81**
RIR	57,68
RNN	57,58
BIv4	56,73
BERT	55,86
RF	49,64

Tabela 6.30: Resultados dos métodos individuais e da fusão dos métodos livres de modelo no conjunto de teste da base CB513.

fusão. Assim como na fusão na base CB6133, o classificador RF obteve valores altos para os pesos na maioria das classes.

Na sequência, comparamos o nosso método com a literatura, conforme mostra na Tabela 6.32. Os resultados mostram que nosso método está a 3,8 pontos percentuais do estado da arte.

Além da métrica acurácia Q8, os métodos da literatura apresentam a precisão e revocação por classe. Como o estado da arte não avaliou as classes com estas métricas, a Tabela 6.33 apresenta apenas o nosso método.

Por fim, a Figura 6.4 ilustra a matriz de confusão da fusão dos métodos livres de modelo. A classificação das classes majoritárias, como “H”, “E” e “L”, atingiram taxas maiores de verdadeiros positivos comparado com classes minoritárias, como “B” e “G”.

(a) RNN.

(b) RF.

(c) BIv4.

(d) RIR.

(e) BERT.

Figura 6.3: Matrizes de confusão dos métodos livres de modelo no conjunto de teste da base CB513.

Base PDB

Inicialmente, avaliamos cada um dos classificadores em relação à acurácia Q8. A Figura 6.5 apresenta as matrizes de confusão dos cinco métodos no conjunto de teste da base PDB. Assim como nas base CB6133 e CB513, as classes majoritárias, como "H" e "E", tiveram maiores taxas de verdadeiros positivos. Outro ponto importante é que os métodos acertaram dados da classe "I".

Classe	RNN	RF	BIv4	RIR	BERT
B	0,182	0,711	0,674	0,720	0,570
E	1,085	0,847	0,018	0,528	0,268
G	0,996	1,514	0,196	0,609	0,604
H	0,301	0,734	0,423	0,955	0,122
I	0,541	0,943	0,707	0,496	0,813
L	0,387	0,948	0,665	0,506	0,094
S	1,037	1,506	0,050	0,618	0,181
T	0,656	1,635	0,087	0,557	0,075

Tabela 6.31: Pesos de cada classe de cada classificador na fusão dos métodos livres de modelo na base CB513.

Método	Acurácia Q8 (%)
Ratul *et al.* [66]	61,6
Nosso método	**57,8**
Guo *et al.* [25]	57,1

Tabela 6.32: Comparação do resultado obtido com outros trabalhos da literatura no conjunto de teste da base CB513.

Classe	Precisão	Revocação	Frequência (%)
H	0,73	0,82	30,88
E	0,55	0,76	21,25
L	0,48	0,49	21,14
T	0,45	0,37	11,81
S	0,37	0,13	9,81
G	0,34	0,08	3,69
B	0,45	0,00	1,39
I	0,00	0,00	0,03

Tabela 6.33: Precisão e revocação da fusão dos métodos livres de modelo no conjunto de teste da base CB513.

Após a análise qualitativa dos resultados de cada classificador na classificação Q8, fizemos a fusão entre os cinco métodos. A Tabela 6.34 apresenta os resultados dos métodos individuais e da fusão, sendo que o resultado da fusão obteve resultado superior aos classificadores individuais. Não foi possível comparar os resultados com a literatura, já que nenhum trabalho apresentou a acurácia Q8 nesta base.

A Tabela 6.35 mostra os pesos encontrados na fusão dos métodos livres de modelo. Assim como nos pesos das bases CB6133 e CB513, o classificador RF possui grande importância na classificação, porém em algumas classes, como "G" e "L", o peso deste classificador é menor comparado com os outros classificadores.

Além da acurácia Q8, apresentamos os valores de precisão e revocação para cada classe na classificação do nosso método na Tabela 6.36.

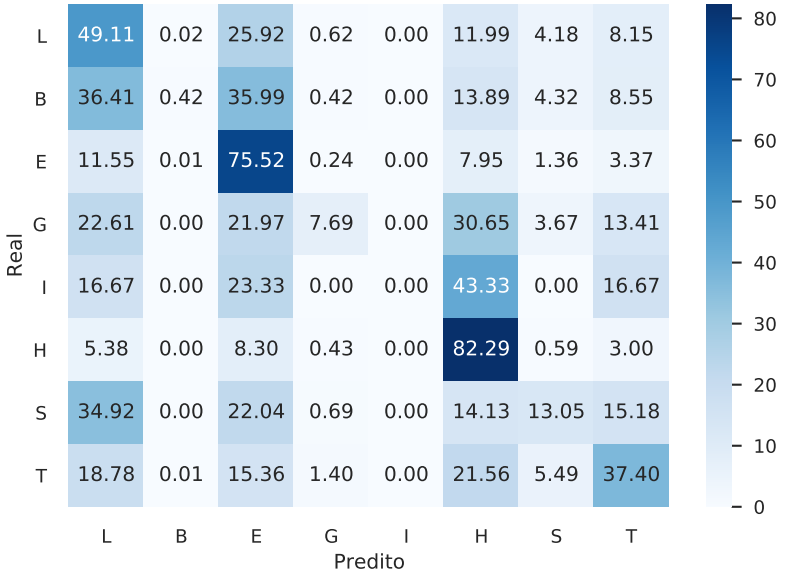


Figura 6.4: Matriz de confusão da fusão dos métodos livres de modelo no conjunto de teste da base CB513.

Método	**Acurácia Q8 (%)**
Fusão	**74,76**
RIR	71,98
BIv4	71,64
RF	69,75
RNN	68,48
BERT	63,04

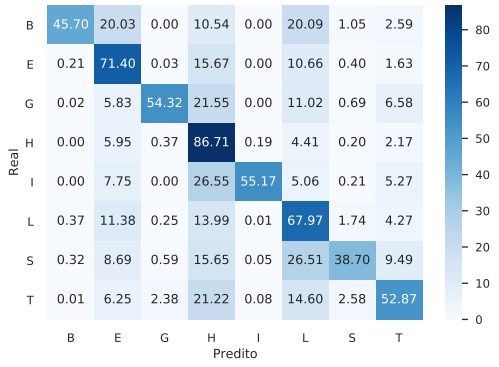
Tabela 6.34: Resultados dos métodos individuais e da fusão dos métodos livres de modelo no conjunto de teste da base PDB para a classificação Q8.

Classe	**RNN**	**RF**	**BIv4**	**RIR**	**BERT**
B	1,004	2,120	0,193	0,913	0,106
E	1,246	3,030	1,048	0,894	0,002
G	2,031	0,606	0,014	1,282	0,192
H	0,731	2,084	0,635	1,206	0,000
I	0,437	3,132	2,221	0,056	0,091
L	0,101	0,736	0,312	2,331	0,089
S	0,942	2,660	1,314	0,563	0,423
T	1,505	2,039	0,493	1,191	0,011

Tabela 6.35: Pesos de cada classe de cada classificador na fusão dos métodos livres de modelo na base PDB para a classificação Q8.

A Figura 6.6 ilustra a matriz de confusão da fusão dos métodos livres de modelo. Como resultado da fusão, classes majoritárias, como “H” e “E”, tiveram maiores taxas de

(a) RNN.



(b) RF.

(c) BIv4.

(d) RIR.

(e) BERT.

Figura 6.5: Matrizes de confusão dos métodos livres de modelo no conjunto de teste da base PDB para a classificação Q8.

verdadeiros positivos. Outro ponto importante é a taxa de verdadeiros positivos acima de 50% para a classe "I", diferentemente do que aconteceu nas outras bases.

Após a análise na classificação Q8, realizamos a classificação Q3. Inicialmente, avaliamos as matrizes de confusão dos métodos, conforme apresentado na Figura 6.7. Pelos valores das matrizes, percebemos que dados da classe "E" são classificados, em média 20% da vezes, como sendo da classe "C", além da confusão entre as classes "H" e "C".

Após a análise qualitativa dos resultados de cada classificador em relação à acurácia

Classe	Precisão	Revocação	Frequência (%)
H	0,83	0,91	33,58
L	0,67	0,72	21,23
E	0,74	0,80	19,36
T	0,64	0,58	10,88
S	0,70	0,43	8,91
G	0,76	0,56	4,11
B	0,84	0,39	1,25
I	0,79	0,56	0,66

Tabela 6.36: Precisão e revocação da fusão dos métodos livres de modelo no conjunto de teste da base PDB para a classificação Q8.

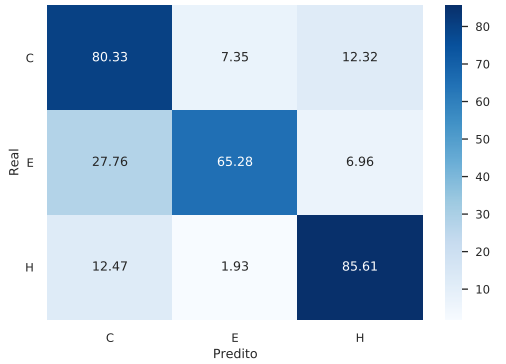
Figura 6.6: Matriz de confusão da fusão dos métodos livres de modelo no conjunto de teste da base PDB para a classificação Q8.

Q3, realizamos a fusão entre os cinco diferentes métodos. A Tabela 6.37 apresenta os resultados dos métodos individuais e da fusão. Não foi possível comparar os resultados com a literatura, já que nenhum trabalho apresentou a acurácia Q3 nesta base.

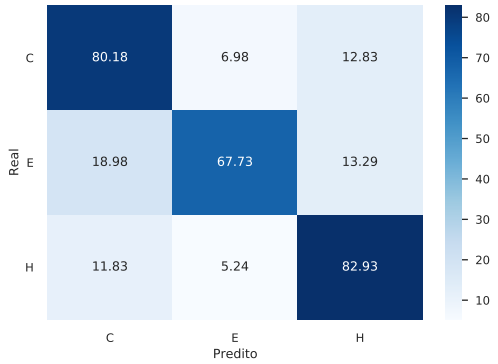
A Tabela 6.38 mostra os pesos encontrados na fusão dos métodos livres de modelo. Para as classes "E" e "H", o classificador RNN possui pesos maiores, enquanto o classificador RF possui maior peso na classe "C".

Além da acurácia Q3, apresentamos os valores de precisão e revocação para cada classe na classificação do nosso método na Tabela 6.39.

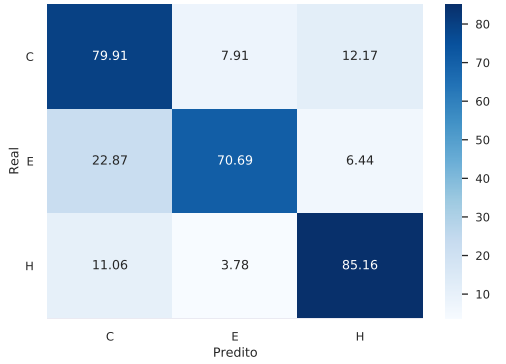
A Figura 6.8 ilustra a matriz de confusão da fusão dos métodos livres de modelo. Assim como nas matrizes apresentadas na Figura 6.7, dados da classe "E" foram preditos como "C" e existe uma confusão entre as classes "C" e "H".



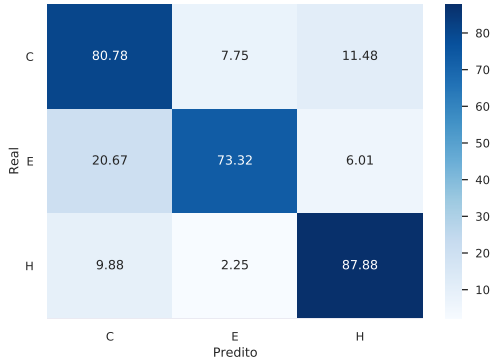
(a) RNN.



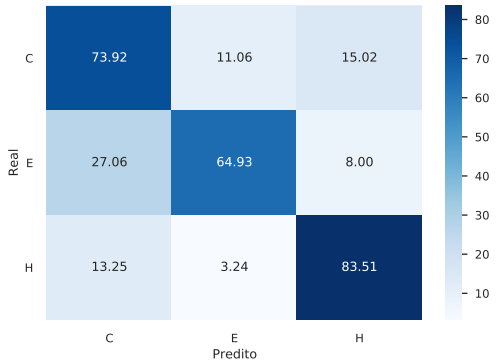
(b) RF.



(c) BIv4.



(d) RIR.



(e) BERT.

Figura 6.7: Matrizes de confusão dos métodos livres de modelo no conjunto de teste da base PDB para a classificação Q3.

6.2 Métodos Baseados em Modelo

Nesta seção, apresentamos os resultados de métodos baseados em modelo. Para a criação de classificadores que usam esta abordagem, utilizamos a ferramenta BLAST para alinhar sequências de proteínas similares das bases em questão com as proteínas de toda a base PDB, predizendo as estruturas a partir das estruturas secundárias das proteínas do PDB.

Como podem ocorrer bons alinhamentos, ou seja, alinhamentos com baixo valor de *E*-

Método	Acurácia Q3 (%)
Fusão	**83,03**
RIR	81,91
RNN	79,22
RF	78,65
BIv4	78,26
BERT	75,68

Tabela 6.37: Resultados dos métodos individuais e da fusão dos métodos livres de modelo no conjunto de teste da base PDB para a classificação Q3.

Classe	RNN	RF	BIv4	RIR	BERT
C	0,652	2,800	0,711	0,709	0,058
E	1,503	1,473	0,000	1,493	0,618
H	2,044	1,138	0,111	1,183	0,386

Tabela 6.38: Pesos de cada classe de cada classificador na fusão dos métodos livres de modelo na base PDB para a classificação Q3.

Classe	Precisão	Revocação	Frequência (%)
C	0,83	0,82	41,69
H	0,85	0,89	37,70
E	0,80	0,76	20,61

Tabela 6.39: Precisão e revocação da fusão de métodos livres de modelo no conjunto de teste da base PDB para a classificação Q3.

value, porém curtos, e alinhamentos não tão significativos, mas com boa parte das estruturas preditas, criamos dois classificadores, sendo um para bons alinhamentos e outro para alinhamentos mais gerais, e fundimos os dois métodos usando a sacola de otimizadores.

