UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE COMPUTAÇÃO

Exame de Qualificação de Mestrado

23 de Abril de 2025

Anotação de Funções Proteicas Utilizando Redes de Interações Proteína-Proteína

Candidato: Yeonatan Mauhnoom

Orientador: Prof. Dr. Zanoni Dias

Coorientador: Prof. Dr. Hélio Pedrini

Sumário

Resumo

As funções desempenhadas pelas proteínas são essenciais para os processos biológicos em qualquer organismos. Entretanto, a anotação experimental dessas funções é um processo custoso e demorado. Nesse contexto, as redes de interações proteína-proteína (PPI) oferecem informações valiosas para a inferência funcional, uma vez que proteínas que interagem frequentemente podem compartilhar funções biológicas similares. Este trabalho propõe o uso de técnicas de aprendizado profundo para a anotação de funções de proteínas, utilizando dados de pontuações de confiança para evidências de PPI obtidos da base STRING. Por tratar-se de uma tarefa de aprendizado supervisionado e de classificação multirrótulo, exploraremos abordagens que abrangem desde métodos mais simples para dados tabulares, como redes neurais e modelos baseados em árvores aleatórias, até representações mais complexas. Estas incluem a conversão dos dados tabulares em imagens para a aplicação de redes convolucionais, bem como o uso de redes neurais baseadas em grafos para investigar as propriedades estruturais das interações proteína-proteína e comparar os melhores modelos desenvolvidos com outros métodos do estado da arte que utilizam dados de redes de interações. Por fim, investigaremos estratégias híbridas que combinem dados de sequências de aminoácidos ou estruturas tridimensionais de proteínas com nossa abordagem baseada em dados de redes de interações. Em particular, pretendemos integrar nossa abordagem final a um modelo baseado em sequências desenvolvido em pesquisas anteriores do grupo, ampliando assim as possibilidades de análise e predição funcional.

Capítulo 1

Introdução

Neste capítulo, introduzimos o tema e a motivação da pesquisa, destacando os principais objetivos, além de apresentar a organização geral deste documento.

1.1 Motivação e Contexto

As proteínas desempenham um papel fundamental como blocos de construção da vida, sendo essenciais para uma ampla gama de funções biológicas, incluindo especificidade de ligação, estabilidade mecânica, catálise de reações bioquímicas, transporte e transdução de sinal [1, 2]. A importância das proteínas está intimamente ligada à sua participação em processos biológicos e moleculares, sendo determinantes para o funcionamento celular de organismos vivos [3]. Essa relevância torna a compreensão de suas funções indispensável em diversas áreas, como a biotecnologia e o desenvolvimento de novos fármacos, uma vez que muitas terapias dependem de proteínas-alvo específicas para o tratamento de doenças [4].

Para organizar e descrever as funções das proteínas, o *Gene Ontology* (GO) as categoriza em classes funcionais hierarquicamente relacionadas, estruturadas em três ontologias: Função Molecular (do inglês, *Molecular Function* – MF), que descreve as atividades bioquímicas específicas realizadas pelas proteínas; Processo Biológico (do inglês, *Biological Process* – BP), que se refere aos contextos biológicos mais amplos nos quais as proteínas estão envolvidas; e Componente Celular (do inglês *Cellular Component* – CC), que indica onde as proteínas estão localizadas ou atuam dentro da célula [5].

Apesar do notável progresso tecnológico e do crescente desenvolvimento dos campos da genômica e proteômica, que possibilitaram o sequenciamento massivo de proteínas e o crescimento exponencial de bases de dados como a UniProt, a validação experimental das funções proteicas ainda é um processo lento, laborioso e caro. Esse tipo de problema encontra nas abordagens computacionais e estatísticas, particularmente em modelos preditivos baseados em aprendizado de máquina e aprendizado profundo, uma aliada estratégica para acelerar a descoberta científica. Essas ferramentas tem mostrado resultados pertinentes para extrair padrões funcionais tanto de sequências aminoacídicas quanto de arquiteturas tridimensionais, permitindo acompanhar de forma eficiente a catalogação crescente de proteínas. A base UniProt, por exemplo, registra atualmente mais de 250 milhões de proteínas sequenciadas. No entanto, a maioria dessas proteínas permanece sem uma anotação funcional confiável, evidenciando uma lacuna significativa no conhecimento biológico [6].

Nesse cenário, a análise de interações proteína-proteína (do inglês, *Protein-Protein Interaction* – PPI) emerge como uma abordagem essencial, fornecendo informações fundamentais

sobre os papéis funcionais das proteínas, indicando os processos biológicos e as vias metabólicas em que estão envolvidas. Assim, as redes de interações de proteínas podem ser utilizadas para inferir padrões que relacionam as funções desempenhadas por essas proteínas [7, 8].

