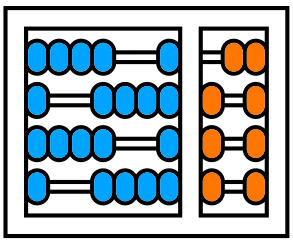
Universidade Estadual de Campinas

Instituto de Computação



Exame de Qualificação Específico

Anotações de Funções de Proteínas utilizando Processamento de Linguagem Natural e Alinhamento Local de Proteínas

---

Candidato: Gabriel Bianchin de Oliveira

Orientador: Prof. Dr. Zanoni Dias

Coorientador: Prof. Dr. Hélio Pedrini

Resumo

Com o avanço de técnicas e programas para a realização de sequenciamento, diversas proteínas tiveram a sequência de aminoácidos determinada nos últimos anos. Entretanto, analisar as funções que cada uma das proteínas exerce requer um grande esforço manual, já que a principal abordagem utilizada consiste de técnicas laboratoriais. Com isto, diversos métodos que realizam a predição de funções de proteínas começaram a se tornar mais comuns para a realização desta tarefa. Mesmo com bons resultados atingidos por estes métodos, a tarefa em questão continua sendo um problema em aberto. Nesta pesquisa, nós propomos a utilização de técnicas de processamento de linguagem natural e alinhamento local na tarefa de predição de funções de proteínas. Para isto, iremos utilizar classificadores que possuem a arquitetura baseada em Transformers, que é o estado da arte em processamento de linguagem natural, e combiná-los com avaliações de alinhamentos locais de proteínas. Iremos também explorar técnicas de aumentação de dados e *Zero-Show Learning* e *Few-Shot Learning* para avaliar o impacto na classificação dos modelos Transformers, principalmente de funções raras. Por fim, avaliaremos a utilização de destilação de conhecimento para criar um método com menos parâmetros e mais eficiente. Os resultados iniciais desta pesquisa mostraram que o método proposto consegue superar outras abordagens na literatura, que utilizam tanto redes convolucionais quanto outras arquiteturas Transformers.

1 Introdução

Esta seção caracteriza o problema a ser investigado, apresenta os principais objetivos e contribuições do trabalho, bem como as questões de pesquisa e a organização do texto.

1.1 Caracterização do Problema

Proteínas são formadas por cadeia de aminoácidos e estão presentes em todas as células dos seres vivos. Elas são responsáveis por diversas funções, tais como catálise, regulação de reações químicas e transporte [3].

O desenvolvimento de técnicas e programas para sequenciamento promoveu o acelerado aumento no número de proteínas sequenciadas nas últimas décadas. Entretanto, este mesmo

avanço não é observado na determinação das funções exercidas por proteínas, considerando a atual metodologia com alto custo e recursos laboratoriais associados [82]. Como exemplo deste fato, a base UniProtKB [69], que é o principal repositório de proteínas sequenciadas, possui cerca de 230 milhões de proteínas, enquanto a base GOA (do inglês, *Gene Ontology Annotation*) [31], que possui as proteínas das base UniProtKB com funções anotadas, tem apenas 1 milhão de amostras (considerando os dados disponíveis em Dezembro de 2022).

Devido a esta diferença, diversos métodos na literatura vêm sendo propostos para realizar a predição de funções que as proteínas exercem, principalmente utilizando abordagens automáticas. Estes métodos utilizam proteínas que já possuem as funções anotadas para assinalar as funções das proteínas alvos [44].

Dentre as abordagens mais recentes, métodos baseados em Processamento de Linguagem Natural (PLN) vêm atingindo bons resultados para a predição de funções de proteínas [30, 57, 65, 72]. Nestes métodos, os classificadores adaptam os conceitos biológicos para tarefas de PLN, considerando proteínas como palavras e aminoácidos como letras. Outros trabalhos mostraram que algoritmos de PLN, principalmente métodos com mecanismos atencionais, como arquiteturas baseadas em Transformers [70], são capazes de atingir bons resultados em outras tarefas no domínio biológico [19, 58, 59, 71, 85], tornando-se uma abordagem viável para este problema, assim como na predição de funções de proteínas [6, 37]. Mesmo com os resultados atuais, o problema de predição de funções de proteínas continua em aberto.

Dentre as arquiteturas disponíveis baseadas em Transformers, os métodos baseados no BERT (do inglês, *Bidirectional Encoder Representations from Transformers*) [17] são as mais utilizadas para o domínio de proteínas, como os modelos ProtBERT [19] e ESM [40]. Estas arquiteturas possuem restrições computacionais em relação ao tamanho de entrada de sequências, o que ocorre devido ao custo computacional dos módulos atencionais. Com isto, sequências maiores que o limite destes métodos precisam ser tratadas com pré-processamento, como cortes para o tamanho máximo permitido e quebra da proteína em pedaços menores, o que pode causar perda de informações em relação ao aprendizado do contexto da proteína como um todo. Além da limitação em relação ao tamanho da entrada de dados dos modelos, as arquiteturas Transformers pré-treinadas disponíveis na literatura

possuem milhões de parâmetros, o que os tornam pouco eficientes para a predição de uma grande quantidade de proteínas.

Mesmo com bons resultados na classificação de funções de proteínas, principalmente de funções mais comuns, pode-se destacar a dificuldade de classificação de funções raras pelos métodos da literatura, o que ocorre devido à falta de amostras destas classes no conjunto de treinamento. Sendo assim, técnicas como aumentação de dados, capazes de gerar amostras com modificações a partir dos dados de treinamento, e técnicas para o aprendizado e classificação com poucas ou sem amostras (do inglês, *Zero-Shot Learning and Few-Shot Learning*) são importantes neste contexto.

Além dos classificadores baseados em PLN, pode-se destacar métodos que utilizam ferramentas que realizam o alinhamento local de proteínas, como o BLAST [1] e o DIAMOND [5]. As abordagens baseadas nestas ferramentas utilizam o princípio de que proteínas homólogas, ou seja, proteínas que possuem certa similaridade na sequência de aminoácidos, possuem, também, funções parecidas. Alguns outros métodos apresentaram uma agregação de predições baseadas em PLN com metodologias de aprendizado de máquina com estas ferramentas [6, 34].

1.2 Objetivos

O objetivo deste trabalho é desenvolver uma abordagem capaz de predizer funções de proteínas utilizando técnicas de PLN e alinhamento local de proteínas.

Para desenvolver a metodologia proposta, alguns objetivos específicos devem ser alcançados:

- Estudar as abordagens disponíveis na literatura para a tarefa em questão;
- Avaliar as bases de dados disponíveis na literatura;
- Investigar técnicas de processamento de linguagem natural e alinhamento local no problema de predição de funções de proteínas;
- Propor um método para realizar a predição de funções de proteínas utilizando técnicas de processamento de linguagem natural e alinhamento local de proteínas;

- Realizar experimentos com o método proposto;
- Avaliar e comparar o método com outras abordagens da literatura;
- Documentar e publicar os resultados obtidos.

1.3 Questões de Pesquisa

Nesta subseção, nós apresentamos as questões de pesquisa que motivam nosso estudo:

- É possível obter resultados competitivos utilizando redes Transformers de processamento de linguagem natural na classificação de funções de proteínas?
- A utilização de métodos baseados em alinhamento local de proteínas pode auxiliar na predição quando combinados com o modelo baseado em processamento de linguagem natural?
- Como tratar funções que estão presentes em poucas amostras e adaptar o método proposto para estes casos?
- Qual é o impacto do uso de aumentação de dados durante o treinamento de modelos baseados em Transformers?
- Considerando técnicas baseadas em regras e métodos generativos, qual é a melhor forma de aplicar aumentação de dados para a tarefa sob investigação?
- A adaptação de um modelo com limitação de entrada para um modelo que aceite uma entrada maior pode melhorar os resultados na tarefa de predição de funções de proteínas?
- É possível passar o conhecimento de um modelo maior para um modelo menor sem perder a eficácia e aumentar a eficiência?

1.4 Organização do Texto

O restante do documento está organizado da seguinte forma. A Seção 2 descreve os conceitos e técnicas relevantes relacionados ao tema sob investigação. A Seção 3 apresenta os principais

métodos da literatura para a classificação de funções de proteínas. A Seção 4 descreve a metodologia proposta, a base de dados, as métricas de avaliação e os recursos computacionais que serão empregados no desenvolvimento do projeto. A Seção 5 apresenta o plano de trabalho e o cronograma de execução das atividades. A Seção 6 descreve os experimentos realizados e as publicações iniciais desta pesquisa. O Apêndice A apresenta a métrica proposta para uma avaliação mais justa dos modelos. O Apêndice B descreve os métodos comparados com o método proposto. O Apêndice C apresenta as modificações na arquitetura utilizada para lidar com sequências maiores.

2 Fundamentação Teórica

Esta seção descreve conceitos e técnicas relevantes relacionados ao tema sob investigação.

2.1 Conceitos Biológicos

Esta subseção apresenta conceitos biológicos utilizados para a predição de funções de proteínas.

2.1.1 Proteínas

As proteínas são constituídas por 20 diferentes aminoácidos que compõem a sequência, ou cadeia, da proteína [3]. A partir da sequência, são formadas estruturas tridimensionais que determinam as propriedades e funções específicas que cada uma delas irá realizar [51].

A análise das funções que cada uma das proteínas exerce possui grande impacto na descoberta de novas aplicações em biotecnologia, tais como desenvolvimento de remédios, análise de relação entre genes e fenótipos e outros estudos na saúde humana [18, 71].

2.1.2 Aminoácidos

Os aminoácidos são os responsáveis por criar a sequência da proteína através das ligações peptídicas. Eles são formados por um átomo de carbono alfa central, ligado por um grupo amina, um grupo carboxila e uma cadeia lateral [43].

Existem vinte diferentes aminoácidos, sendo estes apresentados na Tabela 1. Embora seja possível gerar infinitas sequências dependendo do tamanho da proteína (considerando L o tamanho da proteína, temos 20^L combinações possíveis), existe certa limitação em relação à quantidade de proteínas existentes. Como exemplo deste fato, o genoma humano possui cerca de 35 mil proteínas distintas [75].

Tabela 1: Lista dos 20 aminoácidos existentes.

Aminoácidos			
Alanina	Fenilalanina	Isoleucina	Serina
Arginina	Glicina	Leucina	Tirosina
Asparagina	Glutamato	Lisina	Treonina
Aspartato	Glutamina	Metionina	Triptofano
Cisteina	Histidina	Prolina	Valina

2.1.3 Funções das Proteínas

A tarefa de aplicar métodos automáticos para predizer as funções das proteínas vem se tornando popular com o grande aumento de volume de proteínas sequenciadas nas últimas décadas. Neste período, o Instituto Nacional de Pesquisa do Genoma Humano (do inglês, *National Human Genome Research Institute*) promoveu o desenvolvimento do Consórcio Ontologia Genética (do inglês, *Gene Ontology Consortium*[1]) [10], se tornando o principal método para catalogar funções de proteínas.