6.2.1 Classificador para Bons Alinhamentos

Nesta subseção, apresentamos os resultados do classificador para bons alinhamentos. Inicialmente, selecionamos as configurações da ferramenta BLAST no conjunto de validação da base CB6133, e na sequência, testamos a configuração selecionada nas bases CB6133, CB513 e PDB.

Escolha dos Parâmetros

Para o classificador com bons alinhamentos, avaliamos restrições para alinhamentos com *E-value* menor que 10^{-10}, 10^{-5}, 10^{-1} e 1, além de verificar os 5, 10, 20 melhores ou todos os alinhamentos, e a utilização de pesos, calculados a partir da subtração da quantidade de alinhamentos (P) pela posição no ranking (T) mais um, de modo que alinhamentos melhores tenham pesos maiores. A Tabela 6.40 apresenta os cinco melhores

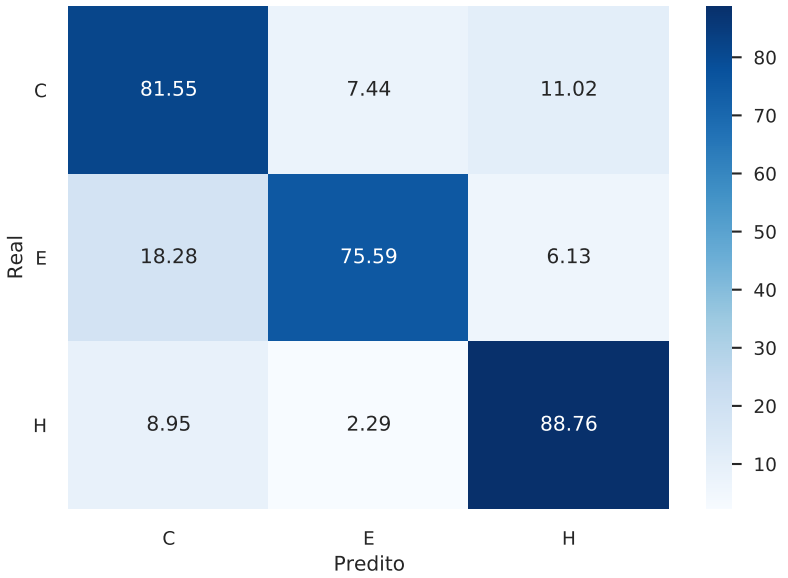


Figura 6.8: Matriz de confusão da fusão dos métodos livres de modelo no conjunto de teste da base PDB para a classificação Q3.

resultados no conjunto de validação da base CB6133, levando em conta a acurácia Q8. Comparando os resultados, selecionamos o classificador com os 5 melhores alinhamentos, com pesos maiores para os melhores alinhamentos e com *E-value* menor que 10^{-10}, já que foi a configuração que obteve a maior acurácia Q8.

Alinhamento Utilizado	***E-value***	**Pesos**	**Acurácia Q8 (%)**	**Estruturas Preditas (%)**
Top 5	10^{-10}	**Sim**	**88,40**	**73,03**
Todos	10^{-10}	Sim	87,49	75,95
Top 10	10^{-5}	Sim	86,96	79,56
Top 10	10^{-1}	Não	85,26	80,48
Top 5	1	Sim	83,63	84,82

Tabela 6.40: Escolha dos parâmetros do classificador com bons alinhamentos no conjunto de validação da base CB6133.

Base CB6133

Após selecionarmos as configurações do classificador com bons alinhamentos no conjunto de validação da base CB6133, realizamos a predição das estruturas secundárias das proteínas do conjunto de teste da base CB6133. Como o classificador pode não conseguir uma classificação para todo aminoácido, consideramos que o método errou estes casos. Com isso, o classificador com bons alinhamentos atingiu 69,15% de acurácia Q8 no conjunto de teste da base CB6133.

Base CB513

Após o experimento na base de dados CB6133, classificamos as estruturas secundárias das proteínas do conjunto de teste da base CB5133. Ao final da predição, o método de classificação de bons alinhamentos alcançou a acurácia de 76,34%.

Base PDB

No experimento nas proteínas de 2018 do PDB, consideramos tanto a classificação Q8 quanto a classificação Q3. Para a classificação Q8 das proteínas do conjunto de teste do PDB, o método obteve 68,63% de acurácia Q8, enquanto para a classificação Q3, o classificador de bons alinhamentos atingiu 82,11% de acurácia Q3.

6.2.2 Classificador com Alinhamentos Gerais

Nesta subseção, apresentamos e discutimos os resultados do classificador com alinhamentos gerais. A escolha dos parâmetros deste classificador foi feita no conjunto de validação da base CB6133. Na sequência, avaliamos o classificador nos conjuntos de teste das bases CB6133, CB513 e PDB.

Escolha dos Parâmetros

Para o classificador com alinhamentos mais gerais, avaliamos restrições de *E-value* menores que 1, 10 e 100, a utilização de pesos maiores para alinhamentos maiores, tamanho de janelas entre 3, 11, 51, 101 e 201 para a classificação de *gaps* (neste caso, a predição das estruturas secundárias levaria em conta os alinhamentos próximos para predizer a estrutura secundária do aminoácido que foi alinhado com um *gap*) e busca extrapolada caso a janela não seja suficiente para a predição de estruturas. A Tabela 6.41 mostra as cinco melhores configurações em termos de acurácia Q8, que foram capazes de predizer 100% das estruturas secundárias do conjunto de validação. Comparando os resultados, selecionamos as configurações destacadas na Tabela 6.41.

Janela	***E-value***	**Pesos**	**Busca Extrapolada**	**Acurácia Q8 (%)**
201	**10**	**Sim**	**Sim**	**76,61**
51	10	Sim	Sim	76,54
101	10	Sim	Sim	76,52
201	10	Sim	Não	76,49
3	10	Sim	Sim	76,38

Tabela 6.41: Escolha dos parâmetros do classificador com alinhamentos gerais no conjunto de validação da base CB6133.

Base CB6133

Após a escolha das configurações para o classificador com alinhamentos gerais, avaliamos este método no conjunto de teste da base CB6133. Com isto, o preditor obteve 75,96% de acurácia Q8.

Base CB513

Na sequência, avaliamos o modelo com alinhamentos gerais no conjunto de teste da base de dados CB513. Neste caso, o método atingiu 86,43% de acurácia Q8.

Base PDB

No experimento na base do PDB, classificamos as proteínas seguindo as categorizações Q8 e Q3. Para a classificação Q8 das proteínas do conjunto de teste do PDB, o classificador com alinhamentos gerais atingiu 86,36% de acurácia Q8, enquanto para a classificação Q3, o método obteve 91,74% de acurácia Q3.

6.2.3 Fusão de Métodos Baseados em Modelo

Após a avaliação individual dos classificadores que utilizam a ferramenta BLAST, realizamos a fusão entre eles. Como o método para bons alinhamentos pode não ter predições para todas as estruturas, adicionamos o vetor de probabilidade com todos os valores zerados neste caso.

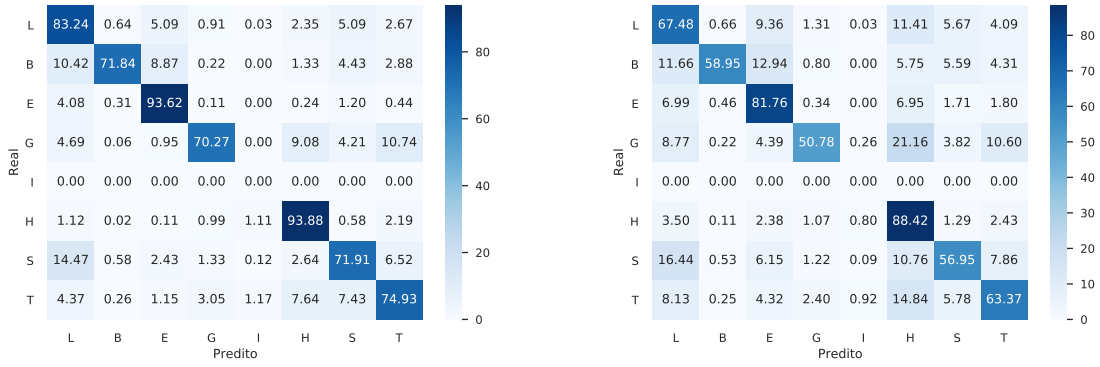
Base CB6133

Antes de fundirmos os classificadores, comparamos as matrizes de confusão dos métodos para avaliarmos quais classes cada um deles possui maior taxa de verdadeiros positivos. A Figura 6.9 ilustra as matrizes de confusão das duas variações de classificadores baseados na ferramenta BLAST no conjunto de teste da base CB6133. Dentre os classificadores, o método com bons alinhamentos atingiu mais verdadeiros positivos em todas as classes comparado com o método de alinhamento gerais.

Na sequência, fundimos os dois métodos. A fusão obteve 78,73% de acurácia no conjunto de teste da base CB6133, conforme apresentado na Tabela 6.42. Não existem resultados para esta base nos trabalhos da literatura.

Método	**Acurácia Q8 (%)**
Fusão	**78,73**
Alinhamentos Gerais	75,96
Bons Alinhamentos	69,15

Tabela 6.42: Resultados dos métodos individuais e da fusão dos métodos baseados em modelo no conjunto de teste da base CB6133.



(a) Bons alinhamentos.

(b) Alinhamentos gerais.

Figura 6.9: Matrizes de confusão dos métodos baseados em modelo no conjunto de teste da base CB6133.

A Tabela 6.43 mostra os pesos dos dois classificadores na fusão entre os métodos. Em todas as classes, exceto "I" e "H", o classificador com bons alinhamentos obteve maiores pesos.

Classe	Bons alinhamentos	Alinhamentos gerais
B	1,711	0,506
E	1,718	0,719
G	1,717	0,515
H	1,239	1,452
I	0,001	0,041
L	1,280	0,910
S	1,546	0,517
T	1,350	0,773

Tabela 6.43: Pesos de cada classe de cada classificador na fusão dos métodos baseados em modelo na base CB6133.

Em seguida, avaliamos a precisão e revocação de cada uma das classes. A Tabela 6.44 apresenta estes dados.

Por fim, a Figura 6.10 ilustra a matriz de confusão da fusão dos métodos baseados em modelo. Dentre os erros, podemos destacar classificações de classes minoritárias, como "B", "G" e "S", como classes majoritárias, especialmente "L" e "H".

Base CB513

Inicialmente, verificamos a matriz de confusão dos dois métodos para o conjunto de teste da base CB513. A Figura 6.11 mostra as matrizes de confusão das duas variações de classificadores baseados na ferramenta BLAST no conjunto de teste da base CB513. Novamente, o classificador com bons alinhamentos obteve maiores taxas de verdadeiros positivos na maioria das classes, sendo que para a classe "I", o classificador com alinhamentos gerais atingiu melhores resultados.

Classe	Precisão	Revocação	Frequência (%)
H	0,82	0,91	35,67
E	0,87	0,84	21,56
L	0,74	0,70	18,57
T	0,72	0,66	11,11
S	0,65	0,62	7,92
G	0,71	0,58	4,06
B	0,70	0,62	1,10
I	—	—	0,00

Tabela 6.44: Precisão e revocação da fusão dos métodos baseados em modelo no conjunto de teste da base CB6133.

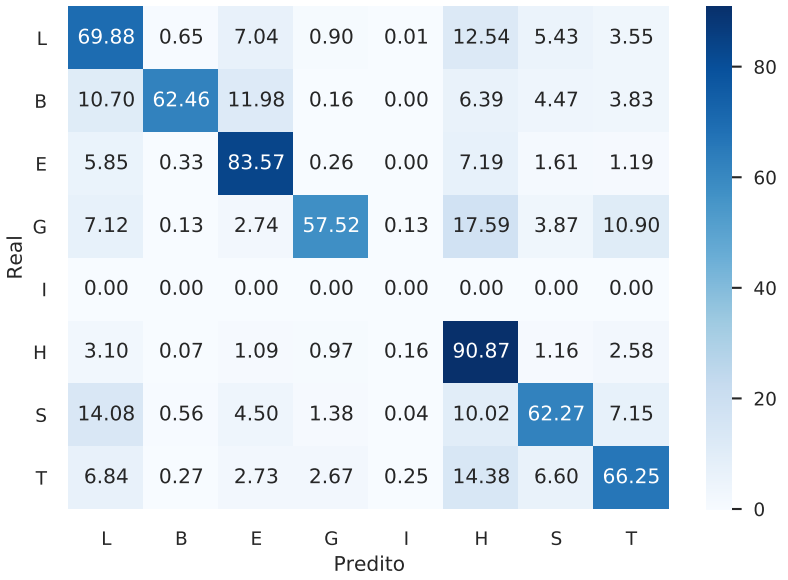


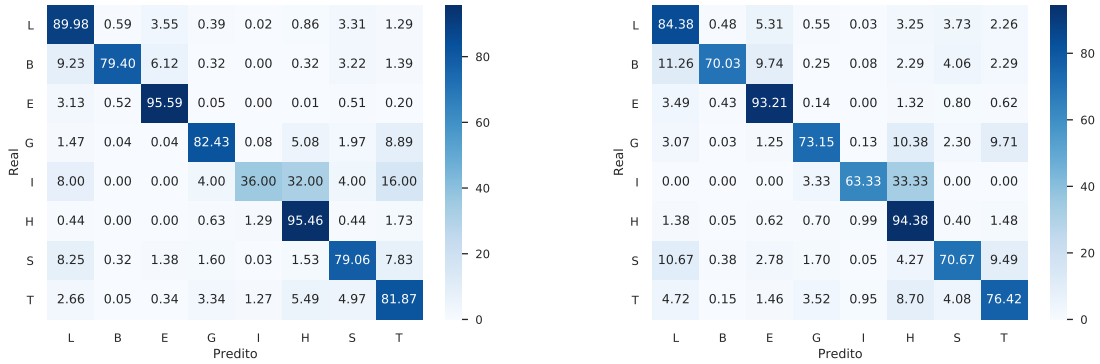
Figura 6.10: Matriz de confusão da fusão dos métodos baseados em modelo no conjunto de teste da base CB6133.

Na sequência, realizamos a fusão dos dois métodos utilizando a sacola de otimizadores. Com isto, a fusão ultrapassou os métodos individuais, conforme mostra a Tabela 6.45.