Embora os métodos atuais para classificação de funções de proteínas tenham apresentado resultados promissores, a anotação funcional permanece um desafio, especialmente devido à elevada complexidade inerente aos problemas de classificação multirrótulo [9], nos quais cada instância pode estar associada a múltiplas categorias simultaneamente (isto é, proteínas podem exercer múltiplas funções biológicas). Nesse sentido, competições recentes, como o desafio CAFA [10], têm evidenciado a demanda por soluções mais robustas capazes de explorar eficientemente as propriedades topológicas das redes de interações proteína-proteína (PPI). Diante de tal cenário, tem-se discutido o potencial das técnicas de aprendizado profundo combinadas à caracterização matemática das redes (por exemplo, por meio de estruturas de grafos) para aprimorar a inferência funcional. Adicionalmente, a integração de múltiplas fontes de dados — como sequências de aminoácidos e redes de interações — tem demonstrado resultados promissores em estudos recentes [11–13], sugerindo que abordagens híbridas e inovadoras podem oferecer melhorias significativas na predição de funções proteicas.

1.2 Objetivos

Este trabalho tem como objetivo principal desenvolver uma nova abordagem para anotação funcional de proteínas utilizando dados de interações proteína-proteína e comparar com métodos da literatura que também utilizem essa informação. Por fim, agregaremos a nossa metodologia ao método desenvolvido pelo grupo de pesquisa anteriormente [14], no servidor *web* SUPERMAGO+web [1]. Para alcançar esse propósito, foram definidos os seguintes objetivos específicos:

- Realizar revisão bibliográfica e estudo das principais técnicas de predição funcional que utilizam redes PPI;
- Desenvolver e testar modelos iniciais (*baseline*), investigando configurações mais simples e estabelecendo uma referência de desempenho;
- Explorar o treinamento de modelos em diferentes representações das redes de interações, como dados tabulares, representações visuais e abordagens baseadas em grafos;
- Investigar a combinação de técnicas de aprendizado autossupervisionado;
- Comparar e validar as abordagens propostas em relação a métodos de referência que utilizam redes PPI, visando mensurar vantagens, limitações e potenciais de melhoria;
- Integrar o melhor modelo resultante ao servidor do laboratório contendo o método baseado em sequência desenvolvido anteriormente pelo grupo de pesquisa, complementando abordagens que utilizam sequências de aminoácidos com informações de interações proteína-proteína;
- Documentar e publicar os resultados obtidos.

[1] `https://supermago.ic.unicamp.br`

Com isso, esse projeto visa contribuir para o desenvolvimento de metodologias avançadas para a predição funcional de proteínas, utilizando redes de interações PPI. Além disso, o trabalho visa fornecer soluções computacionais mais robustas, escaláveis impulsionando os avanços no campo da bioinformática e da proteômica.

1.3 Questões de Pesquisa

A partir dos objetivos delineados, foram estabelecidas as seguintes questões a serem investigadas ao longo deste trabalho:

- Modelos baseados em grafos (por exemplo, GCNNs) podem apresentar vantagens consideráveis em relação a outras representações, ao explorar as relações topológicas inerentes às redes PPI?
- A aplicação de técnicas de aprendizado autossupervisionado (como reconstrução e/ou estratégias contrastivas em *autoencoders*) pode produzir *embeddings* mais robustos, resultando em ganhos de desempenho?
- O uso de *transfer learning* em modelos pré-treinados, adaptados às imagens ou representações derivadas das PPI, é efetivo ou acarreta perdas decorrentes da geração de tais representações visuais das redes?
- A integração dos modelos gerados com base em *embeddings* de sequências pode aprimorar a predição de anotações funcionais ao combinar informações de interações proteína-proteína e dados de sequência?

1.4 Estrutura do Documento

O Capítulo 2 apresenta os conceitos teóricos e as técnicas relacionadas ao tema de estudo, abrangendo abordagens clássicas e recentes na predição funcional de proteínas. O Capítulo 3 detalha a metodologia proposta e introduz os materiais, as métricas de avaliação e os recursos computacionais empregados no desenvolvimento do projeto. O Capítulo 4 apresenta o plano de trabalho e o cronograma de execução. Por fim, para maiores detalhes, os apêndices estão organizados da seguinte forma: o Apêndice A reúne os principais trabalhos correlatos, o Apêndice B descreve as bases de dados e inclui uma análise exploratória, o Apêndice C define formalmente as métricas de avaliação empregadas e o Apêndice D descreve os resultados iniciais e discute suas implicações.

Capítulo 2

Fundamentação Teórica e Revisão Bibliográfica

Neste capítulo, apresentamos uma revisão bibliográfica dos conceitos biológicos e computacionais que embasam a pesquisa, além de apresentar alguns resultados de trabalhos correlatos relevantes.

2.1 Conceitos Biológicos

Nesta seção, buscamos elucidar os principais conceitos biológicos envolvendo o tema de anotação de funções proteicas, ontologia genômica e redes de interações proteína-proteína.