A Ontologia Genética classifica as funções das proteínas baseado em três ontologias, sendo elas: Ontologia de Função Molecular (FM), Ontologia de Processo Biológico (PB) e Ontologia de Componente Celular (CC). A Ontologia de Função Molecular descreve a atividade em nível molecular, como, por exemplo, transporte celular. A Ontologia de Processo Biológico representa um processo maior em que as proteínas estão envolvidas, como, por exemplo, tradução de sinal. A Ontologia de Componente Celular descreve o local onde a proteína executa a função, como, por exemplo, na mitocôndria.

As três ontologias são representadas no formato de grafo acíclico direcionado, em que os nós mais próximos da raiz são funções mais genéricas, enquanto funções mais profundas

[1] `http://geneontology.org`

são mais específicas. Cada uma das proteínas pode realizar diversas funções, assim como todas as funções ancestrais na estrutura da ontologia da função exercida são funções válidas para a proteína, já que são termos mais genéricos, o que transforma a tarefa de predição de funções de proteínas em um problema de classificação multirrótulo [7].

A Ontologia Genética é dinâmica e ocorrem frequentes atualizações nos termos, como troca de nomes, inclusões e remoções. Atualmente, as ontologias de Função Molecular, Processo Biológico e Componente Celular contam com cerca de 11 mil, 28 mil e 4 mil termos, respectivamente.

2.2 Conceitos Computacionais

Esta subseção apresenta conceitos computacionais utilizados para a predição de funções de proteínas.

2.2.1 Alinhamento Local de Proteínas

O alinhamento local de proteínas é feito com o objetivo de encontrar as subsequências de duas proteínas que possuem a maior similaridade entre si. Para isto, são utilizadas pontuações para *matches* (alinhamento perfeito entre aminoácidos das duas sequências), *mismatches* (alinhamento de dois aminoácidos distintos) e *gaps* (alinhamento de um aminoácido com uma lacuna), considerando uma matriz de pontuação.

Como resultado do alinhamento e na matriz de pontuação, o par de subsequências alinhadas obtém um valor de *bitscore*, que é a normalização do pontuação do alinhamento, de modo que, quanto maior, melhor o alinhamento, e um *e-value*, que representa um limiar da quantidade de alinhamentos que podem ser encontrados por acaso com a pontuação do alinhamento relacionado com o tamanho da base de dados avaliada, de modo que, quanto menor, mais rara é a chance.

O BLAST (do inglês, *Basic Local Alignment Search Tool*) [1] é uma ferramenta que realiza o alinhamento local de sequências (como DNA e proteínas) utilizando algumas heurísticas para otimizar o processo de cálculo de alinhamentos locais. Outros métodos para alinhamento foram apresentados na literatura, como o DIAMOND [5], mais eficiente

em relação ao tempo de processamento comparado com o BLAST.

2.2.2 Aprendizado de Máquina

Aprendizado de Máquina é uma subárea da Inteligência Artificial que utiliza um conjunto de amostras para treinar um modelo de modo que este consiga generalizar o aprendizado e ser capaz de realizar a tarefa aprendida corretamente com novas amostras [38].

Diversos trabalhos utilizam aprendizado de máquina, com tarefas envolvendo imagens, tais como diagnósticos médicos [50] e trabalhos relacionados à agricultura [56], a problemas envolvendo textos, como análise de sentimento [33] e sumarização de textos [41], a sistemas de recomendação [24], a séries temporais [67], entre outros.

2.2.3 Transformers

Mecanismos atencionais vêm se tornando presentes em diversas aplicações de aprendizado de máquina, sendo divididos em métodos de atenção *soft*, *hard* e *self-attention*. Dentre esses três mecanismos, a técnica de *self-attention* possui a capacidade de verificar por si só a interação entre seus dados de entrada, além de possuir maior eficácia computacional comparado com os outros dois mecanismos [12].

As arquiteturas baseadas em Transformers [70] se tornaram o estado da arte em diversas tarefas de PLN. Modelos dessa família de arquiteturas utilizam o conceito de *self-attention*, tornando os métodos capazes de analisar a importância de palavras anteriores e posteriores a palavra em questão.

Originalmente proposta por Vaswani *et al.* [70], a arquitetura Transformer utiliza dois diferentes blocos para a análise do texto, conforme apresentado na Figura 1. A primeira parte, chamada de codificador (do inglês, *encoder*), aprende a codificar a entrada de texto para obter uma representação numérica de cada palavra. A segunda parte, chamada de decodificador (do inglês, *decoder*), aprende a decodificar a representação.

O BERT (do inglês, *Bidirectional Encoder Representations from Transformers*) [17] é a principal arquitetura codificadora baseada em Transformers na literatura. Nesta abordagem, a parte do decodificador foi retirada e apenas o codificador foi mantido, portanto, a

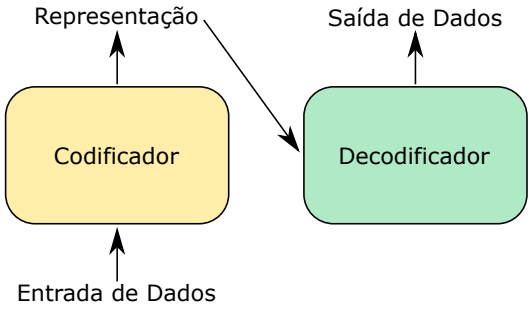


Figura 1: Arquitetura Transformer de maneira simplificada.

arquitetura possui ao final apenas a representação, e não a decodifica. Este algoritmo pode ser utilizado em diversas tarefas, tais como classificação de texto (dado um texto, realizar a classificação dele, como análise de sentimento), reconhecimento de nome-entidade (para cada palavra do texto, designar qual classe ela pertence) e pergunta e resposta (dado um texto, responder uma pergunta).

Em relação às arquiteturas decodificadoras, destaca-se a família de modelos GPTs, como o GPT-3 [4], responsável pela geração e sumarização de textos. Diferentemente do modelo BERT, que é capaz de analisar o texto de forma bidirecional, isto é, tanto palavras à esquerda quanto à direita de uma posição específica da sentença, modelos decodificadores podem apenas considerar palavras à esquerda para gerar novos textos.

Além do BERT e modelos GPT, outras arquiteturas foram propostas ao longo dos últimos anos, como XLNet [79], RoBERTa [42], ELECTRA [9] e LongFormer [2]. No domínio biológico, outras arquiteturas foram propostas, como ProtBERT e ProtBERT-BFD [19], TAPE[58], BioBERT [39] e ProtGPT2 [22].

2.2.4 Aumentação de Dados

Técnicas de aumentação de dados são utilizadas para aumentar a variabilidade dos dados a partir de algumas transformações nas amostras originais, sem a necessidade da coleta de mais dados. Em PLN, as abordagens podem ser a partir de regras manuais, interpolação de dados e rótulos, ou geradas a partir de algum aprendizado [20].

Em tarefas com proteínas, Shen *et al.* [63] propuseram técnicas de trocas de aminoácidos

a partir de um dicionário fixo de trocas, embaralhamentos locais e globais, reversões de sequências e sub-amostragens, além de combinar as transformações, sendo o único trabalho que abordou o tema até o presente momento. Entretanto, mesmo ainda não tendo sido aplicada em muitos trabalhos, os resultados obtidos por Shen *et al.* [63] mostraram que esta técnica pode obter bons resultados no domínio de proteínas.

Além da aumentação de dados tradicional utilizada em aprendizado de máquina, podemos destacar a aplicação de *design* e engenharia de proteínas, em que o objetivo é gerar sequências com propriedades específicas [78].

2.2.5 Zero-Shot Learning e Few-Shot Learning

Durante o aprendizado do classificador para realizar a predição dos dados, algumas classes podem não estar presentes no conjunto de treinamento, assim bem como serem esparsamente representadas por algumas poucas amostras, tornando o aprendizado mais complexo para o algoritmo. Em alguns casos, a coleta de novos dados com as classes mais raras não é viável, dada a dificuldade de encontrar novas amostras, do tempo necessário e do custo de coleta. Com isso, paradigmas como *Zero-Shot Learning* e *Few-Shot Learning* começaram a ser utilizados nestes casos.

A técnica de *Zero-Shot Learning* propõe um modelo que é capaz de reconhecer novas classes a partir do conhecimento adquirido no treinamento, sem que essas tenham sido apresentadas durante o aprendizado [60]. Já na abordagem *Few-Shot Learning*, o modelo deve ser capaz de aprender classes representadas por um conjunto de amostras significativamente restrito e generalizar para novos casos [64, 74].

Na tarefa de predição de funções de proteínas, técnicas de *Zero-Shot Learning* e *Few-Shot Learning* possuem grande importância, já que algumas funções são raras e possuem poucas amostras conhecidas. Meier *et al.* [46] mostraram a capacidade dos modelos Transformers pré-treinados de predizer mutações nas sequências, podendo auxiliar na predição de funções de proteínas. Xu e Wang [77] apresentaram um método capaz de anotar funções com poucas ou nenhuma amostra a partir de projeções usando a sequência, redes de interação de proteínas e a descrição da proteína sequenciada. Kulmanov e Hoehndorf [35] descreveram

um método capaz de anotar funções a partir de características de famílias de proteínas e axiomas lógicos.

2.2.6 Destilação de Conhecimento

Com o desenvolvimento de algoritmos de aprendizado de máquina robustos e com milhões de parâmetros, os resultados se tornaram cada vez melhores em diversas tarefas. Porém, com classificadores com esta configuração, os métodos não conseguem se tornar eficientes [27]. Com isso, técnicas como destilação de conhecimento foram propostas [29].

Na destilação de conhecimento, a partir de uma arquitetura já treinada, que possui o conhecimento da tarefa (chamada de professor), destila seu conhecimento para uma outra rede (chamada de estudante), que pode possuir menos parâmetros que a rede professora. Para isto, o professor supervisiona o estudante, que aprende a imitar o seu comportamento.

Em PLN, diversas abordagens foram apresentadas utilizando destilação de conhecimento. Dentre elas, pode-se destacar o DistilBERT [61], uma versão menor do BERT, e a abordagem proposta por Tang *et al.* [66], que transfere o conhecimento do BERT para uma rede bidirecional LSTM para tarefas específicas.

3 Revisão Bibliográfica

Dentre as abordagens para este problema, Hippe *et al.* [30] propõem quatro grupos diferentes de métodos, sendo eles baseados em homologia, redes, informação e sequência. Além desta divisão proposta por Hippe *et al.* [30], podemos considerar um outro: métodos baseados em mineração de texto.

Os métodos baseados em homologia utilizam da busca de proteínas homólogas em bases de dados que possuem proteínas com funções definidas, partindo do princípio que proteínas com sequências parecidas possuem funções próximas. Para realizar este processo, os métodos utilizam ferramentas que realizam o alinhamento local de proteínas, como o BLAST [1] e o DIAMOND [5]. Gong *et al.* [26] apresentaram um método que busca proteínas homólogas para gerar características da sequência e as probabilidades de cada uma das funções.