Método	Acurácia Q8 (%)
Fusão	**89,16**
Alinhamentos Gerais	86,43
Bons Alinhamentos	76,34

Tabela 6.45: Resultados dos métodos individuais e da fusão dos métodos baseados em modelo no conjunto de teste da base CB513.

A Tabela 6.46 apresenta os pesos da fusão dos dois classificadores baseados em modelo. Novamente, o classificador com bons alinhamentos obteve maiores pesos.



(a) Bons alinhamentos.

(b) Alinhamentos gerais.

Figura 6.11: Matrizes de confusão dos métodos baseados em modelo no conjunto de teste da base CB513.

Classe	Bons alinhamentos	Alinhamentos gerais
B	2,351	0,227
E	1,892	0,549
G	1,457	0,470
H	1,268	1,591
I	0,010	0,271
L	1,843	0,953
S	0,801	0,412
T	1,386	0,235

Tabela 6.46: Pesos de cada classe de cada classificador na fusão dos métodos baseados em modelo na base CB513.

Na sequência, comparamos o nosso método com a literatura, conforme mostra a Tabela 6.47. Os resultados mostram que nosso método está 27,5 pontos percentuais a frente do estado da arte.

Método	Acurácia Q8 (%)
Nosso método	**89,1**
Ratul *et al.* [66]	61,6
Guo *et al.* [25]	57,1

Tabela 6.47: Comparação do resultado com outros trabalhos da literatura no conjunto de teste da base CB513.

Além da métrica acurácia Q8, os métodos da literatura apresentam a precisão e revocação por classe. Como o estado da arte não avaliou as classes com estas métricas, a Tabela 6.48 apresenta apenas o nosso método.

Por fim, a Figura 6.12 ilustra a matriz de confusão da fusão dos métodos baseados em modelo. Pelos valores, apenas a classe "I" ficou abaixo de 75% de verdadeiros positivos.

Classe	Precisão	Revocação	Frequência (%)
H	0,93	0,96	30,88
E	0,94	0,95	21,25
L	0,89	0,87	21,14
T	0,82	0,80	11,81
S	0,82	0,76	9,81
G	0,77	0,80	3,69
B	0,81	0,76	1,39
I	0,13	0,47	0,03

Tabela 6.48: Precisão e revocação da fusão dos métodos baseados em modelos no conjunto de teste da base CB513.

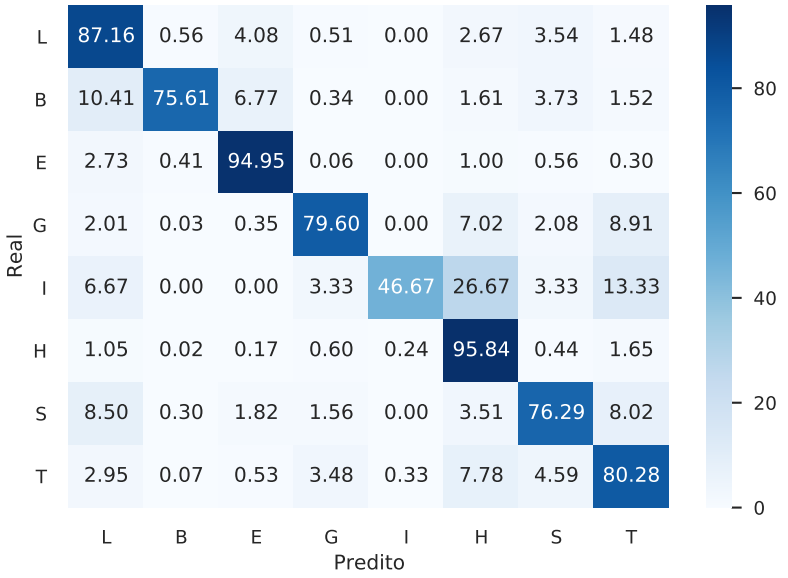


Figura 6.12: Matriz de confusão da fusão dos métodos baseados em modelo no conjunto de teste da base CB513.

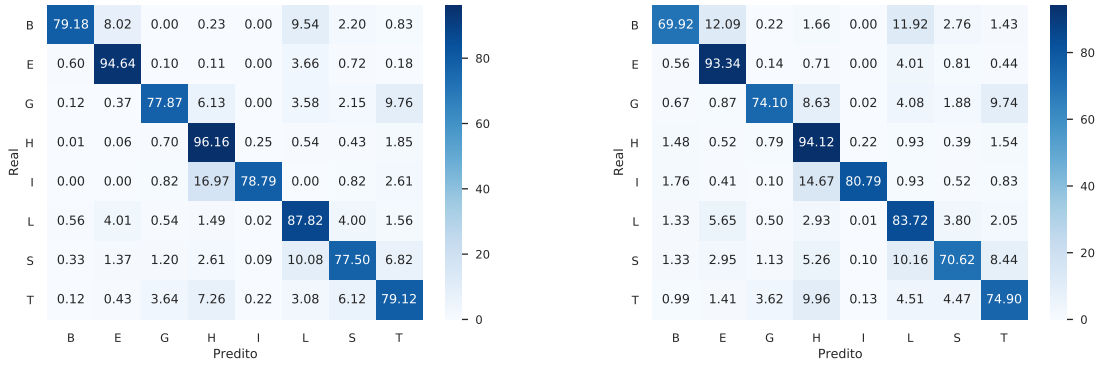
Base PDB

Para a base PDB, avaliamos os modelos tanto para a classificação Q8 quanto para a Q3. A Figura 6.13 apresenta as matrizes de confusão dos dois métodos no conjunto de teste da base PDB na classificação Q8. Ambos os classificadores obtiveram valores acima de 65% de verdadeiros positivos para todas as classes.

Após a análise das matrizes de confusão, fizemos a fusão entre os dois métodos. A Tabela 6.49 mostra os valores de cada um dos métodos e da fusão. Como não existem trabalhos para esta base, não podemos comparar os resultados com a literatura.

A Tabela 6.50 mostra os pesos encontrados na fusão. Na maioria das classes, o classificador com bons alinhamentos obteve pesos maiores comparado com os pesos do classificador com alinhamentos gerais.

Além da acurácia Q8, apresentamos os valores de precisão e revocação para cada classe



(a) Bons alinhamentos.

(b) Alinhamentos gerais.

Figura 6.13: Matrizes de confusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q8.

Método	Acurácia Q8 (%)
Fusão	**88,55**
Alinhamentos Gerais	86,36
Bons Alinhamentos	68,63

Tabela 6.49: Resultados dos métodos individuais e da fusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q8.

Classe	Bons alinhamentos	Alinhamentos gerais
B	1,197	0,724
E	1,429	0,518
G	1,051	0,967
H	1,126	0,747
I	2,219	0,607
L	1,024	0,733
S	0,804	0,920
T	1,214	0,713

Tabela 6.50: Pesos de cada classe de cada classificador na fusão dos métodos baseados em modelo na base PDB para a classificação Q8.

na classificação do nosso método na Tabela 6.51.

A Figura 6.14 ilustra a matriz de confusão do nosso método. Em todas as classes, a taxa de verdadeiros positivos ficou acima de 75%.

Após a avaliação Q8 dos métodos na base PDB, utilizamos a classificação Q3. A Figura 6.15 ilustra as matrizes de confusão dos dois métodos no conjunto de teste da base PDB na classificação Q3. Em ambos os classificadores, a taxa de verdadeiros positivos de todas as classes ficou acima de 90%.

Na sequência, fundimos os dois métodos utilizando a sacola de otimizadores. A Tabela 6.52 mostra os valores de cada um dos métodos e da fusão, sendo que o resultado da

Classe	Precisão	Revocação	Frequência (%)
H	0,93	0,95	33,58
L	0,89	0,87	21,23
E	0,93	0,94	19,36
T	0,81	0,79	10,88
S	0,81	0,76	8,91
G	0,80	0,79	4,11
B	0,46	0,78	1,25
I	0,80	0,81	0,66

Tabela 6.51: Precisão e revocação da fusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q8.

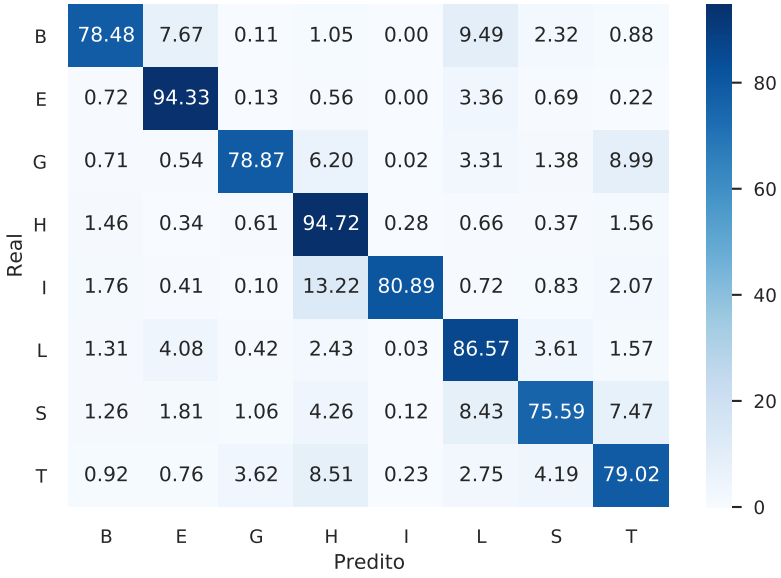


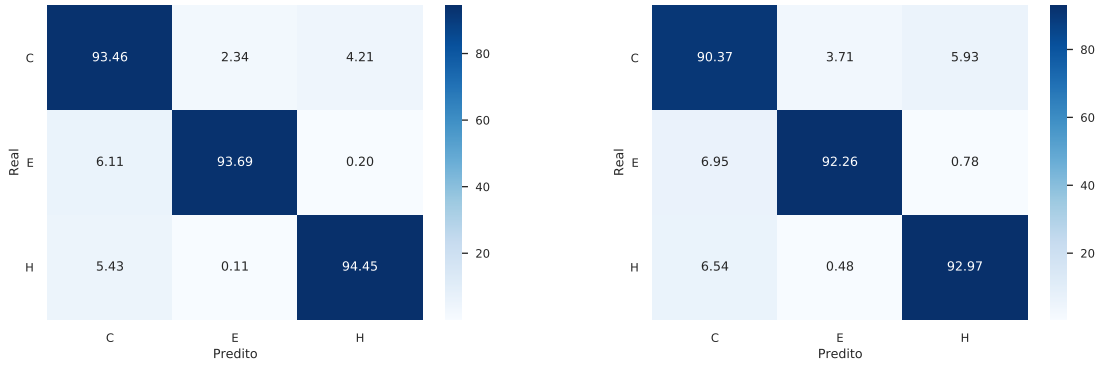
Figura 6.14: Matriz de confusão da fusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q8.

fusão ultrapassou os classificadores individuais. Como não existem trabalhos para esta base, não podemos comparar os resultados com a literatura.

Método	Acurácia Q3 (%)
Fusão	**93,21**
Alinhamentos Gerais	91,74
Bons Alinhamentos	82,11

Tabela 6.52: Resultados dos métodos individuais e da fusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q3.

A Tabela 6.53 apresenta os pesos encontrados na fusão. Nas classes "E" e "H", o classificador com bons alinhamentos obteve pesos maiores, enquanto o classificador com



(a) Bons alinhamentos.

(b) Alinhamentos gerais.

Figura 6.15: Matrizes de confusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q3.

alinhamentos gerais atingiu pesos maiores na classe "C".

Classe	Bons alinhamentos	Alinhamentos gerais
C	0,765	0,931
E	1,544	0,679
H	1,554	0,526

Tabela 6.53: Pesos de cada classe de cada classificador na fusão dos métodos baseados em modelo na base PDB para a classificação Q3.

Além da acurácia Q3 para o modelo, avaliamos os valores de precisão e revocação para cada classe na classificação do nosso método. Os valores destas métricas podem ser vistos na Tabela 6.54.

Classe	Precisão	Revocação	Frequência (%)
C	0,92	0,92	41,69
H	0,93	0,95	37,70
E	0,94	0,94	20,61

Tabela 6.54: Precisão e revocação da fusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q3.

Por fim, a Figura 6.16 ilustra a matriz de confusão do nosso método. Assim como nos métodos individuais, a diagonal principal possui todos os valores acima de 90%.

6.3 Fusão de Métodos Livres de Modelo e Métodos Baseados em Modelo

Nesta seção, apresentamos e discutimos os resultados experimentais da fusão dos métodos livres de modelo e baseados em modelo nas bases CB6133, CB513 e PDB.

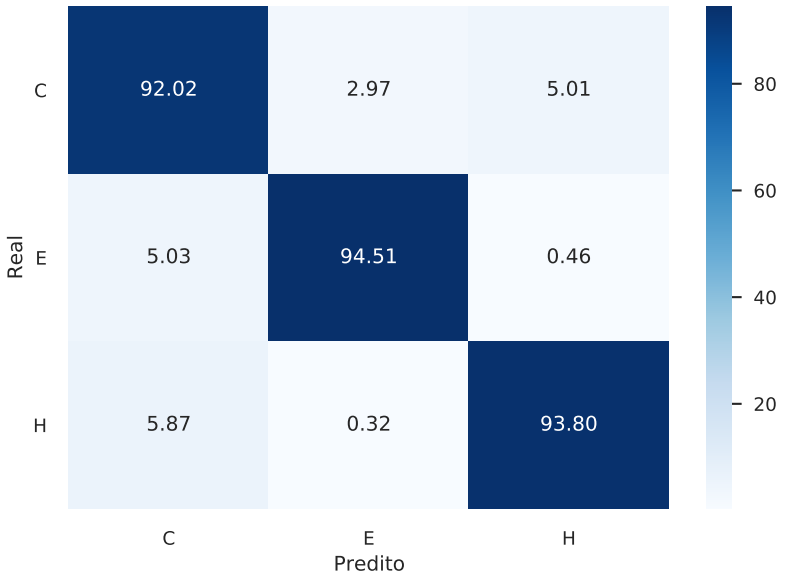


Figura 6.16: Matriz de confusão da fusão dos métodos baseados em modelo no conjunto de teste da base PDB para a classificação Q3.