2.1.1 Proteínas

As proteínas são macromoléculas compostas por longas cadeias de aminoácidos, ligados por ligações peptídicas e constituem os principais componentes estruturais e funcionais das células [15]. Cada aminoácido é formado por um grupo amino, um grupo carboxila e uma cadeia lateral (ou grupo R), que confere propriedades químicas específicas. A sequência e a composição desses aminoácidos, assim como o meio no qual cada proteína se encontra, determinam tanto a estrutura quanto a função final da mesma. Essas conformações resultam de interações não covalentes, incluindo ligações de hidrogênio, forças eletrostáticas e efeitos hidrofóbicos [2]. As proteínas são essenciais para a vida, desempenhando funções cruciais em processos biológicos como catálise de reações químicas, suporte estrutural, transporte de moléculas e regulação de atividades celulares [16].

As proteínas podem ter sua expressão e função alteradas tanto entre diferentes espécies quanto em distintos tecidos de um mesmo organismo. Em regiões especializadas, como muscular, nervosa ou hepática, a síntese proteica é controlada por fatores de transcrição e processos de sinalização específicos, garantindo que cada tipo celular produza o conjunto de proteínas adequado à sua função biológica. Além disso, a localização intracelular — em membranas, organelas ou citoplasma — exerce influência decisiva sobre a atuação dessas macromoléculas, permitindo que participem de processos como sinalização e transporte de metabólitos. A interação coordenada entre diferentes proteínas também desempenha um papel crucial, formando complexos funcionais ou regulando-se mutuamente. Esse arranjo integrado reflete a grande diversidade de vias biológicas em que as proteínas estão envolvidas, desde respostas imunológicas até o controle do ciclo celular e doenças relacionadas a

modificações proteicas (por exemplo, neurodegenerativas, autoimunes ou oncológicas), ressaltando sua importância na manutenção e na adaptação dos organismos em variados contextos fisiológicos e patológicos.

A compreensão das funções proteicas é amplamente apoiada por ontologias, como a *Gene Ontology* (GO), que classifica as proteínas no contexto de suas funções moleculares, processos biológicos e componentes celulares [5].

2.1.2 Ontologia Genética

A *Gene Ontology* (GO) é um consórcio global que visa padronizar a descrição das funções de genes e proteínas, criando um vocabulário controlado e estruturado em três domínios [5]:

- Função Molecular (do inglês, *Molecular Function* – MF): descreve as atividades bioquímicas específicas realizadas pelas proteínas.
- Processo Biológico (do inglês, *Biological Process* – BP): representa os contextos biológicos mais amplos em que essas funções se inserem.
- Componente Celular (do inglês, *Cellular Component* – CC): indica os locais da célula onde as proteínas atuam.

Esses domínios são organizados hierarquicamente em um grafo acíclico direcionado, no qual cada nó corresponde a um termo de ontologia, enquanto as arestas representam relações como "é um tipo de" ou "é parte de", conforme representado na Figura 2.1. Essa estrutura não apenas permite refinar termos gerais em descrições mais específicas, mas também possibilita agregar informações dos nós mais específicos de volta para os nós mais gerais, promovendo uma análise funcional integrada e escalável [3, 6, 17].

Dessa forma, a anotação de proteínas com termos da GO caracteriza-se como um problema de classificação multirrótulo, pois cada proteína pode estar associada simultaneamente a diversos termos em diferentes níveis de especificidade. Nesse contexto, vigora a chamada Regra do Caminho Verdadeiro (*True Path Rule*), que estabelece que, caso uma proteína seja anotada com um termo específico (filho), ela também deve receber todos os termos ancestrais até a raiz, garantindo a consistência hierárquica da anotação. Isto é, a hierarquia de funções na GO reflete relações entre termos gerais (conhecidos como ancestrais) e específicos (filhos). Por exemplo, na categoria de *binding* (ligação), a função geral "binding" engloba todas as atividades de ligação molecular. Um termo mais específico, como "lipid binding" (ligação a lipídios), indica a capacidade de interagir especificamente com lipídios, enquanto "glycolipid binding" (ligação a glicolipídios) refina ainda mais para um tipo particular de lipídio. Essa organização hierárquica permite que informações anotadas para funções específicas sejam propagadas aos termos ancestrais, promovendo análises abrangentes e computacionalmente eficientes [18].

Atualmente, a GO contém um número significativo de termos, que cresce conforme avançam as pesquisas: 10.417 para Função Molecular (MF), 4.022 para Componente Celular (CC) e 29.146 para Processo Biológico (BP), de acordo com a documentação oficial[1].