Törönen *et al.* [68] também realizam o mesmo processo, mas realizam uma filtragem após a obtenção das proteínas homólogas. Zehetner [83] propôs um método baseado no BLAST para fazer o alinhamento local de proteínas e encontrar dados homólogos. Outros métodos utilizam parte da abordagem como baseados em homologia juntamente com aprendizado de máquina, como em You *et al.* [82], Kulmanov e Hoehndorf [34] e Cao e Shen [6]. Dentre os principais problemas desta abordagem, principalmente quando usada sozinha, podemos destacar a dependência de uma base de dados com proteínas significativamente representativas considerando a sequência de aminoácidos da consulta e a dificuldade de classificar proteínas com homologia remota.

Os métodos baseados em redes consideram as redes de interação entre proteínas para realizar a predição das funções. Esta abordagem leva em conta que proteínas com funções próximas se relacionam de alguma maneira. Gligorijević *et al.* [25] apresentaram um método capaz de extrair informações da rede de interação a partir de um *auto-encoder*, que utiliza o vetor latente como entrada de dados para uma máquina de vetores de suporte. Peng *et al.* [53] utilizaram redes de interação de proteínas com técnicas de *random walk* para extrair características das redes, seguido por camadas de *auto-encoders* e convoluções. Wang *et al.* [73] propuseram um método de compressão da rede de interações para um espaço dimensional menor, onde é feita a classificação. Outros métodos utilizam esta informação em conjunto com outras características, como em You *et al.* [81] e Kulmanov *et al.* [36]. Métodos baseados em redes são bastante dependentes da existência da verificação laboratorial de redes proteína-proteína, portanto, a maioria das proteínas não possui esta informação.

Os métodos baseados em informação utilizam características biológicas para realizar a predição, tais como família e estrutura das proteínas. Cozzeto *et al.* [13] utilizaram características estruturais e biofísicas das proteínas para treinar uma máquina de vetores de suporte, enquanto Das *et al.* [15] usaram informações de super-famílias e domínio para realizar a predição. Outros métodos utilizam abordagens baseadas em informação junto com outras características, como homologia, sequência de aminoácidos e redes proteína-proteína [55, 84]. Assim como nos métodos baseados em redes, as proteínas podem não ter disponíveis as características biológicas utilizadas por estes métodos, tornando-se um

obstáculo para estas abordagens.

Os métodos baseados em mineração de texto realizam a análise de artigos científicos que relatem a função das proteínas para encontrar proteínas com funções parecidas. Fodeh *et al.* [23] apresentaram um método que analisa os resumos de artigos com técnicas de processamento de linguagem natural. Outros métodos, como You *et al.* [80] e Shatkay *et al.* [62], utilizam técnicas de mineração de texto junto com outras características. Os métodos baseados em mineração de texto são dependentes de artigos que relatem a função. Como possuem pouca quantidade de amostras, esta abordagem não é capaz de predizer corretamente a função para proteínas que não foram analisadas em laboratório.

Os métodos baseados em sequência são os mais utilizados na literatura, já que a maioria das proteínas possui a sequência analisada, mas não possui funções definidas. Nas abordagens que utilizam este paradigma, os classificadores realizam a predição considerando a sequência de aminoácidos, a partir de algum algoritmo de aprendizado de máquina. Hippe *et al.* [30] apresentaram uma rede neural do tipo *encoder-decoder*, com camadas recorrentes e módulos de atenção. Mansoor *et al.* [45] propuseram uma metodologia baseadas em redes generativas adversariais para gerar e discriminar proteínas, com o intuito de realizar um pré-treinamento da rede discriminadora. Outros trabalhos utilizam a sequência com outras características [34, 72].

Técnicas de processamento de linguagem natural, como TF-IDF (utilizada em Ranjan *et al.* [57]), algoritmos, como ELMo [28, 54] (parte da metodologia de Villegas-Morcillo *et al.* [72]) e pré-treinamento de classificadores com aprendizado *self-supervised*, como em Strodthoff *et al.* [65] e em Cao e Shen [6], vêm mostrando que PLN pode ser um caminho promissor para a tarefa de predição de funções de proteínas.

4 Materiais e Métodos

Nesta seção, nós descrevemos a metodologia, a base de dados, as métricas de avaliação e os recursos computacionais que serão utilizados no desenvolvimento da pesquisa.

4.1 Metodologia

Nesta subseção, nós apresentamos a metodologia proposta para realizar a predição de funções de proteínas.

4.1.1 Método Proposto

O método proposto é dividido em cinco partes, conforme mostra a Figura 2. Nos parágrafos a seguir, detalhamos cada uma das etapas.

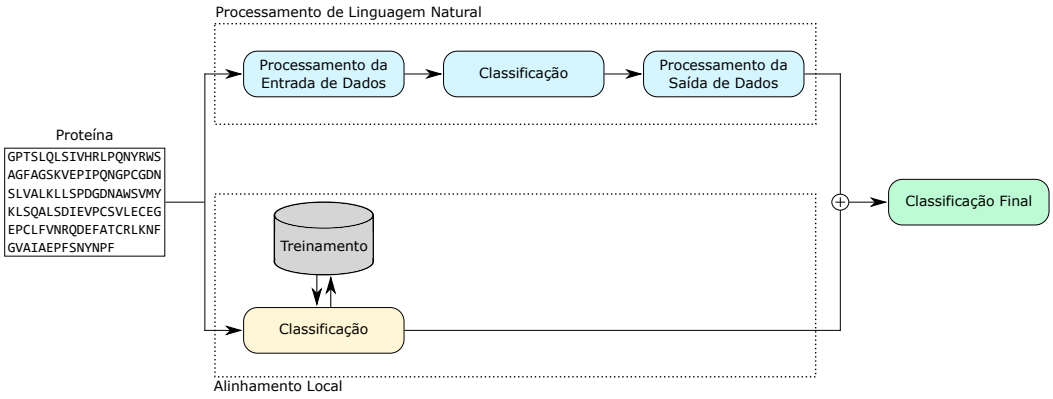


Figura 2: Visão geral do método proposto.

Processamento da Entrada de Dados Devido às limitações dos modelos Transformers em relação ao tamanho de entrada, avaliaremos a quebra da proteína em trechos menores utilizando a técnica de janela deslizante. Após a divisão da proteína, cada trecho será utilizado como entrada de dados para o modelo de classificação baseado em PLN.

Classificação baseada em PLN Para a classificação usando PLN, exploraremos arquiteturas baseadas em Transformers pré-treinadas com proteínas, como ProtBERT [19], pré-treinada na base de dados UniRef100 [11], e ProtBERT-BFD [19], pré-treinada na base BFD [32]. Devido às limitações em relação ao tamanho da entrada destes modelos, investigaremos adaptações deles para modelos que aceitem entradas maiores.

Processamento da Saída de Dados Após o processamento dos dados, devemos reunir as predições de cada um dos trechos da proteína de modo a obter a predição da proteína como um todo. Para isto, investigaremos diversos métodos para reunir a predição dos trechos, principalmente utilizando as características extraídas na última camada dos modelos Transformers.

Classificação baseada em Alinhamento Local Para a classificação baseada em alinhamento local de proteínas, utilizaremos as ferramentas DIAMOND [5] e BLAST [1], que são as ferramentas usadas por métodos da literatura [6, 34, 86]. Para isso, utilizaremos as proteínas de teste como consulta e as proteínas do conjunto de treinamento como base de busca dos alinhamentos. A classificação utilizando estas ferramentas será baseada na Equação 1, em que $S(p, f)$ indica o valor da predição P (entre 0 e 1) para uma proteína p e uma função específica f, considerando toda proteína s do conjunto de proteínas de treinamento E, que possuem certa similaridade com a proteína p, onde T_s são as funções que s exerce e $I()$ é a função identidade.

$$S(p, f) = \frac{\sum_{s \in E} I(f \in T_s) \times \text{bitscore}(p, s)}{\sum_{s \in E} \text{bitscore}(p, s)} \tag{1}$$

Agregação das Classificações Após a classificação utilizando individualmente PLN e alinhamento local de proteínas, agregaremos as predições aplicando a combinação linear entre elas, conforme apresentado na Equação 2, em que $S(p, f)$ indica o valor da predição P (entre 0 e 1) para uma proteína p e uma função específica f, $\hat{y_T}$ representa a classificação do modelo PLN e $\hat{y_D}$ indica a classificação do método de alinhamento local. Para encontrar o valor de α, otimizaremos o parâmetro obtido no conjunto de validação.

$$S(p, f) = \alpha \times \hat{y_T} + (1 - \alpha) \times \hat{y_D} \tag{2}$$

4.1.2 Aumentação de Dados

A técnica de aumentação de dados, tema apresentado na Subseção 2.2.4, para bases de proteínas não é comumente investigada na literatura. Entretanto, resultados prévios [63]

demonstraram que tais soluções apresentam resultados competitivos, sendo, portanto, um caminho de investigação com alto potencial a ser explorado.

Para realizar a aumentação de dados no conjunto de treinamento, exploramos inicialmente a utilização da família de matrizes PAM [16], que é uma coleção de matrizes de substituição que indicam a probabilidade de um aminoácido sofrer mutações, que foram analisadas laboratorialmente. Mais especificamente, pretendemos usar a matriz PAM1, utilizada para comparação de proteínas próximas em termos evolutivos. Vale ressaltar que a matriz PAM1 aponta que, para cada aminoácido, a maior probabilidade é trocada pelo próprio aminoácido, ou seja, a manutenção do aminoácido (sem mutação), além de que aminoácidos mais similares tem maior chance de serem substituídos. Acreditamos que este método de aumentação de dados proposto traga melhores resultados em relação à técnica de Shen *et al.* [63], já que os autores utilizaram trocas fixas de aminoácidos (por exemplo, o aminoácido Fenilalanina só pode ser substituído por Tirosina), enquanto a matriz de substituição PAM indica a probabilidade da troca ocorrer para cada um dos 20 aminoácidos. Os resultados iniciais destes experimentos são apresentados na Seção 6.

Além da substituição de aminoácidos considerando a matriz de substituição PAM1, iremos investigar a inserção e exclusão aleatória dos aminoácidos, inspirado pelo trabalho de Wei e Zou [76] em PLN.

Por fim, iremos também explorar a utilização de técnicas de geração de texto, de modo a gerar sequências de aminoácidos a partir de um modelo de aprendizado de máquina. Para isto, iremos utilizar arquiteturas baseadas em Transformers, como ProtGPT2 [22].

4.1.3 Classificação de Termos Raros

Dentre os trabalhos apresentados na literatura, a grande parte deles realiza a classificação de funções presentes a partir de um certo limiar de quantidade de proteínas (valores como 50 e 100). Entretanto, termos raros possuem grande importância para a determinação das funções específicas que cada proteína realiza.