Base CB6133

Com os resultados individuais dos métodos livres de modelo e baseados em modelo, fizemos a fusão entre eles utilizando a sacola de otimizadores. Para a fusão, consideramos duas abordagens. A primeira abordagem, que chamamos de fusão hierárquica, realiza a fusão entre a fusão dos métodos livres de modelo e a fusão de métodos baseados em modelo. A segunda abordagem, chamada de fusão horizontal, considera os cinco classificadores livres de modelo e os dois classificadores baseados em modelo individualmente, fundindo os sete classificadores. A Tabela 6.55 apresenta os resultados obtidos pelos dois métodos de fusão, sendo que a fusão horizontal obteve 82,83% de acurácia Q8. Não existem resultados para esta base utilizando apenas a sequência de aminoácidos nos trabalhos da literatura.

A Tabela 6.56 apresenta os pesos de cada método na fusão horizontal. Os valores obtidos pela sacola de otimizadores mostram que os maiores pesos foram dados para as predições dos métodos baseados em modelo.

Além da análise de acurácia Q8, avaliamos o método em relação a precisão e revocação, conforme mostra a Tabela 6.57.

A Figura 6.17 apresenta a matriz de confusão da fusão dos métodos livres de modelo e baseados em modelo na base CB6133. Pelos resultados obtidos, a diagonal principal fica evidente, além de erros nas predições de classes minoritárias, como “B”, “G” e “S”, preditas como classes majoritárias, como “H”, “L”, “E” e “T”.

Base CB513

A partir dos classificadores livres de modelo e baseados em modelo, realizamos a fusão entre eles considerando as abordagens hierárquica e horizontal. A Tabela 6.58 apresenta

Método	Acurácia Q8 (%)
Fusão Horizontal	**82,83**
Fusão Hierárquica	82,61
Fusão dos Métodos Baseados em Modelo	78,73
Alinhamentos Gerais	75,96
Bons Alinhamentos	69,15
Fusão dos Métodos Livres de Modelo	62,69
RNN	61,63
RIR	60,99
BIv4	60,40
BERT	59,60
RF	54,20

Tabela 6.55: Resultados dos métodos individuais e das fusões dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB6133.

Classe	RNN	RF	BIv4	RIR	BERT	Bons	Gerais
B	0,631	0,826	0,867	0,079	1,556	4,838	2,698
E	0,147	0,279	0,001	0,365	0,144	3,213	2,370
G	0,547	1,058	0,408	0,089	0,094	4,400	1,529
H	0,375	0,032	0,043	0,291	0,047	3,427	1,727
I	0,571	0,098	0,554	0,026	2,493	0,905	0,067
L	0,172	0,583	0,155	0,218	0,111	2,795	2,185
S	0,076	1,537	0,031	0,297	0,198	2,810	2,122
T	0,334	0,969	0,024	0,018	0,192	2,462	1,895

Tabela 6.56: Pesos de cada classe da fusão horizontal dos métodos livres de modelo e baseados em modelo na base CB6133.

Classe	Precisão	Revocação	Frequência (%)
H	0,90	0,94	35,67
E	0,88	0,89	21,56
L	0,77	0,78	18,57
T	0,76	0,67	11,11
S	0,66	0,64	7,92
G	0,72	0,60	4,06
B	0,71	0,65	1,10
I	—	—	0,00

Tabela 6.57: Precisão e revocação da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB6133.

os resultados individuais dos classificadores e das fusões entre eles. Em relação a acurácia Q8, as fusões pioraram comparado com a fusão dos classificadores baseados em modelo.

Como o melhor resultado das fusões entre os métodos livres de modelo e baseados em modelo foi obtido pela fusão horizontal, utilizamos esta abordagem para explorações

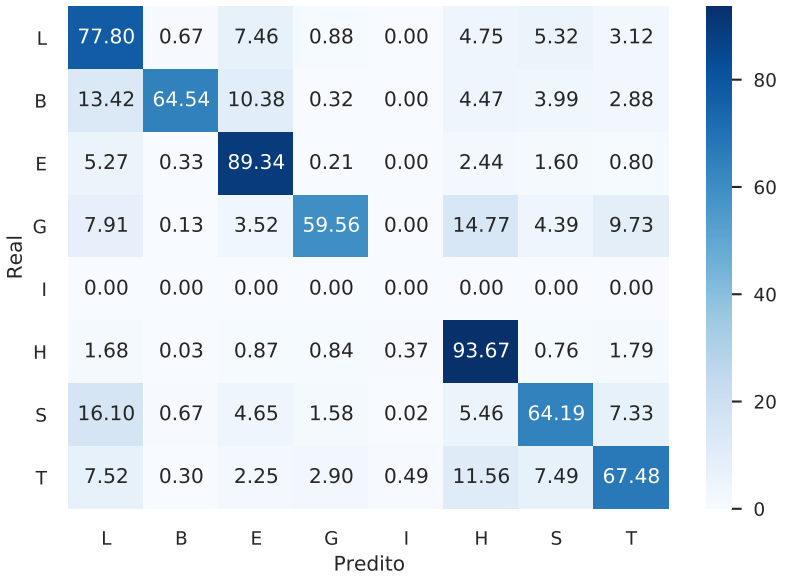


Figura 6.17: Matriz de confusão da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB6133.

Método	Acurácia Q8 (%)
Fusão dos Métodos Baseados em Modelo	**89,16**
Fusão Horizontal	88,79
Fusão Hierárquica	88,69
Alinhamentos Gerais	86,43
Bons Alinhamentos	76,34
Fusão dos Métodos Livres de Modelo	57,81
RIR	57,68
RNN	57,58
BIv4	56,73
BERT	55,86
RF	49,64

Tabela 6.58: Resultados dos métodos individuais e das fusões dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB513.

adicionais. Os pesos de cada método na fusão horizontal são apresentado na Tabela 6.59. Dentre os pesos encontrados, os métodos baseados em modelo receberam pesos maiores comparado com os métodos livres de modelo.

A Tabela 6.60 apresenta os valores de precisão e revocação para cada uma das classes obtidos na fusão horizontal de métodos livres de modelo e baseados em modelo.

A Figura 6.18 apresenta a matriz de confusão da fusão dos métodos livres de modelo e baseados em modelo na base CB513. Assim como nos resultados da base CB6133, a diagonal principal ficou em evidência. Além disto, houveram aminoácidos da classe "I" classificados corretamente.

Classe	RNN	RF	BIv4	RIR	BERT	Bons	Gerais
B	2,819	0,630	0,008	1,017	0,430	3,503	2,429
E	0,226	0,290	0,003	0,190	0,083	2,581	2,020
G	0,475	0,857	0,033	0,219	0,233	3,536	1,683
H	0,192	0,176	0,005	0,383	0,006	3,789	1,649
I	0,293	0,609	4,986	3,703	0,021	0,399	0,208
L	0,289	0,653	0,011	0,035	0,125	2,048	1,776
S	0,235	1,241	0,004	0,213	0,197	2,467	1,660
T	0,219	1,185	0,042	0,022	0,025	2,364	1,437

Tabela 6.59: Pesos de cada classe da fusão horizontal dos métodos livres de modelo e baseados em modelo na base CB513.

Classe	Precisão	Revocação	Frequência (%)
H	0,94	0,96	30,88
E	0,92	0,95	21,25
L	0,88	0,88	21,14
T	0,83	0,78	11,81
S	0,81	0,76	9,81
G	0,76	0,76	3,69
B	0,79	0,75	1,39
I	0,02	0,17	0,03

Tabela 6.60: Precisão e revocação da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB513.

Figura 6.18: Matriz de confusão da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB513.

Base PDB

Após as análises para as bases CB6133 e CB513, avaliamos a fusão de métodos livres de modelo e baseados em modelo na base PDB, tanto para a classificação Q8 quanto para a classificação Q3.

Inicialmente, realizamos a fusão entre os dois diferentes métodos utilizando a classificação Q8. Para a fusão, utilizamos a sacola de otimizadores e consideramos a fusão hierárquica e a fusão horizontal. A Tabela 6.61 apresenta os resultados obtidos pelos métodos individuais e pelas fusões entre eles. Como não existem trabalhos que reportam resultados nesta base, não podemos comparar com a literatura.

Método	**Acurácia Q8 (%)**
Fusão Horizontal	**89,66**
Fusão Hierárquica	89,60
Fusão dos Métodos Baseados em Modelo	88,55
Alinhamentos Gerais	86,36
Fusão dos Métodos Livres de Modelo	74,76
RIR	71,98
BIv4	71,64
RF	69,75
Bons Alinhamentos	68,63
RNN	68,48
BERT	63,04

Tabela 6.61: Resultados dos métodos individuais e das fusões dos métodos livres de modelo e baseados em modelo no conjunto de teste da base PDB para a classificação Q8.

A Tabela 6.62 mostra os pesos encontrados pela sacola de otimizadores para a fusão horizontal. Novamente, pesos maiores foram dados aos classificadores baseados em modelo.

Classe	**RNN**	**RF**	**BIv4**	**RIR**	**BERT**	**Bons**	**Gerais**
B	0,985	0,330	0,327	0,079	0,179	2,432	3,607
E	0,101	0,184	0,140	0,080	0,006	2,985	2,224
G	0,518	0,545	0,162	0,258	0,023	3,371	2,076
H	0,067	0,299	0,142	0,133	0,015	3,269	1,471
I	0,053	0,064	0,269	0,052	0,556	7,658	4,200
L	0,140	0,250	0,233	0,053	0,146	3,085	1,700
S	0,087	0,217	0,002	0,169	0,186	3,607	2,368
T	0,246	0,066	0,225	0,283	0,000	2,840	2,048

Tabela 6.62: Pesos de cada classe da fusão horizontal dos métodos livres de modelo e baseados em modelo na base PDB para a classificação Q8.

A Tabela 6.63 apresenta os valores de precisão e revocação para a fusão horizontal dos métodos livres de modelo e baseados em modelo na classificação Q8 da base PDB.

Classe	Precisão	Revocação	Frequência (%)
H	0,94	0,96	33,58
L	0,89	0,88	21,23
E	0,93	0,95	19,36
T	0,82	0,79	10,88
S	0,82	0,76	8,91
G	0,80	0,80	4,11
B	0,80	0,78	1,25
I	0,79	0,83	0,66

Tabela 6.63: Precisão e revocação da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base PDB para a classificação Q8.

A matriz de confusão do método é ilustrada na Figura 6.19. Pelos valores, a maioria dos dados foi classificada corretamente, sendo que apenas nas classes "B" e "S" houveram classificações equivocadas acima de 20%.

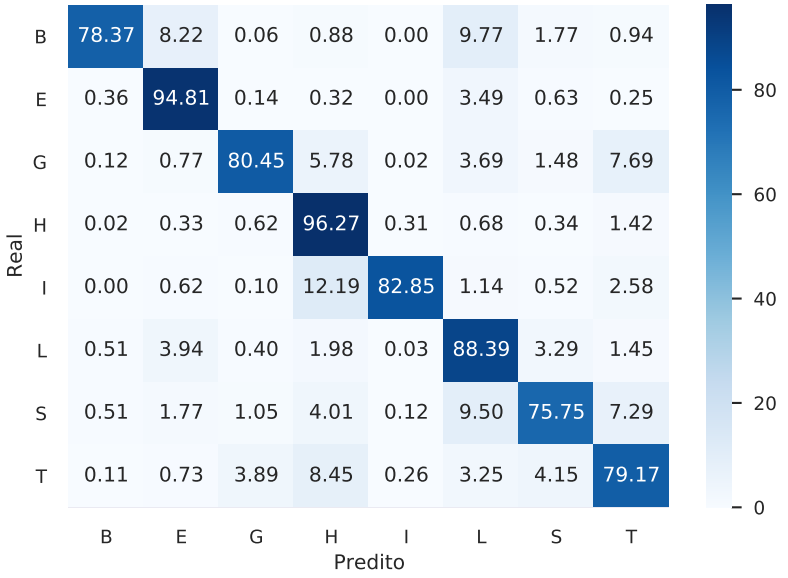


Figura 6.19: Matriz de confusão da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base PDB para a classificação Q8.

Na sequência da análise da classificação Q8, verificamos a classificação Q3. A Tabela 6.64 apresenta os resultados individuais dos métodos e das fusões entre eles. Como não existem trabalhos que reportam a acurácia Q3 nesta base, não podemos comparar os resultados com a literatura.

A Tabela 6.65 apresenta os pesos encontrados na fusão hierárquica. Dentre os valores encontrados pela sacola de otimizadores, pesos maiores foram dados para a fusão de métodos baseados em modelo.

Em seguida, avaliamos os valores de precisão e revocação por classe, conforme mostra a Tabela 6.66.

Método	Acurácia Q3 (%)
Fusão Hierárquica	**93,74**
Fusão Horizontal	93,72
Fusão dos Métodos Baseados em Modelo	93,21
Alinhamentos Gerais	91,74
Fusão dos Métodos Livres de Modelo	83,03
Bons Alinhamentos	82,11
RIR	81,91
RNN	79,22
RF	78,65
BIv4	78,26
BERT	75,68

Tabela 6.64: Resultados dos métodos individuais e das fusões dos métodos livres de modelo e baseados em modelo no conjunto de teste da base PDB para a classificação Q3.

Classe	Baseado em modelo	Livre de modelo
C	2,187	0,042
E	2,219	0,079
H	1,386	0,084

Tabela 6.65: Pesos de cada classe da fusão hierárquica dos métodos livres de modelo e baseados em modelo na base PDB para a classificação Q3.

Classe	Precisão	Revocação	Frequência (%)
C	0,94	0,91	41,69
H	0,94	0,95	37,70
E	0,93	0,96	20,61

Tabela 6.66: Precisão e revocação da fusão hierárquica dos métodos livres de modelo e baseados em modelo no conjunto de teste da base PDB para a classificação Q3.

Por fim, verificamos a matriz de confusão da fusão, ilustrada na Figura 6.20. Em todas as classes, a porcentagem de verdadeiros positivos ficou acima de 90%.