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

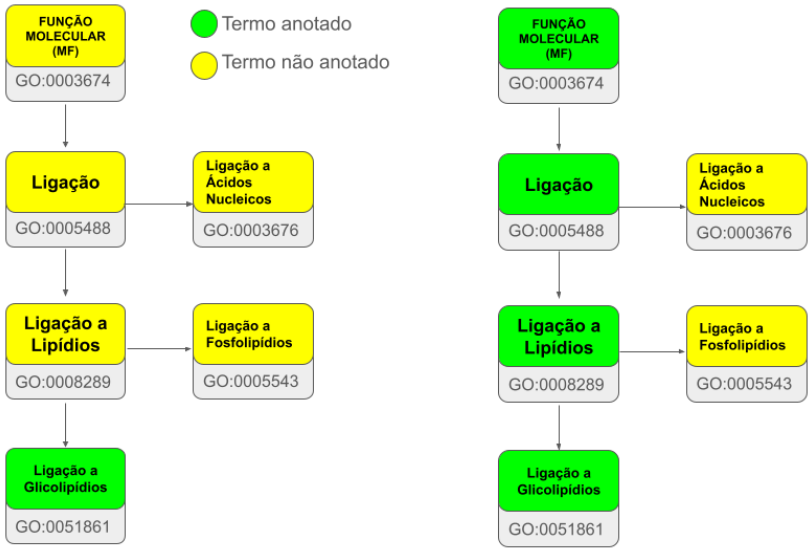

Figura 2.1: Ilustração da Regra do Caminho Verdadeiro com as hierarquias GO representadas como grafos acíclicos direcionados.

2.1.3 Redes de Interações Proteína-Proteína

As proteínas podem interagir de diversas formas, seja por meio de ligações estáveis que resultam em complexos multiproteicos, seja por contatos transitórios que regulam vias de sinalização e processos celulares. Interações proteína-proteína referem-se aos contatos físicos específicos e intencionais estabelecidos entre duas ou mais proteínas para a execução de atividades biológicas determinadas. Essas interações podem ocorrer através de uma variedade de forças e ligações envolvendo interações não covalentes e covalentes. Dentro de um organismo, essas interações podem ocorrer de forma múltipla e específica, originando complexos proteicos, conforme ilustra a Figura 2.2 que desempenham funções cooperativas [19,20]. Técnicas computacionais, aliadas à análise estatística, permitem a construção de redes baseadas em interações proteína-proteína (do inglês, *Protein-Protein Interaction* – PPI). Nestas redes, cada nó representa uma proteína e as arestas indicam as relações funcionais ou físicas entre elas.

Para catalogar e validar essas interações, bases de dados especializadas (como BioGRID[2], IntAct[3] e STRING[4]) reúnem evidências oriundas de diversas fontes, tais como a vizinhança gênica, fusão de genes, coocorrência em diferentes organismos, coexpressão, experimentos laboratoriais (como o ensaio de duplo-híbrido em levedura ou a co-imunoprecipitação), informações curadas e mineração de dados em literatura científica [3, 21]. Assim, as informações consolidadas refletem tanto observações experimentais quanto inferências computacionais, fornecendo um panorama abrangente dos complexos funcionais em que as proteínas podem estar envolvidas [22].

Tome como exemplo o método de co-imunoprecipitação (Co-IP): nele, um anticorpo específico é utilizado para "pescar" uma proteína de interesse em um extrato celular; a co-imunoprecipitação de outras proteínas indica a formação de um complexo físico entre elas. A

[2] `https://thebiogrid.org`
[3] `https://www.ebi.ac.uk/intact`
[4] `https://string-db.org`

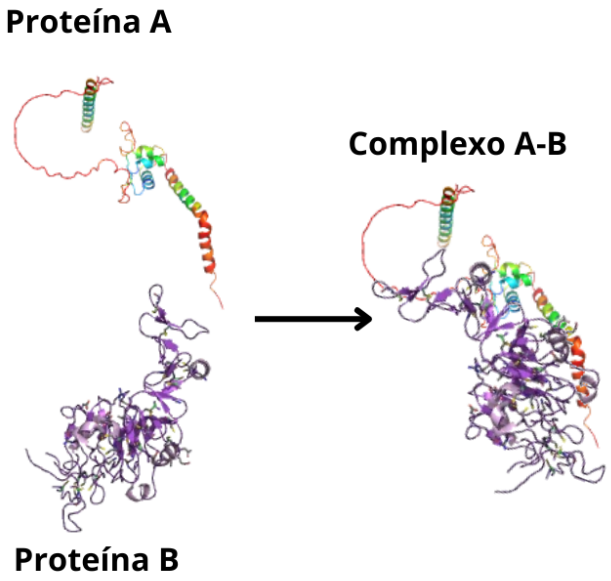

Figura 2.2: Complexo proteico formado pela interação de duas proteínas.

análise sistemática desses relacionamentos, seja por Co-IP ou por outras abordagens (como o ensaio de duplo-híbrido em levedura), permite inferir se as proteínas exercem papéis correlatos ou compartilham vias de regulação. Dessa forma, ao se observar em quais processos as proteínas interagem, torna-se possível identificar padrões que podem revelar funções semelhantes ou complementares [23]. Esse conhecimento é particularmente útil para a anotação de funções proteicas em ontologias, como a GO, pois estabelece correlações entre proteínas bem caracterizadas e aquelas de função desconhecida. Técnicas de aprendizado profundo podem, então, aproveitar esses dados de PPI para aprimorar a previsão e anotação de funções [7, 24].