Para realizarmos a predição de termos raros, iremos abordar técnicas de *Zero-Shot Learning* e *Few-Shot Learning* em tarefas de PLN, descrita na Subseção 2.2.5, assim como

aumentação de dados, para gerarmos proteínas sintéticas com os termos menos frequentes, e avaliar o impacto destas técnicas durante a classificação.

4.1.4 Destilação de Conhecimento

Ao final do desenvolvimento do método proposto, iremos aplicar a técnica de destilação de conhecimento para criar um método com menos parâmetros, entretanto, com maior eficiência, conforme descrita na Subseção 2.2.6, que trata desse tema. Para isto, iremos explorar a destilação de conhecimento para uma arquitetura Transformer mais enxuta, como o DistilBERT [61] em relação ao BERT [17], e alteração para redes ainda mais simples, como proposto por Tang *et al.* [66], com a destilação de conhecimento para redes recorrentes bidirecionais.

4.2 Base de Dados

Para o desenvolvimento desta pesquisa, usaremos as bases utilizadas no desafio de avaliação crítica da anotação de funções de proteínas (do inglês, *Critical Assessment of protein Function Annotation*), conhecido como CAFA[2].

Neste desafio, os organizadores disponibilizam diversas proteínas que foram sequenciadas e que não possuem funções anotadas por métodos laboratoriais. Durante alguns meses, estas proteínas ficam disponíveis para a predição considerando as três ontologias da ontologia genética. Ao final do tempo estipulado, as predições terminam e os organizadores esperam por alguns meses para que estas proteínas tenham as funções anotadas. Depois deste período, as proteínas que receberam verificações laboratoriais são utilizadas para avaliar os métodos propostos. Este processo está ilustrado na Figura 3.

Inicialmente, utilizamos a base derivada do CAFA 3 [86], referente ao terceiro desafio, realizado entre 2016 e 2017 (esta base já está disponível, assim como os resultados obtidos pelos métodos da literatura), criada por Kulmanov e Hoehndorf [34], que possui funções anotadas, considerando pelo menos 50 proteínas para cada rótulo. Ao explorarmos a base de dados, identificamos que existem sequências de proteínas duplicadas nos conjuntos de trei-

[2] `https://www.biofunctionprediction.org/cafa`

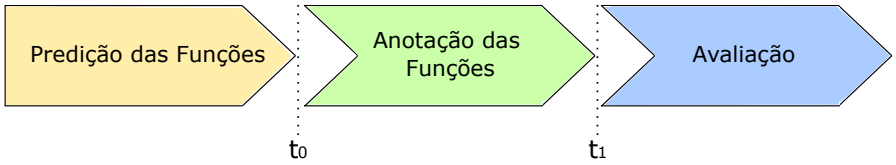


Figura 3: Método utilizado no desafio CAFA. t_0 indica o tempo em que é finalizada as predições das funções das proteínas e inicializado o aguardo das anotações das funções. t_1 indica o tempo em que é finalizada a espera das anotações e o começo da avaliação dos métodos propostos utilizando as proteínas que receberam anotações.

namento, validação e teste, considerando cada uma das ontologias. Com isso, resolvemos excluir os dados duplicados da seguinte forma: (i) removemos sequências do treinamento presentes também no conjunto de teste, (ii) removemos sequências do conjunto de validação presentes também no conjunto de teste e (iii) removemos sequências do conjunto de validação sequências presentes também no conjunto de treinamento. A Tabela 2 apresenta a quantidade de proteínas nos conjuntos de treino, validação e teste, além da quantidade de funções, em cada um das ontologias[3].

Tabela 2: Quantidade de proteínas no treinamento, validação e teste e quantidade de rótulos em cada uma das ontologias na base de dados utilizada na pesquisa.

	Componente Celular	**Função Molecular**	**Processo Biológico**
Treinamento	45.309	32.421	47.691
Validação	4.985	3.587	5.252
Teste	1.265	1.137	2.392
Rótulos	551	677	3.992

Durante o desenvolvimento da pesquisa, iremos avaliar nosso método considerando todas as funções da base CAFA 3, assim como na base CAFA 4, que teve o desafio realizado entre 2019 e 2020 (esta base está disponível, porém sem os resultados do conjunto de teste), e a base CAFA 5 (desafio possivelmente inicializado durante a realização do doutorado).

[3]A base de dados utilizada nesta pesquisa está disponível no link `https://zenodo.org/record/7409660`

4.3 Métricas de Avaliação

O desempenho da metodologia proposta será avaliado utilizando as métricas $F_{\max}$, AuPRC, IAuPRC e $S_{\min}$, que serão explicadas a seguir. As métricas $F_{\max}$, AuPRC e $S_{\min}$ são oficiais do desafio CAFA, enquanto a medida IAuPRC é uma métrica proposta nesta pesquisa.

O $F_{\max}$ calcula o maior valor de F_1 (média harmônica de precisão e revocação) considerando diversos limiares (de 0,01 a 1,00, com passo de tamanho igual a 0,01). As fórmulas da precisão e revocação em um limiar τ são apresentadas nas Equações 3 e 4, respectivamente. Nas equações, $P_i(\tau)$ indica o conjunto de termos que tiveram uma predição maior ou igual a τ para uma proteína i, T_i é o conjunto verdade para a proteína i, $m(\tau)$ reflete a quantidade de proteínas com pelo menos uma função predita com valor maior ou igual a τ, $I()$ é a função identidade, n é a quantidade de proteínas avaliadas e f é uma função da ontologia. A Equação 5 apresenta o F_{max}, como o maior valor de F_1 de todos os limiares τ.

$$\mathrm{pr}(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{\sum_f I(f \in P_i(\tau) \wedge f \in T_i)}{\sum_f I(f \in P_i(\tau))} \tag{3}$$

$$\mathrm{rc}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_f I(f \in P_i(\tau) \wedge f \in T_i)}{\sum_f I(f \in T_i)} \tag{4}$$

$$F_{\max} = \max_{\tau} \left\{ \frac{2 \times \mathrm{pr}(\tau) \times \mathrm{rc}(\tau)}{\mathrm{pr}(\tau) + \mathrm{rc}(\tau)} \right\} \tag{5}$$

Considerando os valores de precisão e revocação gerados para o cálculo do $F_{\max}$, é possível obter a curva de precisão e revocação, e, a partir dela, gerar a medida AuPRC (área sob a curva precisão e revocação, do inglês, *Area under Precision-Recall Curve*). A partir das considerações apresentadas no Apêndice A, propusemos a medida IAuPRC (área sob a curve precisão e revocação interpolada, do inglês, *Interpolated Area under Precision-Recall Curve*). Nesta medida, para todos os 100 valores τ (de 0,01 a 1,00, com passo de tamanho igual a 0,01), de revocação $\mathrm{rc}(\tau)$, atribuímos o maior valor de precisão dada uma revocação $\mathrm{pr}(\mathrm{rc}(\tau'))$, de modo que $\tau \leq \tau' \leq 1$, conforme apresentado na Equação 6.

$$\mathrm{pr}(\mathrm{rc}(\tau)) = \max \{\mathrm{pr}(\mathrm{rc}(\tau')) : \tau \leq \tau' \leq 1\} \tag{6}$$

A métrica $S_{\min}$ mede a distância semântica mínima em diversos limiares (de 0,01 a 1,00, com passo de tamanho igual a 0,01). Para isto, é necessário utilizar o conteúdo da informação, que mede a frequência relativa de um termo c em uma base de dados, como demostrado na Equação 7, em que $Pb(f|Pr(f))$ é a probabilidade de uma função f dado o conjunto de ancestrais $Pr(f)$. Com isto, é possível calcular a incerteza residual (falsos negativos) de um limiar τ, conforme a Equação 8, e a informação perdida (falsos positivos) de um limiar τ, como apresentado na Equação 9. A Equação 10 apresenta o $S_{\min}$, com o menor valor de distância semântica de todos os limiares τ.

$$\mathrm{IC}(f) = -\log(Pb(f|Pr(f)) \tag{7}$$

$$\mathrm{ru}(\tau) = \frac{1}{n}\sum_{i=1}^{n}\sum_{f\in T_i - P_i(\tau)} \mathrm{IC}(f) \tag{8}$$

$$\mathrm{mi}(\tau) = \frac{1}{n}\sum_{i=1}^{n}\sum_{f\in P_i(\tau) - T_i} \mathrm{IC}(f) \tag{9}$$

$$S_{\min} = \min_{\tau} \sqrt{\mathrm{ru}(\tau)^2 + \mathrm{mi}(\tau)^2} \tag{10}$$

4.4 Recursos Computacionais

A implementação deste projeto será feita em linguagem de programação Python, devido ao grande número de bibliotecas disponíveis e com boa documentação. O desenvolvimento utilizará bibliotecas de aprendizado de máquina, aprendizado profundo, PLN, funções científicas e numéricas e apresentação de gráficos. Algumas bibliotecas que podem ser destacadas são: `NumPy`[4], `scikit-learn`[5], `TensorFlow`[6], `HuggingFace`[7], `Matplotlib`[8] e `ktrain`[9]. Os experimentos deste projeto serão realizados no ambiente virtual Google Colab[10].

[4] `https://www.numpy.org`
[5] `https://scikit-learn.org`
[6] `https://www.tensorflow.org`
[7] `https://huggingface.co`
[8] `https://matplotlib.org`
[9] `https://github.com/amaiya/ktrain`
[10] `https://colab.research.google.com`

5 Plano de Trabalho

Nesta seção, apresentamos o cronograma com a lista de atividades realizadas e previstas para o programa de doutorado, iniciado em março de 2021.

O plano de trabalho é composto pelas seguintes atividades:

1. Obtenção dos créditos obrigatórios em disciplinas do programa de doutorado.
2. Participação no Programa de Estágio Docente (PED).
3. Exame de Qualificação Específico (EQE).
4. Revisão da Literatura.
5. Construção do método baseado em PLN utilizando Transformers.
6. Construção do método baseado em alinhamento local.
7. Agregação dos métodos baseados em PLN e alinhamento local.
8. Avaliação das técnicas de aumentação de dados.
9. Adaptação do modelo para Transformers longos.
10. Adaptação do modelo final para *Zero-Shot Learning* e *Few-Shot Learning*.
11. Aplicação de destilação de conhecimento no modelo final.
12. Escrita da tese.
13. Revisão da tese.
14. Defesa da tese.

O cronograma de execução das atividades propostas, em um prazo de 48 meses, é apresentado na Tabela 3.

O prazo para as atividades pode sofrer alterações no decorrer do desenvolvimento desta pesquisa, já que algumas atividades podem apresentar resultados mais promissores do que outras, causando realocações no cronograma.

Tabela 3: Cronograma de atividades dividido em trimestres.