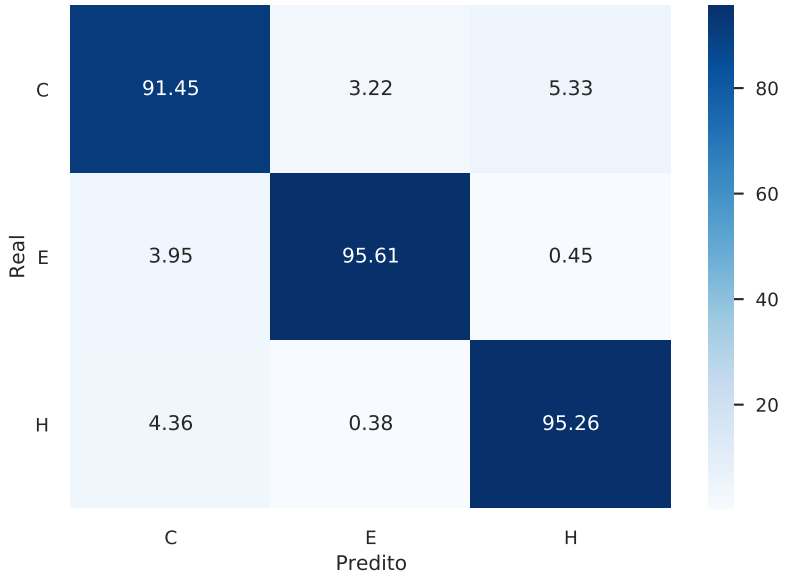


Figura 6.20: Matriz de confusão da fusão hierárquica dos métodos livres de modelo e baseados em modelo no conjunto de teste da base PDB para a classificação Q3.

Capítulo 7

Resultados Experimentais utilizando Sequência de Aminoácidos e Matriz de Pontuação de Posição Específica

Neste capítulo, apresentamos os resultados experimentais utilizando sequência de aminoácidos e matriz de pontuação de posição específica para os métodos livres de modelos, assim como a fusão de métodos livres de modelo e baseados em modelo.

7.1 Métodos Livres de Modelos

Nesta seção, apesentamos e discutimos os resultados experimentais utilizando sequência de aminoácidos e matriz de pontuação de posição específica para os classificadores RNN, RF, BIv4 e RIR, assim como a fusão de todos os métodos livres de modelo.

7.1.1 Redes Neurais Bidirecionais Recorrentes

Nesta subseção, apresentamos os resultados do classificador RNN, iniciando da escolha da arquitetura no conjunto de validação na base CB6133 e finalizando na avaliação da arquitetura nos conjuntos de teste das bases CB6133 e CB513.

Escolha da Arquitetura

Para a escolha das configurações das redes bidirecionais recorrentes, treinamos e validamos cada uma das configurações na base CB6133. Para avaliar os resultados, utilizamos a acurácia Q8.

Inicialmente, utilizamos a rede com uma camada bidirecional recorrente e 100 neurônios para avaliarmos a utilização de uma rede recorrente analisando a sequência de aminoácidos e informações da matriz de pontuação de posição específica no sentido original, ou seja, no sentido encontrado nos arquivos da base de dados, e a fusão de uma rede analisando o sentido original e outra analisando o sentido invertido. A Tabela 7.1 apresenta o resultado das duas configurações, onde a fusão das redes obteve melhores resultados. Com isso, utilizamos esta mesma configuração como base para os próximos experimentos.

Rede	Acurácia Q8 (%)
Rede analisando sentido original	70,73
Fusão de redes analisando ambos os sentidos	**71,68**

Tabela 7.1: Comparação dos resultados das RNNs analisando os dados no sentido original e a fusão das redes analisando ambos os sentidos no conjunto de validação da base CB6133.

Como próxima etapa na escolha das arquiteturas, avaliamos variações da quantidade de camadas, entre 1 a 10, mantendo 100 neurônios por camada. A Tabela 7.2 apresenta os resultados da arquitetura com diferentes quantidade de camadas bidirecionais recorrentes. As redes com 2 até 6 camadas obtiveram os melhores resultados.

Camadas	Acurácia Q8 (%)
1	71,68
2	**73,46**
3	**73,68**
4	**74,07**
5	**74,05**
6	**73,74**
7	73,37
8	73,18
9	73,07
10	72,86

Tabela 7.2: Comparação dos resultados das RNNs com diferentes números de camadas no conjunto de validação da base CB6133.

Após a escolha da quantidade de camadas para as cinco configurações diferentes das redes, verificamos a quantidade de neurônios por camada. Para isto, realizamos a fusão das cinco variações de redes recorrentes e avaliamos a quantidade de neurônios por camada, selecionando valores entre 100 e 1000, de 100 em 100. Para a fusão, utilizamos a sacola de otimizadores. A Tabela 7.3 apresenta os resultados da fusão de cada uma das configurações de neurônios por camada, com o melhor resultado em destaque. Como melhor configuração, escolhemos 600 neurônios por camada.

Após a escolha da melhor configuração para as cinco variações da arquitetura, avaliamos a utilização da camada de *embedding* para a sequência de aminoácidos. A Tabela 7.4 apresenta os resultados da fusão de redes com e sem *embedding*, com melhor resultado em destaque. Com os resultados obtidos, optamos por utilizar as redes com 2 até 6 camadas, 600 neurônios por camada e uma camada de *embedding* para a sequência de aminoácidos.

Base CB6133

A partir da arquitetura selecionada, treinamos e testamos os modelos na base CB6133. O conjunto de teste da base CB6133 contém 272 proteínas e não existem amostras da classe "I". A Tabela 7.5 apresenta os resultados de acurácia Q8 de cada uma das configurações de

Camadas	Acurácia Q8 (%)
100	75,49
200	76,55
300	76,94
400	76,95
500	77,10
600	**77,16**
700	77,08
800	77,00
900	77,12
1000	76,53

Tabela 7.3: Comparação dos resultados das RNNs com diferentes números de neurônios por camada no conjunto de validação da base CB6133.

Rede	Acurácia Q8 (%)
Com *embedding*	**77,27**
Sem *embedding*	77,16

Tabela 7.4: Comparação dos resultados da fusão de RNNs com e sem *embedding* no conjunto de validação da base CB6133.

rede no conjunto de teste. Pelos resultados obtidos pelas cinco diferentes configurações, a rede com 3 camadas obteve o melhor resultado, com 74,43% de acurácia Q8.

Camadas	Acurácia Q8 (%)
2	73,66
3	74,43
4	74,02
5	74,02
6	74,01
Fusão	**76,09**

Tabela 7.5: Resultados das RNNs no conjunto de teste da base CB6133.

Com os resultados das redes separadas, realizamos a fusão entre elas utilizando o método da Seção 5.5. Como resultado da fusão, obtivemos o valor de acurácia Q8 igual a 76,09%.

Base CB513

Na sequência, treinamos os modelos na base filtrada CB6133 e testamos na base CB513. O conjunto de teste possui 514 proteínas. A Tabela 7.6 apresenta os resultados de acurácia Q8 de cada uma das configurações de rede no conjunto de teste. A rede com 4 camadas obteve o melhor resultado, com 69,30% de acurácia Q8.

Camadas	Acurácia Q8 (%)
2	67,14
3	65,67
4	69,30
5	67,88
6	68,81
Fusão	**69,88**

Tabela 7.6: Resultados das RNNs no conjunto de teste da base CB513.

Após avaliarmos os resultados das redes individualmente, realizamos a fusão entre as redes utilizando a sacola de otimizadores. A fusão resultou no valor de acurácia Q8 igual a 69,88%.

7.1.2 Florestas Aleatórias

Nesta subseção, apresentamos os resultados do classificador florestas aleatórias, iniciando da escolha das janelas deslizantes no conjunto de validação na base CB6133 e finalizando na avaliação dos classificadores nos conjuntos de teste das bases CB6133 e CB513.

Escolha das Janelas

Para a escolha das janelas de análise das florestas aleatórias, treinamos e validamos cada uma das configurações na base CB6133. Para avaliar os resultados, utilizamos a acurácia Q8.

Como florestas aleatórias utilizam janelas deslizantes para realizar a classificação e não existe um consenso entre os trabalhos na literatura sobre o tamanhos ótimo de janela, analisamos diversos tamanho e escolhemos os cinco melhores para compor a fusão de RFs. Em relação aos tamanhos das janelas, avaliamos os tamanhos entre 3 a 21, variando de 2 em 2. A Tabela 7.7 apresenta os resultados das janelas no conjunto de validação.

Janela	Acurácia Q8 (%)
3	63,09
5	66,70
7	67,83
9	**68,22**
11	**68,34**
13	**68,48**
15	**68,09**
17	**68,00**
19	67,78
21	67,70

Tabela 7.7: Comparação dos resultados das RFs com diferentes janelas deslizantes no conjunto de validação da base CB6133.

Com os resultados obtidos, optamos por utilizar florestas aleatórias com janelas de tamanho entre 9 e 17 aminoácidos.

Base CB6133

Com as janelas deslizantes selecionadas, treinamos e testamos os classificadores na base CB6133. O conjunto de teste da base CB6133 contém 272 proteínas e não existem amostras da classe "I". A Tabela 7.8 mostra a acurácia Q8 de cada um dos classificadores no conjunto de teste.

Janela	**Acurácia Q8 (%)**
9	67,21
11	67,29
13	67,58
15	67,33
17	67,26
Fusão	**69,30**

Tabela 7.8: Resultados das RFs no conjunto de teste da base CB6133.

Após os resultados de cada classificador, realizamos a fusão das RFs utilizando o método descrito na Seção 5.5. Com a fusão, atingimos 69,30% de acurácia Q8, superando os resultados dos classificadores individuais.

Base CB513

Após os testes na base CB6133, treinamos os classificadores na base CB6133 filtrada e testamos na base CB513. A Tabela 7.9 apresenta os resultados de acurácia Q8 dos classificadores no conjunto de teste.

Janela	**Acurácia Q8 (%)**
9	61,86
11	61,93
13	61,94
15	61,88
17	61,82
Fusão	**64,60**

Tabela 7.9: Resultados das RFs no conjunto de teste da base CB513.

Após a análise de cada um dos classificadores, realizamos a fusão entre os diferentes classificadores usando a técnica de fusão apresentada na Seção 5.5. Ao final da fusão, obtivemos 64,60% de acurácia Q8.

7.1.3 Blocos Inception-v4

Nesta subseção, apresentamos os resultados do classificador blocos Inception-v4 (BIv4), iniciando da escolha dos modelos no conjunto de validação na base CB6133 e finalizando na avaliação dos modelos selecionados nos conjuntos de teste das bases CB6133 e CB513.

Escolha dos Modelos

Para a escolha dos modelos para a predição de estruturas secundárias, testamos os três diferentes blocos que compõem a arquitetura Inception-v4, chamados de blocos A, B e C, além de variações na quantidade de blocos de um mesmo tipo, entre 1 e 10. Com os resultados com camada de *embedding* do classificador de redes recorrentes, adotamos inicialmente a mesma técnica para este classificador.

A Tabela 7.10 apresenta os resultados de acurácia Q8 dos diferentes blocos no conjunto de validação da base CB6133. Com os resultados obtidos, optamos pelos modelos com blocos do tipo B e com 3 a 7 blocos.

	Acurácia Q8 (%)		
Quantidade	**Blocos A**	**Blocos B**	**Blocos C**
1	66,50	71,79	66,77
2	70,15	72,20	71,66
3	71,61	**73,22**	72,65
4	72,27	**73,56**	73,22
5	72,38	**73,73**	73,35
6	72,82	**73,88**	73,54
7	72,33	**73,70**	73,56
8	72,55	72,94	73,50
9	72,70	73,19	73,49
10	72,73	72,87	73,46

Tabela 7.10: Comparação dos resultados com diferentes blocos Inception-v4 e quantidades de blocos no conjunto de validação da base CB6133.

Após a seleção dos cinco modelos, comparamos a fusão com os métodos com camada de *embedding* e verificamos o resultado comparado com a fusão dos mesmos modelos sem a camada de *embedding*. Os resultados apresentados na Tabela 7.11 mostram que a utilização da camada de *embedding* aumentou o resultado de acurácia Q8, assim como no classificador RNN.

Base CB6133

Com os cinco modelos selecionados, treinamos e testamos todos eles no conjunto de dados CB6133. A Tabela 7.12 apresenta os resultados de cada um dos modelos em relação a acurácia Q8. Os melhores resultados foram obtidos com classificadores com 6 e 7 blocos, atingindo 73,74 e 73,91% de acurácia Q8, respectivamente.

Em seguida, realizamos a fusão dos modelos utilizando a sacola de otimizadores. Como resultado, a fusão atingiu 75,42% de acurácia Q8.

Fusão	Acurácia Q8 (%)
Com *embedding*	**75,99**
Sem *embedding*	75,17

Tabela 7.11: Comparação dos resultados da fusão de BIv4 com e sem *embedding* no conjunto de validação da base CB6133.

Blocos	Acurácia Q8 (%)
3	73,52
4	73,01
5	73,45
6	73,74
7	73,91
Fusão	**75,42**

Tabela 7.12: Resultados dos BIv4 no conjunto de teste da base CB6133.

Base CB513

Após os testes na base CB6133, treinamos os modelos no conjunto CB6133 filtrado e testamos na base CB513. Os resultados de acurácia Q8 dos modelos são mostrados na Tabela 7.13. Assim como na base CB6133, os modelos que obtiveram os melhores valores de acurácia Q8 foram com 6 e 7 blocos do tipo B empilhados.

Blocos	Acurácia Q8 (%)
3	69,09
4	68,78
5	68,93
6	69,10
7	69,63
Fusão	**70,93**

Tabela 7.13: Resultados dos BIv4 no conjunto de teste da base CB513.

Na sequência, avaliamos a fusão dos cinco diferentes modelos utilizando a técnica de fusão apresentada na Seção 5.5. Como resultado, a fusão atingiu 70,93% de acurácia Q8, obtendo resultados melhores que os modelos separados.

7.1.4 Rede Inception Recorrente

Nesta subseção, apresentamos os resultados dos classificadores Rede Inception Recorrente (RIR), iniciando na escolha das arquiteturas no conjunto de validação da base CB6133 e finalizando na avaliação das arquiteturas selecionadas nos conjuntos de teste das bases CB6133 e CB513.

Escolha da Arquitetura

Para os classificadores RIR, avaliamos no conjunto de validação da base CB6133 as cinco variações utilizadas nos classificadores blocos Inception-v4 com diferentes configurações de camadas recorrentes ao final dos blocos de convolução.

Avaliamos configurações de rede de 1 até 5 camadas e de 100 até 500 neurônios por camada. A Tabela 7.14 apresenta os cinco melhores resultados da fusão das redes testadas.