2.2 Conceitos Computacionais

Nesta seção, apresentamos os fundamentos dos principais métodos computacionais empregados na anotação funcional de proteínas.

2.2.1 Aprendizado Supervisionado e Classificação Multirrótulo

Aprendizado supervisionado é um paradigma de aprendizado de máquina no qual o modelo aprende a partir de exemplos rotulados, buscando identificar um padrão que relacione as variáveis de entrada (características ou atributos) aos rótulos de saída [25]. Em outras palavras, o algoritmo recebe um conjunto de amostras já classificadas ou associadas a um valor de predição conhecido e, a partir disso, ajusta seus parâmetros para que as previsões sobre novos dados sejam as mais precisas possíveis. Esse processo de treinamento implica minimização de uma função de custo, que avalia o quão distantes as predições do modelo estão em relação aos valores verdadeiros.

As tarefas de aprendizado supervisionado são geralmente subdivididas em dois grupos principais:

- Classificação: prediz uma categoria discreta (por exemplo, identificar se um e-mail é *spam* ou *não spam*).

- Regressão: estima um valor numérico contínuo (por exemplo, prever o preço de um imóvel com base em suas características).

Entre os modelos clássicos de aprendizado supervisionado, destacam-se as redes neurais artificiais, que ajustam pesos e vieses em várias camadas para mapear entradas em saídas, e as máquinas de vetores de suporte (do inglês, *Support Vector Machines* – SVMs), que buscam separar os dados em espaços de alta dimensão por meio de hiperplanos ótimos. Árvores de decisão e métodos de *ensemble* (por exemplo, florestas aleatórias) também são amplamente utilizados, cada um com vantagens específicas em termos de interpretabilidade, velocidade de treinamento e capacidade de generalização.

No caso de classificação multirrótulo, cada amostra pode estar simultaneamente associada a diversos rótulos, tornando o problema mais complexo em relação à classificação tradicional de rótulo único [26]. Em tais cenários, o modelo deve capturar não apenas os padrões que distinguem cada rótulo individualmente, mas também as relações de dependência entre os diferentes rótulos. Aplicações típicas incluem sistemas de recomendação, classificação de textos (onde um documento pode abordar vários temas) e, em bioinformática, a anotação funcional de proteínas, em que cada proteína pode exercer múltiplas funções.

2.2.2 Redes Neurais e Aprendizado Profundo

As redes neurais foram desenvolvidas com inspiração na ideia vigente à época de como os neurônios humanos funcionam, baseando-se no modelo clássico do perceptron. Nesse modelo, cada neurônio artificial recebe entradas ponderadas — onde os pesos das conexões, que ligam os neurônios entre si, representam a influência de cada sinal — e um termo de *bias* é adicionado para ajustar a ativação do neurônio, permitindo a modelagem de funções de decisão mais flexíveis. Esses neurônios são organizados em camadas, e redes neurais profundas (*deep neural networks*) se distinguem por possuírem múltiplas camadas ocultas, o que possibilita a extração de representações cada vez mais complexas e abstratas dos dados. No entanto, esse aumento na profundidade e na quantidade de parâmetros (incluindo os pesos e os *bias*) eleva significativamente o custo computacional, tanto em termos de memória quanto de tempo de treinamento.

Durante o processo de treinamento, os algoritmos de otimização, como o gradiente descendente, ajustam esses parâmetros iterativamente para minimizar a função de perda, mas a complexidade desses ajustes em redes profundas exige, frequentemente, recursos computacionais robustos e estratégias avançadas de regularização para evitar problemas como o sobreajuste (do inglês, *overfitting*) [27].

Perceptron Simples e a Fronteira de Decisão

O perceptron simples é a unidade de processamento fundamental em diversas arquiteturas neurais. Ele executa uma combinação linear dos atributos de entrada seguida de uma função de ativação, definindo uma fronteira de decisão (*decision boundary*). Em problemas de classificação binária, essa fronteira separa duas classes de forma que, se a soma ponderada dos atributos exceder determinado limiar, a instância é classificada como positiva; caso contrário, como negativa. Matematicamente, se representarmos os atributos de entrada por

$\mathbf{x} = (x_1, x_2, \ldots, x_n)$, o perceptron calcula:

$$\hat{y} = \sigma\big(\mathbf{w}^\top \mathbf{x} + b\big),$$

onde $\mathbf{w}$ são os pesos, b é o *bias* e $\sigma(\cdot)$ é a função de ativação (por exemplo, sign ou ReLU, dependendo da tarefa), conforme ilustrado na Figura 2.3. A fronteira de decisão, nesse caso, é o hiperplano definido por $\mathbf{w}^\top \mathbf{x} + b = 0$.