Atividades	**1º ano**				**2º ano**				**3º ano**				**4º ano**			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
1	•	•	•	•												
2			•	•	•	•	•	•	•	•						
3									•							
4	•	•	•	•	•	•	•	•	•	•	•	•				
5			•	•	•	•	•	•	•							
6					•	•										
7						•	•	•	•							
8						•	•	•	•							
9								•	•	•						
10										•	•	•				
11											•	•	•	•		
12													•	•		
13															•	•
14																•

6 Resultados Preliminares

Nesta seção, nós apresentamos os resultados da pesquisa em desenvolvimento, assim como outros resultados relacionados ao aprendizado de máquina. No Apêndice B, apresentamos os métodos (Ingênuo, DIAMOND, DeepGO [36], DeepGOPlus [34], TALE+ [6]) contra os quais comparamos a nossa metodologia.

6.1 Seleção da Arquitetura

Inicialmente, avaliamos as arquiteturas ProtBERT e ProtBERT-BFD na base de dados da Ontologia de Função Molecular. Utilizamos apenas a Ontologia de Função Molecular nos primeiros experimentos devido ao fato de que a função molecular é mais dependente da sequência de aminoácidos [3].

Como as proteínas da base podem ter sequências maiores que 510 (o limite computacional do tamanho das sentenças das arquiteturas baseadas no BERT, desconsiderando os *tokens* de início e final da frase), optamos por aplicar a técnica de janela deslizante, utilizando

inicialmente a janela de tamanho igual a 100. Por exemplo, caso a proteína tenha 1.843 aminoácidos, teremos as 18 janelas de tamanho igual a 100, referente às 1.800 bases iniciais, e 1 janela com 43 aminoácidos.

Ao final do ajuste fino das arquiteturas, aplicamos uma etapa de pré-processamento de modo a obter uma predição única por proteína, e não classificações para cada uma das janelas. Para isto, utilizamos a média das previsões entre as janelas para cada um dos rótulos, já que obteve os melhores resultados considerando outras operações (mediana, mínimo e máximo). A Tabela 4 apresenta os resultados em relação à métrica $F_{\max}$ das duas arquiteturas no conjunto de validação, indicando que a arquitetura ProtBERT-BFD obteve o melhor resultado. Com isto, optamos por continuar utilizando este modelo nos próximos experimentos.

Tabela 4: $F_{\max}$ no conjunto de validação da Ontologia de Função Molecular.

Método	$F_{\max}$
ProtBERT-BFD	0,620
ProtBERT	0,609

Como resultado deste primeiro experimento, investigamos também uma agregação (do inglês, *ensemble*) dos dois modelos utilizando-os como extratores de características e os avaliamos no nosso conjunto de teste contra o método ingênuo e na base de dados utilizada pelo trabalho de Kulmanov e Hoehndorf [34], superando tanto o método ingênuo quanto o método DeepGOPlus [34]. Estes resultados estão descritos no relatório técnico "Predição de Funções Moleculares de Proteínas utilizando Aprendizado de Máquina" (Felipe Lopes de Mello, Gabriel Bianchin de Oliveira e Zanoni Dias) [47] e no artigo "Prediction of Protein Molecular Functions Using Transformers" (Felipe Lopes de Mello, Gabriel Bianchin de Oliveira, Hélio Pedrini e Zanoni Dias) [48], apresentado na *21st International Conference on Artificial Intelligence and Soft Computing* (ICAISC 2022).

6.2 Seleção da Janela Deslizante

Na sequência, avaliamos tamanhos variados para a técnica de janela deslizante, conforme apresentado na Tabela 5. Nesta avaliação, optamos por utilizar a profundidade 1 da Ontologia de Função Molecular, isto é, funções filhas do nó raiz, e proteínas que possuem até 1.000 aminoácidos. Optamos por esta configuração devido ao rápido tempo de processamento do ajuste fino do modelo neste conjunto. Como resultado, temos que, quanto maior a janela, maior o $F_{\max}$, indicando que o modelo consegue compreender melhor o contexto da proteína, sendo o melhor resultado com a janela de tamanho 500, próximo ao limite computacional do modelo. Novamente, aplicamos a média para obter o resultado para cada proteína a partir das janelas.

Tabela 5: $F_{\max}$ de janelas deslizantes com tamanhos variados no conjunto de validação da Ontologia de Função Molecular considerando a profundidade 1.

Janela	$F_{\max}$
100	0,887
200	0,894
300	0,895
400	0,895
500	0,898

Posteriormente, aplicamos a janela de tamanho 500 na Ontologia de Função Molecular considerando todas as funções da base. Além da versão com janela de tamanho igual a 500, criamos também uma versão com janelas adicionais, caso seja possível. Para esta versão com janelas adicionais, caso seja possível obter os 250 últimos aminoácidos da primeira janela e os 250 aminoácidos da segunda janela, criamos um novo trecho para a proteína em questão, conforme mostra os exemplos com sequências de tamanhos igual a 700 e 800 da Figura 4, em que as janelas em verde são as janelas padrão e a janela em azul são as janelas adicionais. A Tabela 6 mostra os resultados da classificação direta após o ajuste fino das duas abordagens.

Como um avanço da metodologia, optamos pela extração de características de cada uma das janelas de cada uma das proteínas após o ajuste fino. Para isto, utilizamos a saída do primeiro *token* da última camada da arquitetura (*token* `CLS`, responsável pela agregação da informação da sequência como um todo nas arquiteturas baseadas em BERT). Depois,

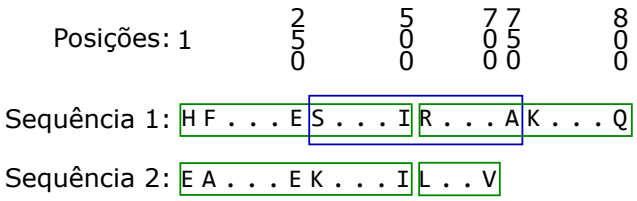


Figura 4: Exemplos das janelas criadas com os métodos de janela 500 e de janela 500 mais janelas adicionais. Para a sequência 1, com 800 aminoácidos, criamos as janelas de tamanho 500 em verde e, como é possível criar uma janela adicional, já que a primeira e a segunda janela tem pelo menos 250 aminoácidos, adicionamos esta janela (em azul, da posição 250 a 749). Para a sequência 2, com 700 aminoácidos, não é possível criar uma janela adicional, então ambos os métodos possuem as mesmas janelas.

Tabela 6: $F_{\max}$ utilizando a classificação direta e a extração de características com as janelas 500 e 500 mais janelas adicionais no conjunto de validação da Ontologia de Função Molecular.

Janela	$F_{\max}$
Classificação Direta	
500	0,636
500 + janelas adicionais	0,631
Extração de Características	
500	0,673
500 + janelas adicionais	0,675

com os vetores de características de cada trecho com 1.024 valores reais, aplicamos a média por posição, de modo a obter uma representação única por proteína. Após a agregação dos trechos de uma mesma proteína, aplicamos o vetor de características resultante em uma rede neural com multi-camadas *perceptron*, em que realizamos uma busca dos hiperparâmetros (número de camadas e quantidade de neurônios por camada), com os melhores resultados obtidos com 1.000 neurônios e apenas uma camada oculta. Os resultados são apresentados na parte inferior da Tabela 6, mostrando uma melhora significativa em relação à classificação direta e melhores resultados para a abordagem com janelas adicionais.

6.3 Seleção da Aumentação de Dados

Para a aumentação de dados, utilizamos a matriz PAM1 [16] para aplicar substituições nos aminoácidos das sequências. Para selecionar a quantidade de substituições, utilizamos a

Equação 11, em que a quantidade de substituições de uma proteína p depende do tamanho da sequência L e de uma constante k.

$$Subs(p) = L \times k \tag{11}$$

Para a verificação do impacto da aumentação de dados, aplicamos a abordagem de classificação direta na base da Ontologia de Função Molecular com a profundidade 1 (funções que são filhas diretas do nó raiz e com sequências de até 1.000 aminoácidos). Para a aumentação de dados, optamos pelo modo *offline*, isto é, criamos os dados antes da execução, ou seja, para cada dado do treinamento, criamos uma cópia e aplicamos as substituições, de modo que durante o processo de treinamento do modelo, temos a base com duas vezes o tamanho original. Vale ressaltar que os rótulos dos dados criados seguem os mesmos rótulos dos dados originais. A Tabela 7 apresenta os resultados de cinco execuções, com a média e desvio padrão, para diversos k, em que a primeira linha, indicada como "—", apresenta o resultado sem aumentações. Pela tabela, encontramos que, com k igual a 2, atingimos o maior resultado de $F_{\max}$ e o menor desvio padrão. Portanto, optamos por selecionar este valor.

Tabela 7: $F_{\max}$ por k no conjunto de validação da Ontologia de Função Molecular considerando a profundidade 1.

k	$F_{\max}$
—	0,899 ± 0,010
0,5	0,899 ± 0,012
1,0	0,899 ± 0,013
1,5	0,898 ± 0,011
2,0	0,900 ± 0,005
2,5	0,900 ± 0,007

Após a seleção do valor k no conjunto de validação da base de Ontologia de Função Molecular considerando a profundidade 1, aplicamos o melhor método (ajuste fino do modelo do modelo, seguido pela extração de características do *token* `CLS`, agregação de todas as janelas de uma mesma proteína e a utilização de um classificador final) com a geração de dados aumentados no treinamento na ontologia completa. Neste experimento, verificamos

a quantidade de vezes que poderíamos criar dados para cada proteína, ou seja, o impacto de duplicar, triplicar, quadruplicar, quintuplicar e sextuplicar o conjunto de treinamento, conforme mostra a Tabela 8. Os resultados mostram que duplicar os dados, ou seja, gerar uma cópia por proteína, atinge os melhores resultados, superando a utilização sem dados aumentados.

Tabela 8: $F_{\max}$ por quantidade de dados gerados para cada proteína no conjunto de validação da Ontologia de Função Molecular.

Quantidade de Dados Gerados (por proteína)	$F_{\max}$
—	0,675
1	0,681
2	0,672
3	0,671
4	0,677
5	0,661

6.4 Ensemble com DIAMOND e Avaliação em Todas as Ontologias

Com a melhor configuração do modelo (ajuste fino, seguido da extração de características, agregação e classificação com rede neural) e da aumentação, com k igual a 2, aplicamos o nosso método, que chamamos de TEMPROT (*Transformer-based EMbeddings for PROTein function annotation*), em todas as ontologias, conforme apresentado nas Tabelas 9 e 10, com o melhor resultado em destaque. Considerando os métodos baseados em aprendizado de máquina (DeepGO, DeepGOPlusCNN e TALE+Transformers) e o método ingênuo, o TEMPROT superou os outros métodos em 11 das 12 medidas.

Na sequência, aplicamos o *ensemble* do TEMPROT com o método DS, que chamamos de DS-TEMPROT (*DIAMOND Score and Transformer-based EMbeddings for PROTein function annotation*), e comparamos com métodos que também utilizam esta abordagem (DeepGOPlus e TALE+) e com o DS individualmente. Como resultado, o DS-TEMPROT superou os outros métodos em 8 das 11 medidas, além de empatar em outras duas avaliações.