Camadas	Neurônios	Acurácia Q8 (%)
3	**100**	**74,19**
2	200	74,10
4	100	73,77
3	300	73,50
3	200	73,43

Tabela 7.14: Comparação das variações da arquitetura RIR no conjunto de validação da base CB6133.

Para experimentos adicionais, escolhemos as arquiteturas com 3 camadas bidirecionais recorrentes e com 100 neurônios por camada.

Base CB6133

Após selecionarmos a arquitetura das redes, treinamos e testamos os modelos na base de dados CB6133. A Tabela 7.15 mostra os resultados de acurácia Q8 de cada um dos modelos. Os melhores resultados foram obtidos com classificadores com 6, atingindo 74,07% de acurácia Q8.

Blocos	Acurácia Q8 (%)
3	73,61
4	73,94
5	73,77
6	74,07
7	73,84
Fusão	**75,29**

Tabela 7.15: Resultados das redes RIR no conjunto de teste da base CB6133.

Na sequência, realizamos a fusão das cinco diferentes redes utilizando a sacola de otimizadores. Como resultado da fusão, obtivemos 75,29% de acurácia Q8.

Base CB513

Além dos experimentos na base CB6133, treinamos o modelo na versão filtrada da base CB6133 e realizamos o teste na base CB513. O melhor resultado foi obtido com o classificador com 7 blocos, atingindo 69,78% de acurácia Q8, conforme apresenta a Tabela 7.16.

Blocos	Acurácia Q8 (%)
3	68,55
4	69,52
5	69,36
6	69,57
7	69,78
Fusão	**71,16**

Tabela 7.16: Resultados das redes RIR no conjunto de teste da base CB513.

Após a análise dos classificadores separadamente, realizamos a fusão entre os modelos seguindo a técnica de fusão apresentada na Seção 5.5. Como resultado, a fusão atingiu 71,16% de acurácia Q8, ultrapassando os resultados obtidos pelas redes individuais.

7.1.5 Fusão dos Métodos Livres de Modelo

Após a análise individual dos classificadores livres de modelo, realizamos experimentos para a fusão entre os quatro classificadores que utilizam informações de matriz de pontuação de posição específica (RNN, RF, BIv4 e RIR) e do BERT, que usa apenas as informações da sequência de aminoácidos, tanto para a base CB6133 quanto para a base CB513.

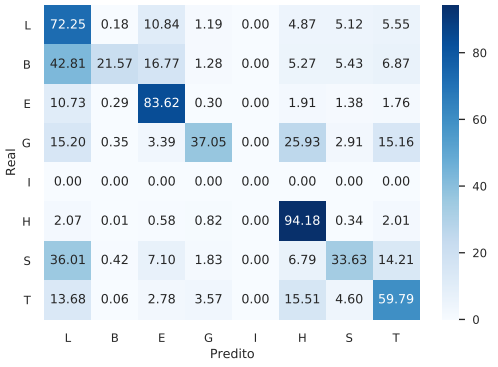
Base CB6133

Antes de realizar a fusão, comparamos as matrizes de confusão dos cinco métodos para avaliarmos quais classificadores são bons em quais classes. A Figura 7.1 apresenta a matriz de confusão no conjunto de teste dos cinco métodos. Pelos valores, é possível perceber que o classificador RNN é o melhor em classificação das classes "B" e "G", enquanto o classificador RIR é melhor na classe "S". Dentre as classes majoritárias, como "H", "E" e "L", os modelos RNN, BIv4 e RIR atingiram bons resultados nas classes em estruturas que aparecem em sequências maiores, como as classes "H" e "E", conforme apresentado na Tabela 4.1. Dentre os modelos, os piores resultados obtidos foram com o BERT, já que o modelo utiliza apenas a informação de sequência de aminoácidos.

Após a análise qualitativa dos resultados de cada classificador, realizamos a fusão entre os cinco diferentes métodos. O resultado obtido pela fusão dos métodos livres de modelo é apresentado na Tabela 7.17, revelando que a fusão obteve resultado superior comparado com os métodos individuais.

A Tabela 7.18 mostra os pesos encontrados pelo método de fusão para cada classe de cada classificador. Mesmo não atingindo a maior acurácia Q8, o classificador RF possui valores altos nos pesos das suas predições, mostrando que possui grande influência no resultado. Em contrapartida, o classificador BIv4 obteve os menores pesos, indicando que não é tão influente na fusão.

Após a análise qualitativa da fusão, comparamos o nosso método com os trabalhos da literatura, conforme apresenta a Tabela 7.19. Não comparamos os resultados com o trabalho de Kumar *et al.* [45], pois utiliza a divisão da base de modo diferente. Os

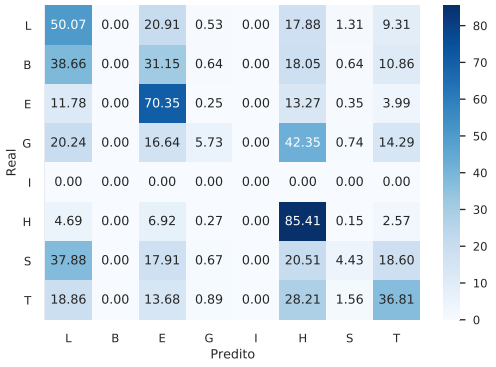


(a) RNN.

(b) RF.

(c) BIv4.

(d) RIR.



(e) BERT.

Figura 7.1: Matrizes de confusão dos métodos livres de modelo no conjunto de teste da base CB6133.

valores apontam que nosso método possui resultados competitivos, com cerca de 0,2 pontos percentuais atrás do estado da arte.

Dentre os trabalhos da literatura, é comum apresentar a precisão e a revocação de cada classe, e comparar com o estado da arte. Como o estado da arte [66] não apresentou esta análise, a Tabela 7.20 apresenta apenas os valores de precisão e revocação do nosso método.

A Figura 7.2 apresenta a matriz de confusão da fusão dos métodos livres de modelo.

Método	Acurácia Q8 (%)
Fusão	**76,67**
RNN	76,09
BIv4	75,42
RIR	75,29
RF	69,30
BERT	59,60

Tabela 7.17: Resultados dos métodos individuais e da fusão dos métodos livres de modelo no conjunto de teste da base CB6133.

Classe	RNN	RF	BIv4	RIR	BERT
B	2,249	1,744	0,481	0,996	1,927
E	2,031	1,293	0,073	0,798	0,011
G	1,216	1,270	0,729	1,316	1,728
H	1,159	0,866	0,412	1,512	0,004
I	0,227	0,156	0,215	0,040	1,547
L	1,569	3,343	0,153	0,385	0,063
S	1,484	1,902	0,491	1,281	0,432
T	1,482	1,608	0,271	0,734	0,784

Tabela 7.18: Pesos de cada classe de cada classificador na fusão dos métodos livres de modelo na base CB6133.

Método	Acurácia Q8 (%)
Ratul *et al.* [66]	76,9
Nosso método	**76,7**
Drori *et al.* [20]	76,3
Gou *et al.* [26]	75,7
Johansen *et al.* [36]	74,8
Guo *et al.* [25]	74,2
Zhou *et al.* [96]	74,0
Zhou e Troyanskaya [95]	72,1

Tabela 7.19: Comparação do resultado obtido com outros trabalhos da literatura no conjunto de teste da base CB6133.

Pelos valores da matriz de confusão, é possível perceber que as classes majoritárias, como "H" e "E", atingiram altos valores de verdadeiros positivos, enquanto classes minoritárias foram prejudicadas. Além disto, classes que pertencem ao mesmo conjunto na classificação Q3 foram confundidas mais frequentemente, como dados da classe "S", que foram classificados como "L" e "T", e da classe "G", que foram classificados como "H". Outro fato importante é a classificação de estruturas da classe "B" como "L".

Classe	Precisão	Revocação	Frequência (%)
H	0,89	0,94	35,67
E	0,83	0,85	21,56
L	0,62	0,71	18,57
T	0,64	0,62	11,11
S	0,58	0,36	7,92
G	0,58	0,41	4,06
B	0,71	0,21	1,10
I	—	—	0,00

Tabela 7.20: Precisão e revocação da fusão dos métodos livres de modelo no conjunto de teste da base CB6133.

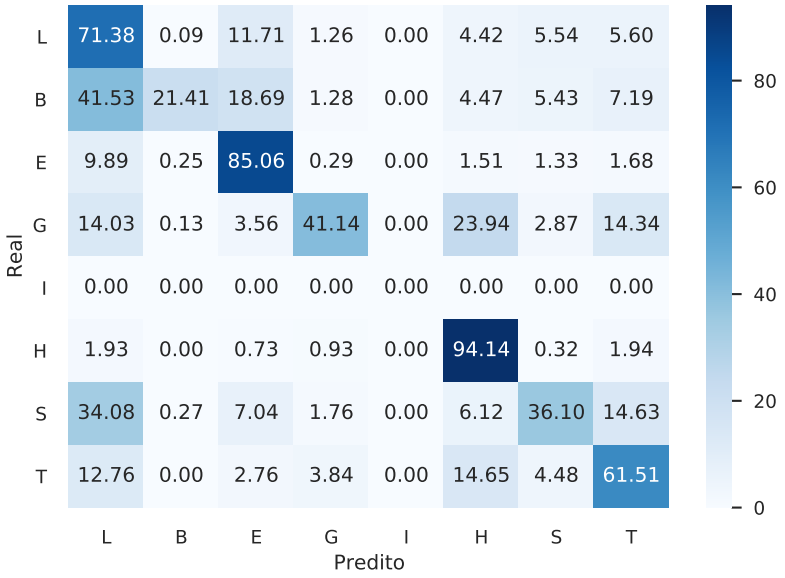


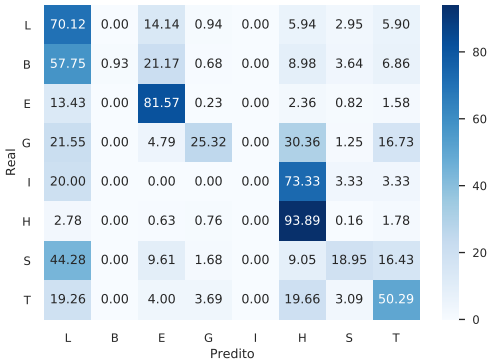
Figura 7.2: Matriz de confusão da fusão dos métodos livres de modelo no conjunto de teste da base CB6133.

Base CB513

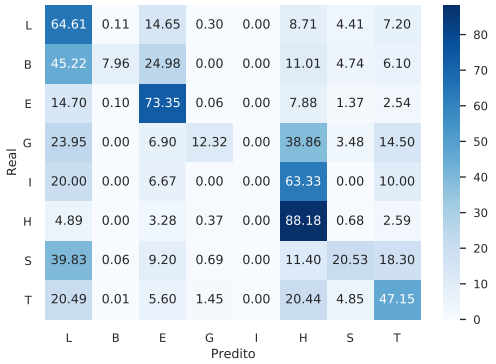
Assim como na base CB6133, comparamos as matriz de confusão dos cinco métodos para avaliarmos quais classificadores são bons em quais classes na base CB513. A Figura 7.3 apresenta a matriz de confusão dos cinco métodos. Os resultados mostram que o classificador RF possui melhores resultados na classe "B" e que os modelos RNN, BIv4 e RIR atingem bons resultados nas classes majoritárias, como "H" e "E". Nenhum dos classificadores foi capaz de predizer corretamente os dados da classe "I" do conjunto de teste.

Na sequência, realizamos fusões entre os cinco diferentes métodos. Os resultados obtidos pelos métodos e pela fusão são apresentados na Tabela 7.21.

A Tabela 7.22 mostra os pesos encontrados pelo método de fusão para cada classe de cada classificador. Dentre os pesos encontrados, os classificadores RNN, RF e BIv4 obtiveram os maiores pesos.



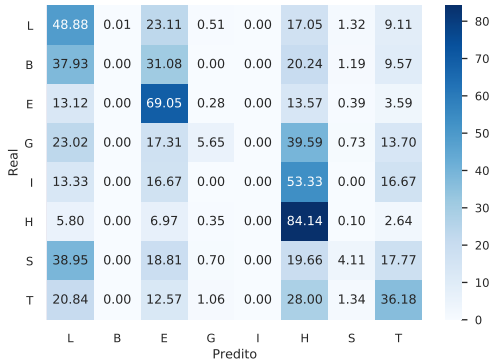
(a) RNN.



(b) RF.

(c) BIv4.

(d) RIR.



(e) BERT.

Figura 7.3: Matrizes de confusão dos métodos livres de modelo no conjunto de teste da base CB513.

Após a verificação dos pesos da fusão, comparamos o nosso método com os trabalhos da literatura, conforme exibido na Tabela 7.23. Os resultados mostram que nosso método possui resultados competitivos comparado com os resultados disponíveis na literatura, atrás do estado da arte por 0,4 pontos percentuais.

Além da métrica acurácia Q8, os métodos da literatura geralmente disponibilizam a precisão e revocação por classe. Como o estado da arte não avaliou as classes em relação a precisão e revocação, a Tabela 7.24 apresenta apenas os valores de precisão e revocação

Método	Acurácia Q8 (%)
Fusão	**71,48**
RIR	71,16
BIv4	70,93
RNN	69,88
RF	64,60
BERT	55,86

Tabela 7.21: Resultados dos métodos individuais e da fusão dos métodos livres de modelo no conjunto de teste da base CB513.

Classe	RNN	RF	BIv4	RIR	BERT
B	0,399	1,049	0,641	0,533	0,721
E	0,415	0,811	0,892	0,190	0,081
G	0,811	0,168	0,508	0,163	0,558
H	0,624	0,035	0,231	0,673	0,017
I	0,491	0,313	0,800	0,149	0,988
L	0,574	0,487	0,796	0,140	0,028
S	0,860	0,703	0,445	0,226	0,104
T	0,484	0,975	0,388	0,228	0,243

Tabela 7.22: Pesos de cada classe de cada classificador na fusão dos métodos livres de modelo na base CB513.