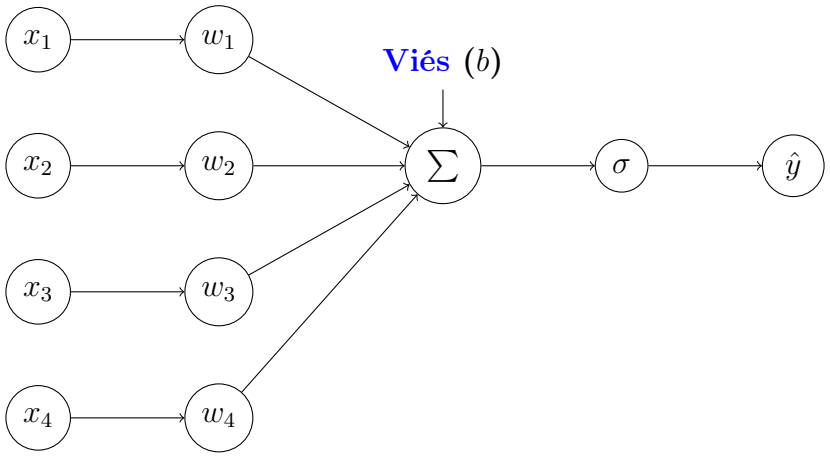

Figura 2.3: Esquema de um Perceptron Simples.

Multi-Layer Perceptrons

Os perceptrons multicamadas (do inglês, *Multi-Layer Perceptrons* – MLPs) estendem o conceito do perceptron simples ao utilizar múltiplas camadas de neurônios (camadas densas). Cada neurônio em um MLP funciona como um Perceptron que aplica uma função de ativação não linear (por exemplo, ReLU, sigmoíde ou tangente hiperbólica). A saída de um neurônio j na camada l pode ser escrita como:

$$z_j^{(l)} = \sigma\left(\sum_{i=1}^{n_{l-1}} w_{ij}^{(l)}\, a_i^{(l-1)} + b_j^{(l)}\right),$$

onde $w_{ij}^{(l)}$ e $b_j^{(l)}$ são, respectivamente, os pesos e o viés que conectam o neurônio i na camada anterior ao neurônio j na camada l, $a_i^{(l-1)}$ é a ativação na camada anterior, e $\sigma(\cdot)$ é a função de ativação. Conforme ilustra a Figura 2.4.

O treinamento dessas redes é realizado por meio do algoritmo de retropropagação de erro (*backpropagation*), introduzido por Rumelhart, Hinton e Williams [28]. Após a etapa de *forward pass*, em que a saída da rede é calculada, obtém-se uma função de custo (por exemplo, entropia cruzada). Em seguida, calculam-se os gradientes do erro em relação a cada parâmetro (pesos e vieses) por meio de derivadas parciais, propagando esses gradientes de trás para frente. Essa informação é usada para atualizar os parâmetros de modo a reduzir o erro.

A atualização de parâmetros ocorre, tipicamente, por descida de gradiente estocástica (ou

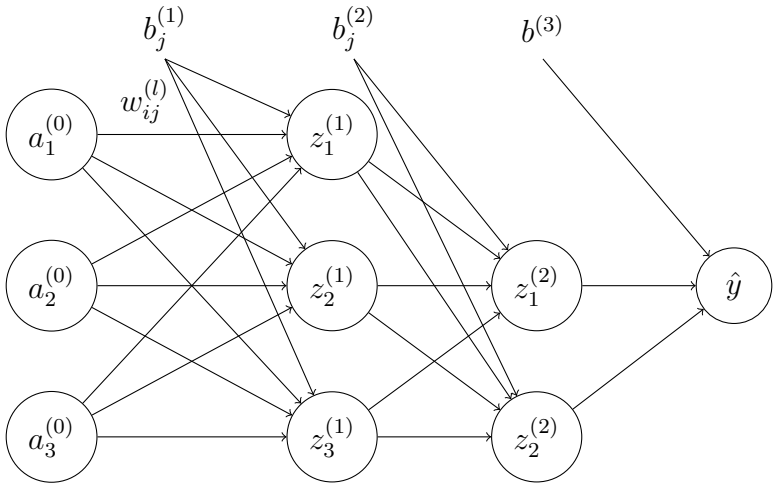

Figura 2.4: Arquitetura de um MLP com duas camadas ocultas.

uma de suas variações), conforme a expressão:

$$\theta \leftarrow \theta - \eta \nabla_\theta J(\theta),$$

onde θ representa os parâmetros (pesos e vieses), η (taxa de aprendizado) controla a intensidade da atualização e $J(\theta)$ é a função de custo. Métodos como *Adam* [29], RMSProp [30] e Adagrad [31] introduzem ajustes adaptativos na taxa de aprendizado, acelerando a convergência para um classificador que minimize o custo.

Em MLPs, hiperparâmetros como número de camadas e neurônios, taxa de aprendizado, função de ativação e número de épocas de treinamento podem ser definidos manualmente ou ajustados ao longo do processo de treinamento. A escolha adequada desses valores afeta diretamente o desempenho do modelo [27]. Em geral, um número maior de camadas e neurônios aumenta o poder de representação, mas pode levar a *overfitting* e exigir maior poder computacional. Além disso, uma taxa de aprendizado muito alta faz o treinamento divergir, enquanto taxas muito baixas o tornam excessivamente lento.