Em relação à aumentação de dados, as sequências geradas possuem, na média, 2,04%, 2,03% e 2,03% de modificações nas ontologias de Componente Celular (CC), Função Mole-

Tabela 9: $F_{\max}$ e $S_{\min}$ do TEMPROT, DS-TEMPROT e dos métodos da literatura no conjunto de teste das três ontologias (CC, FM e PB).

	$F_{\max}$			$S_{\min}$		
Método	CC	FM	PB	CC	FM	PB
Ingênuo	0,611	0,446	0,402	10,268	9,349	25,423
DeepGO	0,379	0,489	0,337	11,880	8,821	27,414
DeepGOPlusCNN	0,664	0,531	0,498	9,783	8,240	23,799
TALE+Transformers	0,661	0,550	0,491	9,682	8,115	23,929
TEMPROT	**0,689**	**0,643**	**0,499**	**9,209**	**6,973**	**23,652**
DS	0,593	0,572	0,519	9,957	7,164	23,066
DeepGOPlus	0,677	0,619	0,553	9,515	7,090	22,648
TALE+	0,681	0,631	0,555	9,363	6,949	22,615
DS-TEMPROT	**0,692**	**0,658**	**0,562**	**9,187**	**6,761**	**22,504**

Tabela 10: AuPRC e IAuPRC do TEMPROT, DS-TEMPROT e dos métodos da literatura no conjunto de teste das três ontologias (CC, FM e PB).

	AuPRC			IAuPRC		
Método	CC	FM	PB	CC	FM	PB
Ingênuo	0,521	0,228	0,266	0,634	0,370	0,345
DeepGO	0,257	0,309	0,247	0,382	0,465	0,304
DeepGOPlusCNN	0,637	0,460	0,444	0,634	0,528	0,465
TALE+Transformers	0,613	0,444	**0,477**	0,706	0,549	0,469
TEMPROT	**0,639**	**0,561**	0,459	**0,719**	**0,664**	**0,483**
DS	0,237	0,320	0,286	0,483	0,462	0,417
DeepGOPlus	0,638	0,559	0,514	0,717	0,635	0,536
TALE+	0,643	**0,621**	**0,547**	**0,724**	0,643	**0,540**
DS-TEMPROT	**0,648**	0,584	0,510	**0,724**	**0,683**	**0,540**

cular (FM) e Processo Biológico (PB), respectivamente.

Comparamos os resultados obtidos pelo TEMPROT e DS-TEMPROT com os outras abordagens da literatura considerando as medidas $F_{\max}$ e IAuPRC na Ontologia de Função Molecular no artigo "Protein Molecular Function Annotation Based on Transformer Embeddings" (Gabriel Bianchin de Oliveira, Hélio Pedrini e Zanoni Dias) [52], apresentado na *11th Brazilian Conference on Intelligent Systems* (BRACIS 2022). Todas as avaliações, considerando as três ontologias e todas as métricas de avaliação, assim como outras análises, como a classificação de termos raros, predição de domínios biológicos e um estudo da utilização de ajuste fino e aumentação de dados está presente no artigo "TEMPROT and DS-TEMPROT: Protein Function Annotation using Transformers Embeddings and Homo-

logy Search" (Gabriel Bianchin de Oliveira, Hélio Pedrini e Zanoni Dias), submetido para um periódico internacional.

6.5 Transformação da Arquitetura para Transformers Longos

Por fim, adaptamos a arquitetura ProtBERT-BFD para aceitar sequências maiores (até 1022, no modelo ProtBERT-BFD-1022, e até 1534, na arquitetura ProtBERT-BFD-1534). A metodologia utilizada para a adaptação do modelo original para as novas variações é apresentada no Apêndice C.

Comparamos o modelo que utiliza o ajuste fino e extração de características a partir da sequência de aminoácidos com janela 500 (arquitetura ProtBERT-BFD), janela 1000 (modelo ProtBERT-BFD-1022) e 1500 (arquitetura ProtBERT-BFD-1534). Para todas os modelos, criamos janelas adicionais caso seja possível (250, 500 e 750 dos últimos aminoácidos da primeira janela e 250, 500 e 750 primeiros aminoácidos da segunda janela, respectivamente). Neste experimento, também utilizamos aumentação de dados para todas as arquiteturas.

Os resultados para cada arquitetura na Ontologia de Função Molecular são apresentados na Tabela 11. Notamos que a arquitetura ProtBERT-BFD-1022 atingiu o maior $F_{\max}$.

Tabela 11: $F_{\max}$ das arquiteturas de Transformers longos no conjunto de validação da Ontologia de Função Molecular.

Janela	$F_{\max}$
ProtBERT-BFD	0,681
ProtBERT-BFD-1022	0,686
ProtBERT-BFD-1534	0,682

6.6 Outros Resultados

Além das atividades relacionadas com o programa de doutorado, o candidato pretende continuar colaborando com outras pesquisas relacionadas ao aprendizado de máquina, não relacionadas diretamente aos objetivos desta proposta.

Até o momento, o candidato foi autor ou coautor dos seguintes artigos:

- "Algorithmic Fairness Applied to the Multi-Label Classification Problem" (Ana Paula dos Santos Dantas, Gabriel Bianchin de Oliveira, Daiane Mendes de Oliveira, Hélio Pedrini, Cid Carvalho de Souza e Zanoni Dias) [14], que será apresentado na *18th International Conference on Computer Vision Theory and Applications* (VISAPP 2023).

- "Bias Assessment in Medical Imaging Analysis: a Case Study on Retinal OCT Image Classification" (Gabriel Oliveira, Lucas David, Rafael Padilha, Ana Paula da Silva, Francine de Paula, Lucas Infante, Lucio Jorge, Patricia Xavier e Zanoni Dias) [49], apresentado na *14th International Conference on Agents and Artificial Intelligence* (ICAART 2022).

- "Ensemble of Patches for COVID-19 X-ray Image Classification" (Thiago Dong Chen, Gabriel Bianchin de Oliveira e Zanoni Dias) [8], apresentado na *14th International Conference on Agents and Artificial Intelligence* (ICAART 2022).

- "Deep Learning-based COVID-19 diagnostics of low-quality CT images" (Daniel Ferber, Felipe Vieira, João Dalben, Mariana Ferraz, Nicholas Sato, Gabriel Oliveira, Rafael Padilha e Zanoni Dias) [21], apresentado no *Brazilian Symposium on Bioinformatics* (BSB 2021).

Referências

[1] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[2] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, pages 1–17, 2020.

[3] R. Bonetta and G. Valentino. Machine learning techniques for protein function prediction. *Proteins: Structure, Function, and Bioinformatics*, 88(3):397–413, 2020.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *arXiv:2005.14165*, pages 1–75, 2020.

[5] B. Buchfink, K. Reuter, and H.-G. Drost. Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nature Methods*, 18(4):366–368, 2021.

[6] Y. Cao and Y. Shen. TALE: Transformer-based protein function Annotation with joint sequence–Label Embedding. *Bioinformatics*, 37(18):2825–2833, 2021.

[7] R. Cerri, R. C. Barros, A. A. Freitas, and A. C. P. L. F. de Carvalho. Evolving Relational Hierarchical Classification Rules for Predicting Gene Ontology-Based Protein Functions. In *Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 1279–1286. ACM, 2014.

[8] T. D. Chen, G. B. Oliveira, and Z. Dias. Ensemble of Patches for COVID-19 X-ray Image Classification. In *14th International Conference on Agents and Artificial Intelligence (ICAART)*, pages 561–567. SciTePress, 2022.

[9] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv:2003.10555*, pages 1–18, 2020.

[10] G. O. Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32(suppl_1):D258–D261, 2004.

[11] T. U. Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, pages 1–9, 2022.

[12] A. d. S. Correia and E. L. Colombini. Attention, please! A survey of Neural Attention Models in Deep Learning. *arXiv:2103.16775*, pages 1–66, 2021.

[13] D. Cozzetto, F. Minneci, H. Currant, and D. T. Jones. FFPred 3: feature-based function prediction for all Gene Ontology domains. *Scientific Reports*, 6(1):1–11, 2016.

[14] A. P. S. Dantas, G. B. Oliveira, D. M. Oliveira, H. Pedrini, C. C. Souza, and Z. Dias. Algorithmic Fairness Applied to the Multi-Label Classification Problem. In *18th International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 1–8. SciTePress, 2023.

[15] S. Das, I. Sillitoe, D. Lee, J. G. Lees, N. L. Dawson, J. Ward, and C. A. Orengo. CATH FunFHMMer web server: protein functional annotations using functional family assignments. *Nucleic Acids Research*, 43(W1):W148–W153, 2015.

[16] M. O. Dayhoff. *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, 1972.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, pages 1–16, 2018.

[18] M. E. Elhaj-Abdou, H. El-Dib, A. El-Helw, and M. El-Habrouk. Deep_CNN_LSTM_GO: Protein function prediction from amino-acid sequences. *Computational Biology and Chemistry*, 95:107584, 2021.

[19] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv:2007.06225*, pages 1–29, 2021.

[20] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. A survey of data augmentation approaches for NLP. *arXiv:2105.03075*, pages 1–24, 2021.

[21] D. Ferber, F. Vieira, J. Dalben, M. Ferraz, N. Sato, G. Oliveira, R. Padilha, and Z. Dias. Deep Learning-based COVID-19 diagnostics of low-quality CT images. In *Brazilian Symposium on Bioinformatics (BSB)*, pages 69–80. Springer, 2021.

[22] N. Ferruz, S. Schmidt, and B. Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):1–10, 2022.

[23] S. Fodeh, A. Tiwari, and H. Yu. Exploiting PubMed for protein molecular function prediction via NMF based multi-label classification. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 446–451. IEEE, 2017.

[24] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu. A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Transactions on Cybernetics*, 49(3):1084–1096, 2018.

[25] V. Gligorijević, M. Barot, and R. Bonneau. deepNF: deep network fusion for protein function prediction. *Bioinformatics*, 34(22):3873–3881, 2018.

[26] Q. Gong, W. Ning, and W. Tian. GoFDR: a sequence alignment based method for predicting protein functions. *Methods*, 93:3–14, 2016.

[27] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[28] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, 20(1):1–17, 2019.

[29] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, pages 1–9, 2015.

[30] K. Hippe, S. Gbenro, and R. Cao. ProLanGO2: protein function prediction with ensemble of encoder-decoder networks. In *11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB)*, pages 1–6. ACM, 2020.

[31] R. P. Huntley, T. Sawford, P. Mutowo-Meullenet, A. Shypitsyna, C. Bonilla, M. J. Martin, and C. O'Donovan. The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Research*, 43(D1):D1057–D1063, 2015.

[32] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[33] A. Karimi, L. Rossi, and A. Prati. Adversarial training for aspect-based sentiment analysis with BERT. In *25th International Conference on Pattern Recognition (ICPR)*, pages 8797–8803. IEEE, 2021.

[34] M. Kulmanov and R. Hoehndorf. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2019.

[35] M. Kulmanov and R. Hoehndorf. DeepGOZero: Improving protein function prediction from sequence and zero-shot learning based on ontology axioms. *bioRxiv*, pages 1–9, 2022.