Método	Acurácia Q8 (%)
Ratul *et al.* [66]	71,9
Nosso método	**71,5**
Busia *et al.* [10]	71,4
Uddin *et al.* [82]	70,9
Johansen *et al.* [36]	70,9
Drori *et al.* [20]	70,7
Fang *et al.* [22]	70,6
Zhou *et al.* [96]	70,3
Gou *et al.* [26]	70,2
Li e Yu [48]	69,4
Lin *et al.* [49]	68,4
Wang *et al.* [87]	68,2
Sønderby e Winther [75]	67,4
Zhou e Troyanskaya [95]	66,4
Hattori *et al.* [29]	66,0

Tabela 7.23: Comparação do resultado obtido com outros trabalhos da literatura no conjunto de teste da base CB513.

do nosso método.

A Figura 7.4 mostra a matriz de confusão do nosso método no conjunto de teste da

Classe	Precisão	Revocação	Frequência (%)
H	0,85	0,93	30,88
E	0,78	0,83	21,25
L	0,57	0,71	21,14
T	0,58	0,54	11,81
S	0,58	0,25	9,81
G	0,49	0,34	3,69
B	0,88	0,04	1,39
I	0,00	0,00	0,03

Tabela 7.24: Precisão e revocação da fusão dos métodos livres de modelo no conjunto de teste da base CB513.

base CB513. Pelos valores da matriz de confusão, é possível perceber que as classes majoritárias, como "H" e "E", atingiram valores de verdadeiros positivos altos, enquanto classes minoritárias foram prejudicadas. As classes que pertencem ao mesmo conjunto na classificação Q3 foram confundidas mais frequentemente, como dados da classe "S", classificados como "L" e "T", e da classe "G", que foram classificados como "H".

Figura 7.4: Matriz de confusão da fusão dos métodos livres de modelo no conjunto de teste da base CB513.

7.2 Fusão dos Métodos Livres de Modelo e Métodos Baseados em Modelo

Nesta seção, apresentamos e discutimos os resultados experimentais da fusão dos métodos livres de modelo e baseados em modelo nas bases CB6133 e CB513.

Base CB6133

Com os resultados da fusão de métodos livres de modelo e métodos baseados em modelo, realizamos a fusão entre as duas abordagens. Como métodos de fusão, consideramos dois cenários, fusão hierárquica, ou seja, a fusão entre as fusões de métodos livres de modelo e métodos baseados em modelo, e a fusão horizontal, considerando os cinco métodos livres de modelo e os dois métodos baseados em modelo individualmente. A Tabela 7.25 apresenta os resultados obtidos pelos dois métodos de fusão, sendo que a fusão horizontal obteve os melhores valores de acurácia Q8.

Método	**Acurácia Q8 (%)**
Fusão Horizontal	**85,08**
Fusão Hierárquica	84,86
Fusão dos Métodos Baseados em Modelo	78,73
Fusão dos Métodos Livres de Modelo	76,67
RNN	76,09
Alinhamentos Gerais	75,96
BIv4	75,42
RIR	75,29
RF	69,30
Bons Alinhamentos	69,15
BERT	59,60

Tabela 7.25: Resultados dos métodos individuais e das fusões dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB6133.

A Tabela 7.26 apresenta os pesos da fusão horizontal. Os valores mostram que os maiores pesos foram dados para as predições dos métodos baseados em modelo.

Classe	**RNN**	**RF**	**BIv4**	**RIR**	**BERT**	**Bons**	**Gerais**
B	0,002	1,853	0,326	0,886	0,002	2,937	2,908
E	0,430	0,442	0,280	0,092	0,012	2,806	2,203
G	0,203	1,585	0,137	0,302	0,198	2,981	2,416
H	0,246	0,370	0,242	0,385	0,012	4,013	1,631
I	0,218	0,971	1,368	0,480	0,131	1,958	0,313
L	0,106	1,626	0,231	0,195	0,088	2,639	1,549
S	0,254	1,487	0,243	0,028	0,903	2,668	1,809
T	0,817	0,785	0,189	0,027	0,001	2,962	2,019

Tabela 7.26: Pesos de cada classe da fusão horizontal dos métodos livres de modelo e baseados em modelo na base CB6133.

Após a análise qualitativa da fusão, comparamos o resultado obtido com os métodos disponíveis na literatura, conforme apresentado na Tabela 7.27. Como resultado, a fusão dos métodos ultrapassou o estado da arte em cerca de 8,2 pontos percentuais.

Além da análise de acurácia Q8, avaliamos o método em relação à precisão e revocação, conforme mostra a Tabela 7.28. Os valores de precisão e revocação das classes minoritárias

Método	Acurácia Q8 (%)
Nosso método	**85,1**
Ratul *et al.* [66]	76,9
Drori *et al.* [20]	76,3
Gou *et al.* [26]	75,7
Johansen *et al.* [36]	74,8
Guo *et al.* [25]	74,2
Zhou *et al.* [96]	74,0
Zhou e Troyanskaya [95]	72,1

Tabela 7.27: Comparação do resultado obtido com outros trabalhos da literatura no conjunto de teste da base CB6133.

aumentaram comparado com os métodos livres de modelo, muito por conta da fusão com os métodos baseados em modelo.

Classe	Precisão	Revocação	Frequência (%)
H	0,92	0,95	35,67
E	0,91	0,91	21,56
L	0,79	0,80	18,57
T	0,76	0,73	11,11
S	0,69	0,66	7,92
G	0,74	0,62	4,06
B	0,75	0,62	1,10
I	—	—	0,00

Tabela 7.28: Precisão e revocação da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB6133.

A Figura 7.5 apresenta a matriz de confusão da fusão horizontal dos métodos livres de modelo e baseados em modelo na base CB6133. Os resultados estão próximos da diagonal principal, com maiores variações na classificação das classes minoritárias.

Base CB513

Após a análise dos modelos na base CB6133, realizamos a fusão hierárquica e fusão horizontal entre as duas abordagens utilizando a sacola de otimizadores. A Tabela 7.29 apresenta os resultados obtidos pelos dois métodos de fusão, sendo que a fusão horizontal obteve 89,55% de acurácia Q8.

A Tabela 7.30 apresenta os pesos de cada método na fusão horizontal. Os valores mostram que os maiores pesos foram dados às predições dadas pelos métodos baseados em modelo.

Na sequência da análise qualitativa da fusão, comparamos o resultado obtido com os métodos disponíveis na literatura, conforme apresentado na Tabela 7.31. Como resultado, a fusão dos métodos ultrapassou o estado da arte em 17,6 pontos percentuais.

Figura 7.5: Matriz de confusão da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB6133.

Método	Acurácia Q8 (%)
Fusão Horizontal	**89,55**
Fusão dos Métodos Baseados em Modelo	89,16
Fusão Hierárquica	89,14
Alinhamentos Gerais	86,43
Bons Alinhamentos	76,34
Fusão dos Métodos Livres de Modelo	71,48
RIR	71,16
BIv4	70,93
RNN	69,88
RF	64,60
BERT	55,86

Tabela 7.29: Resultados dos métodos individuais e das fusões dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB513.

Além da análise de acurácia Q8, avaliamos o método em relação à precisão e revocação, conforme mostra a Tabela 7.32. Os valores de precisão e revocação das classes minoritárias aumentaram comparado com os métodos livres de modelo, muito por conta da fusão com os métodos baseados em modelo.

A Figura 7.6 apresenta a matriz de confusão da fusão horizontal dos métodos livres de modelo e baseados em modelo na base CB513. Assim como na base CB6133, os resultados estão próximos da diagonal principal, com maiores variações na classificação das classes minoritárias. Além disto, a classe "I" obteve predições corretas.

Classe	RNN	RF	BIv4	RIR	BERT	Bons	Gerais
B	0,308	0,948	0,189	0,203	0,726	5,489	2,893
E	0,183	0,710	0,001	0,311	0,078	2,783	1,724
G	0,414	1,342	0,010	0,144	0,905	3,016	2,535
H	0,359	0,628	0,041	0,332	0,015	4,145	1,364
I	0,345	0,295	0,125	0,187	1,041	1,136	1,102
L	0,365	1,102	0,055	0,276	0,034	3,220	1,126
S	0,307	1,692	0,078	0,080	0,250	2,862	1,765
T	0,502	1,383	0,017	0,113	0,046	2,959	1,718

Tabela 7.30: Pesos de cada classe da fusão horizontal dos métodos livres de modelo e baseados em modelo na base CB513.

Método	Acurácia Q8 (%)
Nosso método	**89,5**
Ratul *et al.* [66]	71,9
Busia *et al.* [10]	71,4
Uddin *et al.* [82]	70,9
Johansen *et al.* [36]	70,9
Drori *et al.* [20]	70,7
Fang *et al.* [22]	70,6
Zhou *et al.* [96]	70,3
Gou *et al.* [26]	70,2
Li e Yu [48]	69,4
Lin *et al.* [49]	68,4
Wang *et al.* [87]	68,2
Sønderby e Winther [75]	67,4
Zhou e Troyanskaya [95]	66,4
Hattori *et al.* [29]	66,0

Tabela 7.31: Comparação do resultado obtido com outros trabalhos da literatura no conjunto de teste da base CB513.

Classe	Precisão	Revocação	Frequência (%)
H	0,94	0,96	30,88
E	0,94	0,95	21,25
L	0,88	0,88	21,14
T	0,82	0,80	11,81
S	0,82	0,76	9,81
G	0,80	0,79	3,69
B	0,82	0,73	1,39
I	0,02	0,20	0,03

Tabela 7.32: Precisão e revocação da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB513.

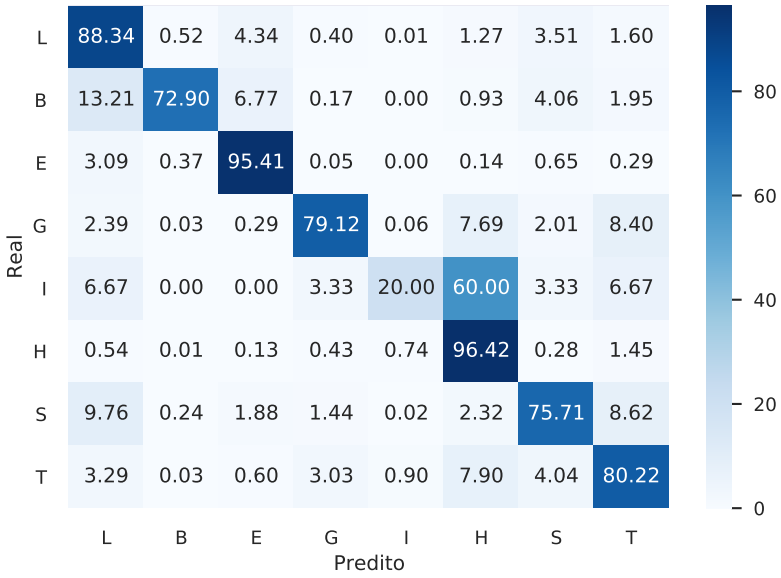


Figura 7.6: Matriz de confusão da fusão horizontal dos métodos livres de modelo e baseados em modelo no conjunto de teste da base CB513.

Capítulo 8

Conclusões e Trabalhos Futuros

A predição de estruturas secundárias de proteínas possui grande relevância na análise do enovelamento e na determinação das funções exercidas pelas proteínas. Com os avanços tecnológicos nas últimas décadas, o sequenciamento de aminoácidos, que formam as cadeias das proteínas, tornou-se rápido e barato, porém métodos mais custosos são necessários para determinar estruturas tridimensionais e funções das proteínas. Com isso, grandes esforços vêm sendo feitos para classificar estas características. Dentre as opções, a classificação partindo do enovelamento que cada aminoácido forma, ou estruturas secundárias, mostrou-se promissora.

Neste trabalho, investigamos duas abordagens para a predição de estruturas secundárias de proteínas, sendo elas métodos livres de modelo e métodos baseados em modelo. Dentre os métodos livres de modelo, utilizamos classificadores baseados em redes neurais, como redes recorrentes, redes convolucionais e *Transformers*, e baseados em algoritmos de aprendizado de máquina, como florestas aleatórias. Além disto, avaliamos o desempenho de classificadores globais, que usam a sequência inteira como entrada de dados, e classificadores locais, que usam trechos e janelas deslizantes para realizar a classificação.

Para a classificação utilizando métodos baseados em modelo, investigamos a utilização da ferramenta BLAST para buscar e encontrar proteínas similares das bases do teste com as proteínas de todo o PDB. Os nossos resultados mostraram que esta abordagem é capaz de produzir bons resultados para a classificação de estruturas secundárias. O principal ponto negativo neste tipo de classificador consiste na dependência de proteínas similares tanto no conjunto de teste quanto na base de busca.

Em relação às métricas de avaliação, a acurácia Q3 e acurácia Q8 não são as melhores escolhas para avaliar o desempenho dos modelos, já que a classificação errada de classes minoritárias não possui grande impacto no resultado final. Outras métricas, como a precisão e revocação por classe, que já são utilizadas na maioria dos trabalhos da literatura, e a acurácia balanceada para o modelo, deveriam ser amplamente divulgadas para a comparação mais justa das classes.

Ao final do desenvolvimento desta pesquisa, podemos responder às questões de pesquisa do Capítulo 1:

- Utilizar apenas a sequência de aminoácidos pode produzir resultados próximos aos resultados utilizando sequência de aminoácidos e informações evolutivas.

A utilização de informações evolutivas, como a matriz de pontuação de posição específica, junto com a sequência de aminoácidos, mostrou-se promissora para aumentar a acurácia Q8 nas bases CB6133 e CB513, comparado com os resultados utilizando apenas a sequência de aminoácidos. Porém, em grandes bases de dados, como todos os arquivos do PDB, isto não é possível, principalmente pelo tempo necessário para gerar essas características.

- A transformação do vetor *one-hot encoding* esparso em um vetor denso pode ajudar na classificação de estruturas secundárias das proteínas?

  A transformação da sequência de aminoácidos em um vetor denso através de camadas de *embedding* permitiu aumentar a acurácia final das redes neurais. Entretanto, um ponto importante nesta abordagem é a falta de significado das informações transformadas, perdendo o sentido original das características.

- Qual é o impacto da fusão de classificadores locais e globais na classificação?

  A fusão de classificadores locais e globais aumentou a acurácia resultante comparado com a predição feita individualmente pelos classificadores locais e globais. O impacto da fusão se deve ao poder de classificação de estruturas próximas dos classificadores locais com o poder de classificação de estruturas distantes dos classificadores globais. A utilização das duas vertentes de classificadores é importante, pois tanto aminoácidos próximos quanto distantes impactam a estrutura formada de um aminoácido.