Para evitar sobreajuste, pode-se lançar mão de diversas estratégias de regularização — como regularização L1/L2, *dropout*, augmentação de dados (quando aplicável) e *early stopping* — em combinação com uma estratégias de busca de hiperparâmetros (por exemplo, *grid search*, *random search* ou *Bayesian optimization*), de modo a equilibrar a capacidade de representação e o poder de generalização da rede.

Redes Convolucionais

As Redes Convolucionais (do inglês, *Convolutional Neural Networks* – CNNs) introduziram o mecanismo de convolução para explorar estruturas locais em dados, particularmente imagens [32]. Em vez de camadas densas, as CNNs utilizam filtros que percorrem a entrada, extraindo padrões em diferentes escalas. Em um nível mais técnico, cada camada convolucional recebe como entrada um tensor de três dimensões (altura, largura e profundidade), aplicando filtros — também representados por tensores 3D, cujo tamanho espacial (altura e largura) é menor que o da entrada, mas cujo canal de profundidade coincide ou se relaciona ao número de canais da entrada. À medida que o filtro se desloca (ou "convolui")

sobre as dimensões espaciais da entrada, realiza multiplicações ponto a ponto entre os valores do filtro e o *patch* correspondente na entrada, produzindo um mapa de características bidimensional. Cada filtro corresponde, portanto, a um conjunto de pesos aprendíveis que a rede ajusta de forma a detectar padrões específicos (por exemplo, bordas, texturas ou outras características) [33].

A profundidade do tensor de saída nessa camada depende do número de filtros empregados, pois cada filtro gera um mapa de características distinto. Após as camadas convolucionais, costumam-se empilhar blocos convolucionais, que são sequências de convoluções, funções de ativação e operações de normalização, intercaladas ou seguidas por camadas de *pooling* (por exemplo, *max-pooling* ou *average-pooling*). Essas operações reduzem a dimensionalidade das representações, auxiliando no controle do número de parâmetros e mitigando riscos de *overfitting*. Por fim, no topo da rede, isto é, na saída do modelo, encontra-se uma camada densa totalmente conectada que agrega as características extraídas pelas camadas anteriores e realiza a classificação ou regressão final, como mostra o esquema na Figura 2.5.

Além das redes convolucionais tradicionais, também existem Redes Convolucionais 1D, que são projetadas para trabalhar com dados organizados em uma única dimensão, como séries temporais ou sequências genômicas. Essas redes utilizam filtros 1D que se deslocam apenas ao longo de uma dimensão, tornando-as adequadas para capturar padrões locais em dados sequenciais [34].

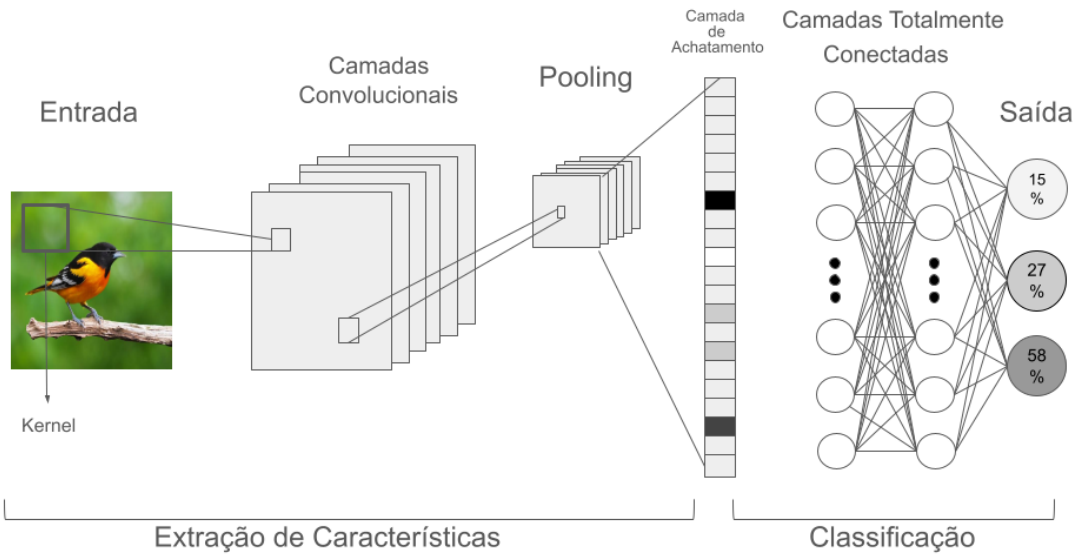

Figura 2.5: *Backbone* de uma rede convolucional, ilustrando a passagem de filtros convolucionais, formação dos mapas de características, os *poolings* e a etapa de classificação.