[36] M. Kulmanov, M. A. Khan, and R. Hoehndorf. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668, 2018.

[37] B. Lai and J. Xu. Accurate protein function prediction via graph attention networks with predicted structure information. *Briefings in Bioinformatics*, 23(1):bbab502, 2022.

[38] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[39] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[40] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. d. S. Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, pages 1–28, 2022.

[41] Y. Liu and M. Lapata. Text summarization with pretrained encoders. *arXiv:1908.08345*, pages 1–11, 2019.

[42] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv:1907.11692*, pages 1–13, 2019.

[43] H. Lodish, A. Berk, P. Matsudaira, C. A. Kaiser, M. Krieger, M. P. Scott, L. Zipursky, and J. Darnell. *Molecular Cell Biology*. W. H. Freeman and Company, United States of America, 2003.

[44] S. Makrodimitris, R. C. H. J. Van Ham, and M. J. T. Reinders. Automatic gene function prediction in the 2020's. *Genes*, 11(11):1264, 2020.

[45] M. Mansoor, M. Nauman, H. U. Rehman, and A. Benso. Gene Ontology GAN (GO-GAN): a novel architecture for protein function prediction. *Soft Computing*, 26(1):7653–7667, 2022.

[46] J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *34th Advances in Neural Information Processing Systems (NeurIPS)*, pages 29287–29303, 2021.

[47] F. L. Mello, G. B. Oliveira, and Z. Dias. Predição de Funções Moleculares de Proteínas utilizando Aprendizado de Máquina. Technical Report IC-PFG-21-44, Institute of Computing, University of Campinas, December 2021.

[48] F. L. Mello, G. B. Oliveira, H. Pedrini, and Z. Dias. Prediction of Protein Molecular Functions Using Transformers. In *21st International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, pages 379–387. Springer, 2022.

[49] G. Oliveira, L. David, R. Padilha, A. P. d. Silva, F. d. Paula, L. Infante, L. Jorge, P. Xavier, and Z. Dias. Bias Assessment in Medical Imaging Analysis: a Case Study on Retinal OCT Image Classification. In *14th International Conference on Agents and Artificial Intelligence (ICAART)*, pages 574–580. SciTePress, 2022.

[50] G. Oliveira, R. Padilha, A. Dorte, L. Cereda, L. Miyazaki, M. Lopes, and Z. Dias. COVID-19 X-ray Image Diagnostic with Deep Neural Networks. In *Brazilian Symposium on Bioinformatics (BSB)*, pages 57–68. Springer, 2020.

[51] G. B. Oliveira, H. Pedrini, and Z. Dias. Ensemble of Template-Free and Template-Based Classifiers for Protein Secondary Structure Prediction. *International Journal of Molecular Sciences*, 22(21):11449, 2021.

[52] G. B. Oliveira, H. Pedrini, and Z. Dias. Protein Molecular Function Annotation Based on Transformer Embeddings. In *11th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 210–220. Springer, 2022.

[53] J. Peng, H. Xue, Z. Wei, I. Tuncali, J. Hao, and X. Shang. Integrating multi-network topology for gene function prediction using deep neural networks. *Briefings in Bioinformatics*, 22(2):2096–2105, 04 2020.

[54] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv:1802.05365*, pages 1–15, 2018.

[55] D. Piovesan and S. C. E. Tosatto. INGA 2.0: improving protein function prediction for the dark proteome. *Nucleic Acids Research*, 47(W1):W373–W378, 2019.

[56] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes. Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8:1852, 2017.

[57] A. Ranjan, D. Fernandez-Baca, S. Tripathi, and A. Deepak. An ensemble tf-idf based approach to protein function prediction via sequence segmentation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(8):1–12, 2021.

[58] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, and Y. S. Song. Evaluating protein transfer learning with TAPE. In *32nd Advances in Neural Information Processing Systems (NeurIPS)*, page 9689. NIH Public Access, 2019.

[59] R. M. Rao, J. Liu, R. Verkuil, J. Meier, J. Canny, P. Abbeel, T. Sercu, and A. Rives. MSA Transformer. In *38th International Conference on Machine Learning (ICML)*, pages 8844–8856, 2021.

[60] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *32nd International Conference on Machine Learning (ICML)*, pages 2152–2161. PMLR, 2015.

[61] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108*, pages 1–5, 2019.

[62] H. Shatkay, S. Brady, and A. Wong. Text as data: Using text-based features for proteins representation and for computational prediction of their characteristics. *Methods*, 74(1):54–64, 2015.

[63] H. Shen, L. C. Price, M. T. Bahadori, and F. Seeger. Improving Generalizability of Protein Sequence Models with Data Augmentations. *bioRxiv*, pages 1–17, 2021.

[64] Y. Song, T. Wang, S. K. Mondal, and J. P. Sahoo. A comprehensive survey of few-shot learning: evolution, applications, challenges, and opportunities. *arXiv:2205.06743*, pages 1–24, 2022.

[65] N. Strodthoff, P. Wagner, M. Wenzel, and W. Samek. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409, 2020.

[66] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin. Distilling task-specific knowledge from BERT into simple neural networks. *arXiv:1903.12136*, pages 1–8, 2019.

[67] S. J. Taylor and B. Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[68] P. Törönen, A. Medlar, and L. Holm. PANNZER2: a rapid functional annotation web server. *Nucleic Acids Research*, 46(W1):W84–W88, 2018.

[69] UniProt. UniProt Database. `https://www.uniprot.org`. Online; acessado em 27 de Dezembro de 2022.

[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *30th Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[71] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher, and N. F. Rajani. BERTology meets biology: Interpreting attention in protein language models. *arXiv:2006.15222*, pages 1–24, 2020.

[72] A. Villegas-Morcillo, S. Makrodimitris, R. C. H. J. van Ham, A. M. Gomez, V. Sanchez, and M. J. T. Reinders. Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics*, 37(2):162–170, 2021.

[73] S. Wang, H. Cho, C. Zhai, B. Berger, and J. Peng. Exploiting ontology graph for predicting sparsely annotated gene function. *Bioinformatics*, 31(12):i357–i364, 2015.

[74] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):1–34, 2020.

[75] W. Wardah, M. G. M. Khan, A. Sharma, and M. A. Rashid. Protein secondary structure prediction using neural networks and deep learning: A review. *Computational Biology and Chemistry*, 81:1–8, 2019.

[76] J. Wei and K. Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv:1901.11196*, pages 1–9, 2019.

[77] H. Xu and S. Wang. ProTranslator: zero-shot protein function prediction using textual description. In *26th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 279–294, 2022.

[78] K. K. Yang, Z. Wu, and F. H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019.

[79] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *32nd Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–11. NIH Public Access, 2019.

[80] R. You, X. Huang, and S. Zhu. DeepText2GO: Improving large-scale protein function prediction with deep semantic text representation. *Methods*, 145:82–90, 2018.

[81] R. You, S. Yao, H. Mamitsuka, and S. Zhu. DeepGraphGO: graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics*, 37(Supplement_1):i262–i271, 2021.

[82] R. You, Z. Zhang, Y. Xiong, F. Sun, H. Mamitsuka, and S. Zhu. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, 34(14):2465–2473, 2018.

[83] G. Zehetner. OntoBlast function: From sequence similarities directly to potential functional annotations by ontology terms. *Nucleic Acids Research*, 31(13):3799–3803, 2003.

[84] C. Zhang, P. L. Freddolino, and Y. Zhang. COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Research*, 45(W1):W291–W299, 2017.

[85] H. Zhang, F. Ju, J. Zhu, L. He, B. Shao, N. Zheng, and T.-Y. Liu. Co-evolution Transformer for Protein Contact Prediction. *34th Advances in Neural Information Processing Systems (NeurIPS)*, pages 14252–14263, 2021.

[86] N. Zhou, Y. Jiang, T. R. Bergquist, A. J. Lee, B. Z. Kacsoh, A. W. Crocker, K. A. Lewis, G. Georghiou, H. N. Nguyen, M. N. Hamid, L. Davis, T. Dogan, V. Atalay, A. S. Rifaioglu, A. Dalkıran, R. Cetin Atalay, C. Zhang, R. L. Hurto, P. L. Freddolino, Y. Zhang, P. Bhat, F. Supek, J. M. Fernández, B. Gemovic, V. R. Perovic, R. S. Davidović, N. Sumonja, N. Veljkovic, E. Asgari, M. R. Mofrad, G. Profiti, C. Savojardo, P. L. Martelli, R. Casadio, F. Boecker, H. Schoof, I. Kahanda, N. Thurlby, A. C. McHardy, A. Renaux, R. Saidi, J. Gough, A. A. Freitas, M. Antczak, F. Fabris, M. N. Wass, J. Hou, J. Cheng, Z. Wang, A. E. Romero, A. Paccanaro, H. Yang, T. Goldberg, C. Zhao, L. Holm, P. Törönen, A. J. Medlar, E. Zosa, I. Borukhov, I. Novikov, A. Wilkins, O. Lichtarge, P.-H. Chi, W.-C. Tseng, M. Linial, P. W. Rose, C. Dessimoz, V. Vidulin, S. Dzeroski, I. Sillitoe, S. Das, J. G. Lees, D. T. Jones, C. Wan, D. Cozzetto, R. Fa, M. Torres, A. Warwick Vesztrocy, J. M. Rodriguez, M. L. Tress, M. Frasca, M. Notaro, G. Grossi, A. Petrini, M. Re, G. Valentini, M. Mesiti, D. B. Roche, J. Reeb, D. W. Ritchie, S. Aridhi, S. Z. Alborzi, M.-D. Devignes, D. C. E. Koo, R. Bonneau, V. Gligorijević, M. Barot, H. Fang, S. Toppo, E. Lavezzo, M. Falda, M. Berselli, S. C. Tosatto, M. Carraro, D. Piovesan, H. Ur Rehman, Q. Mao, S. Zhang, S. Vucetic, G. S. Black, D. Jo, E. Suh, J. B. Dayton, D. J. Larsen, A. R. Omdahl, L. J. McGuffin, D. A. Brackenridge, P. C. Babbitt, J. M. Yunes, P. Fontana, F. Zhang, S. Zhu, R. You, Z. Zhang, S. Dai, S. Yao, W. Tian, R. Cao, C. Chandler, M. Amezola, D. Johnson, J.-M. Chang, W.-H. Liao, Y.-W. Liu, S. Pascarelli, Y. Frank, R. Hoehndorf, M. Kulmanov, I. Boudellioua, G. Politano, S. Di Carlo, A. Benso, K. Hakala, F. Ginter, F. Mehryary, S. Kaewphan, J. Björne, H. Moen, M. E. Tolvanen, T. Sala-

koski, D. Kihara, A. Jain, T. Šmuc, A. Altenhoff, A. Ben-Hur, B. Rost, S. E. Brenner, C. A. Orengo, C. J. Jeffery, G. Bosco, D. A. Hogan, M. J. Martin, C. O'Donovan, S. D. Mooney, C. S. Greene, P. Radivojac, and I. Friedberg. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, 20(1):244, 2019.