- A fusão entre métodos baseados em modelo e livres de modelo pode melhorar os resultados?

  A fusão entre métodos baseados em modelo e livres de modelo, na grande maioria dos casos, melhorou os resultados, em comparação com os resultados dos métodos individuais. Pelos resultados obtidos, os métodos baseados em modelo impactaram mais na fusão, porém isso se deve às proteínas similares das bases de teste com a base de dados de busca do BLAST, o que pode não acontecer em casos gerais.

Como possíveis próximos passos desta pesquisa, podemos destacar etapas de pré-processamento e pós-processamento. Na etapa de pré-processamento, técnicas de aumentação de dados de processamento de linguagem natural, como o EDA [89], podem ser investigadas, assim como a utilização de outras características evolutivas, como perfis de modelos ocultos de Markov [67]. Já para classificadores baseados em *Transformers*, como o BERT, etapas de pré-processamento podem ser usadas para que este tipo de classificador possa utilizar as informações de matriz de pontuação de posição específica, como a discretização dos valores desta característica, e, assim, obter resultados competitivos e aumentar a acurácia Q8 e Q3 na fusão resultante entre os métodos.

Na etapa de pós-processamento, abordagens de explicabilidade de classificadores de aprendizado de máquina e aprendizado profundo podem ser utilizadas para realizar uma análise sistemática dos acertos e dos erros, podendo indicar as tomadas de decisão e melhorar os resultados obtidos por cada um dos classificadores. Ainda na etapa de pós-processamento, outros métodos de fusão podem ser investigados, como a utilização

de meta-classificadores de aprendizado de máquina e aprendizado profundo, por exemplo *Support Vector Machines* (SVM) e redes rasas, assim como a utilização de outras características além das probabilidades da saída de cada classificador, como *deep features*.

Referências Bibliográficas

[1] Bruce Alberts, Dennis Bray, Karen Hopkin, Alexsander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Essential Cell Biology*. Garland Science, Taylor and Francis Group, New York, United States of America, 2015.

[2] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[3] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Helen M Berman. The Protein Data Bank: a historical perspective. *Acta Crystallographica Section A: Foundations of Crystallography*, 64(1):88–95, 2008.

[6] Helen M Berman, Gerard J Kleywegt, Haruki Nakamura, and John L Markley. The Protein Data Bank at 40: reflecting on the past to prepare for the future. *Structure*, 20(3):391–396, 2012.

[7] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[8] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[9] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[10] Akosua Busia and Navdeep Jaitly. Next-step conditioned deep convolutional neural networks improve protein secondary structure prediction. *arXiv preprint arXiv:1702.03865*, 2017.

[11] Alessio Ceroni, Paolo Frasconi, and Gianluca Pollastri. Learning protein secondary structure from sequential and relational data. *Neural Networks*, 18(8):1029–1039, 2005.

[12] Ricardo Cerri, Rafael G Mantovani, Márcio P Basgalupp, and André CPLF de Carvalho. Multi-label feature selection techniques for hierarchical multi-label protein function prediction. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2018.

[13] Jinyong Cheng, Yihui Liu, and Yuming Ma. Protein secondary structure prediction based on integration of CNN and LSTM model. *Journal of Visual Communication and Image Representation*, page 102844, 2020.

[14] Jong Cheol Jeong, Xiaotong Lin, and Xue-Wen Chen. On position-specific scoring matrix for protein function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(2):308–315, 2010.

[15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[16] Peter Y Chou and Gerald D Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, 1974.

[17] James A Cuff and Geoffrey J Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] Iddo Drori, Isht Dwivedi, Pranav Shrestha, Jeffrey Wan, Yueqi Wang, Yunchu He, Anthony Mazza, Hugh Krogh-Freeman, Dimitri Leggas, Kendal Sandridge, Linyong Nan, Kaveri Thakoor, Chinmay Joshi, Sonam Goenka, Chen Keasar, and Itsik Peer. High quality prediction of protein Q8 secondary structure by diverse neural network architectures. *arXiv preprint arXiv:1811.07143*, 2018.

[21] Chao Fang, Yi Shang, and Dong Xu. A new deep neighbor residual network for protein secondary structure prediction. In *29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 66–71. IEEE, 2017.

[22] Chao Fang, Yi Shang, and Dong Xu. MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 86(5):592–598, 2018.

[23] Jean Garnier, David J Osguthorpe, and Barry Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*, 120(1):97–120, 1978.

[24] Kushal K Ghosh, Soulib Ghosh, Sagnik Sen, Ram Sarkar, and Ujjwal Maulik. A two-stage approach towards protein secondary structure classification. *Medical & Biological Engineering & Computing*, 58(1):1723–1737, 2020.

[25] Yanbu Guo, Weihua Li, Bingyi Wang, Huiqing Liu, and Dongming Zhou. DeepACLSTM: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction. *BMC Bioinformatics*, 20(1):341, 2019.

[26] Yanbu Guo, Bingyi Wang, Weihua Li, and Bei Yang. Protein secondary structure prediction improved by recurrent neural networks integrated with two-dimensional convolutional neural networks. *Journal of Bioinformatics and Computational Biology*, 16(05):1850021, 2018.

[27] Yuzhi Guo, Jiaxiang Wu, Hehuan Ma, Sheng Wang, and Junzhou Huang. Bagging MSA Learning: Enhancing Low-Quality PSSM with Deep Learning for Accurate Protein Structure Property Prediction. In *International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 88–103. Springer, 2020.

[28] Haris Hasic, Emir Buza, and Amila Akagic. A hybrid method for prediction of protein secondary structure based on multiple artificial neural networks. In *40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1195–1200. IEEE, 2017.

[29] Leandro T Hattori, Cesar M V Benitez, and Heitor S Lopes. A deep bidirectional long short-term memory approach applied to the protein secondary structure prediction problem. In *4th IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE, 2017.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[31] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *National Academy of Sciences*, 89(22):10915–10919, 1992.

[32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[33] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, London, England, 1992.

[34] Howard Holley and Martin Karplus. Protein secondary structure prediction with a neural network. *National Academy of Sciences*, 86(1):152–156, 1989.

[35] Sujun Hua and Zhirong Sun. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology*, 308(2):397–407, 2001.

[36] Alexander R Johansen, Casper K Sønderby, Søren K Sønderby, and Ole Winther. Deep recurrent conditional random field network for protein secondary prediction. In *8th International Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB)*, pages 73–78. ACM, 2017.

[37] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202, 1999.

[38] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

[39] Hetunandan Kamisetty and Christopher J Langmead. A Bayesian approach to protein model quality assessment. In *26th Annual International Conference on Machine Learning (ICML)*, pages 481–488. ACM, 2009.

[40] James Kennedy and Russell Eberhart. Particle swarm optimization. In *1995 - International Conference on Neural Networks (ICNN)*, volume 4, pages 1942–1948. IEEE, 1995.

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[42] Donald G Kneller, Fred E Cohen, and Robert Langridge. Improvements in protein secondary structure prediction by an enhanced neural network. *Journal of Molecular Biology*, 214(1):171–182, 1990.

[43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105. Curran Associates, Inc., 2012.

[44] Yasushi Kubota, Sho Takahashi, Ken Nishikawa, and Tatsuo Ooi. Homology in protein sequences expressed by correlation coefficients. *Journal of Theoretical Biology*, 91(2):347–361, 1981.

[45] Prince Kumar, Sanjay Bankapur, and Nagamma Patil. An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features. *Applied Soft Computing*, 86:105926, 2020.

[46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[47] Jonathan M Levin, Barry Robson, and Jean Garnier. An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS Letters*, 205(2):303–308, 1986.

[48] Zhen Li and Yizhou Yu. Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. In *25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2560–2567. AAAI Press, 2016.

[49] Zeming Lin, Jack Lanchantin, and Yanjun Qi. MUST-CNN: a multilayer shift-and-stitch deep convolutional architecture for sequence-based protein structure prediction. In *30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 27–34, 2016.

[50] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, pages 1–38, 2015.

[51] Yihui Liu, Jinyong Cheng, Yuming Ma, and Yehong Chen. Protein secondary structure prediction based on two dimensional deep convolutional neural networks. In *3rd International Conference on Computer and Communications (ICCC)*, pages 1995–1999. IEEE, 2017.

[52] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[53] Harvey Lodish, Arnold Berk, Paul Matsudaira, Chris A Kaiser, Monty Krieger, Matthew P Scott, Lawrence Zipursky, and James Darnell. *Molecular Cell Biology*. W. H. Freeman and Company, United States of America, 2003.

[54] Shiyang Long and Pu Tian. Protein secondary structure prediction with context convolutional neural network. *RSC Advances*, 9(66):38391–38396, 2019.

[55] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[56] Christophe N Magnan and Pierre Baldi. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597, 2014.

[57] Alfonso E Márquez-Chamorro, Gualberto Asencio-Cortés, Cosme E Santiesteban-Toca, and Jesús S Aguilar-Ruiz. Soft computing methods for the prediction of protein tertiary structures: A survey. *Applied Soft Computing*, 35:398–410, 2015.

[58] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (CASP) - round x. *Proteins: Structure, Function, and Bioinformatics*, 82:1–6, 2014.

[59] Ken Nishikawa and Tatsuo Ooi. Amino acid sequence homology applied to the prediction of protein secondary structures, and joint prediction with existing methods. *Biochimica et Biophysica Acta (BBA) - Protein Structure and Molecular Enzymology*, 871(1):45–54, 1986.

[60] Gabriel Bianchin de Oliveira, Helio Pedrini, and Zanoni Dias. Ensemble of bidirectional recurrent networks and random forests for protein secondary structure prediction. In *27th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 311–316, Rio de Janeiro, RJ, Brazil, 2020. IEEE.

[61] Gabriel Bianchin de Oliveira, Helio Pedrini, and Zanoni Dias. Fusion of BLAST and Ensemble of Classifiers for Protein Secondary Structure Prediction. In *33rd Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 308–315. IEEE, 2020.

[62] Gabriel Bianchin de Oliveira, Helio Pedrini, and Zanoni Dias. Protein Secondary Structure Prediction Based on Fusion of Machine Learning Classifiers. In *36th ACM/SIGAPP Symposium On Applied Computing - Bioinformatics Track (ACM SAC BIO)*, pages 26–29, Gwangju, South Korea, 2021. ACM.

[63] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[64] Dariusz Przybylski and Burkhard Rost. Alignments grow, secondary structure prediction improves. *Proteins: Structure, Function, and Bioinformatics*, 46(2):197–205, 2002.

[65] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical Report, OpenAI*, 2018.

[66] Md Aminur Rab Ratul, Maryam Tavakol Elahi, M Hamed Mozaffari, and WonSook Lee. PS8-Net: A deep convolutional neural network to predict the eight-state protein secondary structure. *arXiv preprint arXiv:2009.10380*, 2020.

[67] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, 9(2):173–175, 2012.

[68] Chandrayani N Rokde and Manali Kshirsagar. Bioinformatics: Protein structure prediction. In *4th International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2013.

[69] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

[70] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[71] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[72] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[73] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W R Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[74] Maxim Shapovalov, Roland L Dunbrack Jr, and Slobodan Vucetic. Multifaceted analysis of training and testing convolutional neural networks for protein secondary structure prediction. *PloS One*, 15(5):e0232528, 2020.

[75] Søren K Sønderby and Ole Winther. Protein secondary structure prediction with long short term memory networks. *arXiv preprint arXiv:1412.7828*, 2014.

[76] Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

[77] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[78] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

[79] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. IEEE, 2015.

[80] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[81] Mirko Torrisi, Manaz Kaleel, and Gianluca Pollastri. Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction. *Scientific Reports*, 9(1):1–12, 2019.

[82] Mostofa Rafid Uddin, Sazan Mahbub, Md Saifur Rahman, and Md Shamsuzzoha Bayzid. SAINT: self-attention augmented inception-inside-inception network improves protein secondary structure prediction. *bioRxiv*, page 786921, 2019.

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017.

[84] Guoli Wang and Roland L Dunbrack Jr. PISCES: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

[85] Guoren Wang, Yi Zhao, and Di Wang. A protein secondary structure prediction framework based on the extreme learning machine. *Neurocomputing*, 72(1-3):262–268, 2008.

[86] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports*, 6:18962, 2016.

[87] Yangxu Wang, Hua Mao, and Zhang Yi. Protein secondary structure prediction by using deep learning method. *Knowledge-Based Systems*, 118:115–123, 2017.

[88] Wafaa Wardah, Mohammed GM Khan, Alok Sharma, and Mahmood A Rashid. Protein secondary structure prediction using neural networks and deep learning: A review. *Computational Biology and Chemistry*, 81:1–8, 2019.

[89] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[90] Bingru Yang, Qu Wu, Zhou Ying, and Haifeng Sui. Predicting protein secondary structure using a mixed-modal SVM method in a compound pyramid model. *Knowledge-Based Systems*, 24(2):304–313, 2011.

[91] Xin-She Yang and Suash Deb. Cuckoo search via lévy flights. In *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 210–214. IEEE, 2009.

[92] Buzhong Zhang, Jinyan Li, and Qiang Lü. Prediction of 8-state protein secondary structures by a novel deep learning architecture. *BMC Bioinformatics*, 19(1):293, 2018.

[93] Zhen Zhang and Nan Jing. Radial basis function method for prediction of protein secondary structure. In *International Conference on Machine Learning and Cybernetics*, volume 3, pages 1379–1383. IEEE, 2008.

[94] Wei Zhong, Jieyue He, Robert Harrison, Phang C Tai, and Yi Pan. Clustering support vector machines for protein local structure prediction. *Expert Systems with Applications*, 32(2):518–526, 2007.

[95] Jian Zhou and Olga Troyanskaya. Deep Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction. In *31st International Conference on Machine Learning (ICML)*, pages 745–753, 2014.

[96] Jiyun Zhou, Hongpeng Wang, Zhishan Zhao, Ruifeng Xu, and Qin Lu. CNNH_PSS: protein 8-class secondary structure prediction by convolutional neural network with highway. *BMC Bioinformatics*, 19(4):60, 2018.

[97] Zhun Zhou, Bingru Yang, and Wei Hou. Association classification algorithm based on structure sequence in protein secondary structure prediction. *Expert Systems with Applications*, 37(9):6381–6389, 2010.