A Considerações sobre AuPRC e IAuPRC

A medida AuPRC, junto com $F_{\max}$ e $S_{\min}$, é amplamente utilizada para avaliação dos métodos da literatura. Para obter esta métrica, utiliza-se dos valores de precisão e revocação em todos os limiares do cálculo da medida $F_{\max}$.

Mesmo sendo uma das principais medidas utilizadas na literatura, a AuPRC possui desvantagens. Alguns métodos podem realizar classificações que possuem somente valores mais altos de revocação, que deveria ser recompensado pela medida, visto que valores mais altos de revocação indicam baixa presença de falsos negativos. Entretanto, por não conseguirem predizer baixos valores de revocação, a área sob a curva fica comprometida, principalmente pela região de baixa revocação. A Figura 5 apresenta um exemplo da curva precisão e revocação de dois métodos, em que o método A obteve 0,584 e o método B atingiu 0,618 de AuPRC. Pela Figura 5, fica evidente que a curva do método A é superior em quase todos os pontos de revocação em relação ao método B, porém o método A não é capaz de predizer valores baixos de revocação, o que não deveria ser uma punição para este método.

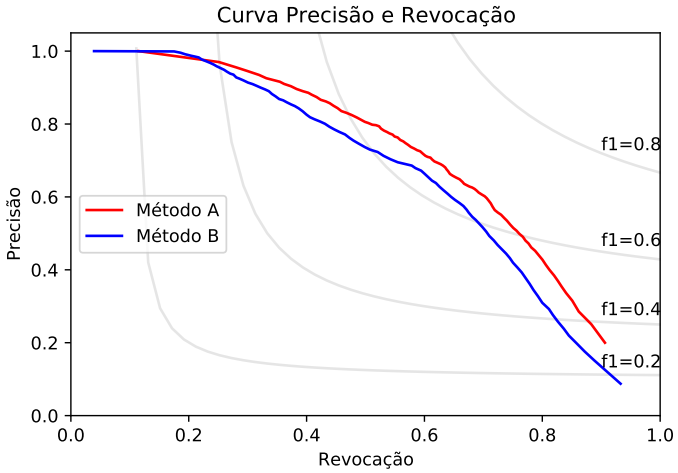


Figura 5: Avaliação considerando a métrica AuPRC dos métodos A e B.

Sendo assim, propomos a utilização da métrica IAuPRC, conforme expressa na Equação 6, na Seção 4. A partir da interpolação aplicada com a medida AuPRC, a comparação dos dois métodos fica mais justa, sendo que, com a interpolação, o método A atingiu

0,683 e o método B obteve 0,643. Os gráficos dos dois métodos são apresentados na Figura 6.

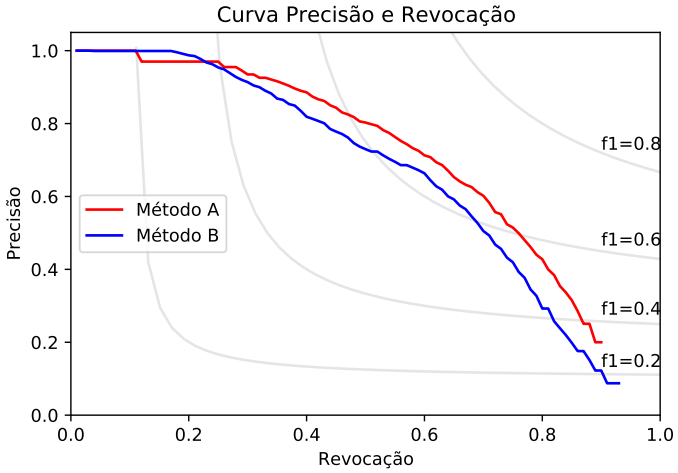


Figura 6: Avaliação considerando a métrica IAuPRC dos métodos A e B.

B Métodos Comparados

Para compararmos o nosso método baseado em PLN (TEMPROT) e o *ensemble* do nosso método baseado em PLN com o alinhamento local de proteínas baseado no DIAMOND (DS-TEMPROT), utilizamos os métodos de comparação do desafio CAFA e métodos da literatura. Para os métodos da literatura, selecionamos aqueles que possuem a classificação baseada em sequência de aminoácidos utilizando PLN e aprendizado de máquina, além de possuírem o código disponível para reprodução. Para todos os métodos utilizados para comparação, reexecutamos cada um deles na base de dados utilizada nesta pesquisa. Descrevemos cada um das abordagens na sequência.

Ingênuo A primeira abordagem de comparação que foi proposto pelo desafio CAFA é o método ingênuo. Nele, a predição de uma função f para uma proteína p no conjunto de teste é igual a divisão do número de proteínas que contém aquela função específica no conjunto de treinamento N_f pelo número total de proteínas no conjunto de treinamento N, conforme apresentado na Equação 12.

$$S(p, f) = \frac{N_f}{N} \tag{12}$$

DIAMOND Score O segundo método para comparação proposto pelo desafio CAFA corresponde à utilização de ferramentas de alinhamento local para realizar a predição. Para isto, utilizamos o DIAMOND, assim como outros métodos da literatura [6, 34]. Para realizar o alinhamento local entre as proteínas de teste ou validação com as proteínas de treinamento, utilizamos o *e-value* igual a 0,001 como valor máximo para este parâmetro e aplicamos a Equação 1. Chamaremos este método de DIAMOND Score (DS) no restante deste texto.

DeepGO O método DeepGO [36] aplica trigramas na sequência de aminoácidos para gerar as características que seguem para uma rede convolucional. Ao final das camadas convolucionais, a arquitetura possui neurônios no formato da hierarquia da ontologia em questão. Este método pode utilizar características de redes de interação entre proteínas, entretanto, nos nossos experimentos aplicamos somente a sequência para fazer uma com-

paração justa. O DeepGO possui restrições em relação ao tamanho máximo das proteínas durante o treinamento, restringindo a 1.000 aminoácidos por sequência.

DeepGOPlusCNN Este método [34] é uma evolução do método DeepGO, transformando os aminoácidos em *one-hot encoding* e aplicando as sequências para redes convolucionais. Assim como o modelo anterior, o DeepGOPlusCNN possui restrições em relação ao tamanho máximo de entrada durante o treinamento, com limite de 2.000 aminoácidos por proteína.

DeepGOPlus A partir das predições do DeepGOPlusCNN e do DS, a combinação linear entre os dois métodos é realizada, utilizando a mesma fórmula utilizada no nosso *ensemble* (Equação 2), a fim de combinar o método baseado em *deep learning* com o alinhamento local de proteínas. Para realização o *ensemble*, aplicamos os mesmos valores de α reportados pelos autores [34].

TALE+Transformers Esta abordagem utiliza a arquitetura codificadora do Transformers original [70] para a sequência de aminoácidos e a organização da Ontologia Genética para realizar a classificação. Para cada ontologia, os autores [6] aplicam um *ensemble* das predições a partir de modificações na arquitetura Transformers, como a quantidade de blocos codificadores e cabeças de atenção. Para os resultados envolvendo o TALE+Transformers, aplicamos o *ensemble* conforme reportado pelos autores. Este método possui restrição em relação ao tamanho da sequência durante o treinamento, em que para sequências maiores do que 1.000 aminoácidos, um trecho único de tamanho igual a 1.000 aminoácidos é utilizado.

TALE+ Assim como utilizado no método DeepGOPlus, o método TALE+ [6] aplica a combinação linear entre as predições do TALE+Transformers com o DS. Para realização o *ensemble*, aplicamos os mesmos valores de α reportados pelos autores [6].

C Adaptação do ProtBERT-BFD para Transformers Longos

A arquitetura LongFormer [2] apresenta o módulo atencional global e local, em que os *tokens* comuns do texto conseguem analisar apenas uma janela próxima, enquanto o *token* `CLS`, responsável pelo entendimento da sentença como um todo, continua de modo global. Com esta adaptação, a complexidade do módulo de atenção cai de $\mathcal{O}(n^2)$, referente à atenção padrão das arquiteturas baseadas em BERT, para $\mathcal{O}(nk)$, com o módulo de atenção global e local, em que n é o tamanho da sequência e k é o tamanho da janela, o que permite a entrada de textos maiores comparado com os modelos com módulo atencional com complexidade quadrática.

No trabalho que apresenta o LongFormer, os autores mostraram que é possível adaptar um modelo que já possui conhecimento para aceitar sequências maiores de entrada, apenas alterando os módulos atencionais e pré-treinando ele com poucas amostras. Baseado na adaptação proposta pelo LongFormer, aplicamos o mesmo conceito para a arquitetura ProtBERT-BFD, que consegue lidar com 510 aminoácidos, mais os dois *tokens* especiais da arquitetura(`CLS`, de início da sequência, e `SEP`, de final da sequência), para sequências maiores. Para isto, aplicamos o conceito de atenção por janela do LongFormer, considerando a janela de tamanho igual a 512 para cada *token*, reduzindo a complexidade de quadrática para linear. Vale ressaltar que o modelo ProtBERT-BFD possui a codificação posicional de tamanho 4096, entretanto, por usar atenção de todos os *tokens* para todos os *tokens* (atenção quadrática), não é possível, com o poder computacional disponível, usar sequências maiores que o 510.

Como pré-treinamento do modelo para sequências maiores, via adaptação do módulo de atenção, selecionamos aleatoriamente 10.000 sequências com mais de 510 aminoácidos até o limite da arquitetura (1022 para ProtBERT-BFD-1022 e 1534 para ProtBERT-BFD-1534) da base UniRef50 [11]. Para a adaptação e o pré-treinamento das arquiteturas adaptadas, aplicamos a tarefa de predição de *tokens* mascarados, conforme o treinamento da arquitetura ProtBERT-BFD, com o mascaramento de 15% dos *tokens*. O pré-treinamento utilizou uma

época, conforme utilizado no trabalho do LongFormer para a adaptação do RoBERTa para modelo longo, com *learning rate* igual a 10^{-5}, *batch size* igual a 2 para o ProtBERT-BFD-1022 e *batch size* igual a 1 para o ProtBERT-BFD-1534 e aquecimento com 500 sequências. Todo o processo de pré-treinamento demorou cerca de 3 horas para a arquitetura ProtBERT-BFD-1022 e 9 horas para a arquitetura ProtBERT-BFD-1534. Após o pré-treinamento, as arquiteturas podem receber o ajuste fino para a tarefa em que serão aplicadas. A Tabela 12 apresenta a quantidade de parâmetros de cada uma das arquiteturas.

Tabela 12: Comparativo da quantidade de parâmetros entre as arquiteturas ProtBERT-BFD, ProtBERT-BFD-1022 e ProtBERT-BFD-1534.

Arquitetura	**Quantidade de Parâmetros**
ProtBERT-BFD	419.931.136
ProtBERT-BFD-1022	380.019.712
ProtBERT-BFD-1534	380.544.000