Arquiteturas avançadas, como *ResNet* [35] e *DenseNet* [36], empregam blocos residuais e conexões de atalho (*skip connections*) para aliviar problemas relacionados ao desaparecimento de gradiente em redes muito profundas. Além disso, redes modernas, como a MobileNetV2 [37, 38], utilizam artifícios avançados como blocos residuais invertidos (*inverted residual blocks*) e convoluções separáveis em profundidade (*depthwise separable convolutions*), que reduzem significativamente a complexidade computacional sem comprometer o desempenho.

Transfer Learning

A técnica de *transfer learning* aproveita redes previamente treinadas em tarefas de grande escala (por exemplo, ImageNet [39]) para acelerar o treinamento em uma nova tarefa. Nesse processo, as primeiras camadas (que capturam características gerais) são mantidas e apenas as camadas finais são ajustadas (esse procedimento de continuar o treinamento de uma rede já pré-treinada é conhecido como *fine-tuning*). Arquiteturas como *ResNet* [40], *EfficientNet* [41] e *ConvNeXt* [42] são frequentemente empregadas em cenários de *transfer learning*, principalmente por sua eficiência, barateando o treinamento do modelo e permitindo o "descongelamento" de um número maior de camadas por mais épocas ou com conjuntos de dados mais extensos.

Graph Convolutional Neural Networks

As redes convolucionais de grafos (do inglês, *Graph Convolutional Neural Networks* – GCNNs) estendem o conceito de convolução para dados estruturados em grafos [43]. Em vez de aplicar filtros em janelas de pixels, as GCNNs propagam e agregam informações dos nós vizinhos:

$$h_v^{(l+1)} = \sigma\Big(W^{(l)} \cdot \mathrm{AGG}\{\, h_u^{(l)} : u \in N(v) \cup \{v\}\}\Big),$$

onde $h_v^{(l)}$ é a representação do nó v na camada l, $N(v)$ é o conjunto de nós vizinhos de v no grafo, $\mathrm{AGG}\{\cdot\}$ é alguma função de agregação (por exemplo, soma ou média) e $W^{(l)}$ são os pesos. Em aplicações de interações proteína–proteína, cada nó representa uma proteína, e as arestas denotam interações (físicas, funcionais ou preditas) entre elas. Assim, as GCNNs podem ajudar a capturar padrões topológicos úteis para a predição de funções proteicas.

2.2.3 Aprendizado Autossupervisionado

O aprendizado autossupervisionado (ou *self-supervised learning*) [44] é uma abordagem que permite extrair representações ricas de dados não rotulados, definindo tarefas auxiliares que induzem o modelo a aprender características intrínsecas dos dados. Essas tarefas são projetadas para que os rótulos sejam gerados automaticamente a partir da própria estrutura dos dados, sem a necessidade de anotação manual. Exemplos comuns incluem reconstrução de dados, previsão de partes ausentes e aprendizado contrastivo, onde o modelo aprende a diferenciar entre pares de exemplos semelhantes e não semelhantes [43, 45–47].

No contexto de reconstrução, uma arquitetura frequentemente utilizada é o *autoencoder*, que consiste em duas partes principais: um *encoder*, responsável por mapear os dados de entrada para uma representação latente de dimensionalidade reduzida, e um *decoder*, que reconstrói os dados originais a partir dessa representação comprimida. A Figura 2.6 ilustra um exemplo de diagrama de autoencoder, destacando o processo de codificação e decodificação.

Ao treinar o autoencoder para reconstruir os dados de entrada, o *encoder* é forçado a descartar redundâncias e reter apenas as informações mais relevantes, resultando em representações latentes robustas. Essas representações podem ser utilizadas em tarefas subsequentes, como a anotação funcional de proteínas em redes de interações proteína-proteína (PPI). Por exemplo, o autoencoder pode ser treinado para reconstruir um vetor de interações de cada proteína, gerando *embeddings* que capturam padrões estruturais importantes da rede [47].

Outra abordagem comum no aprendizado autossupervisionado é o aprendizado contrastivo. Nesse caso, o modelo recebe pares de exemplos positivos (por exemplo, proteínas que

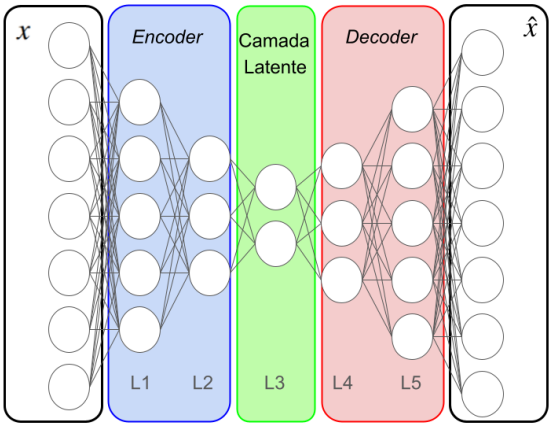

Figura 2.6: Exemplo de arquitetura de autoencoder, evidenciando a etapa de codificação (redução dimensional) e a posterior fase de decodificação (reconstrução).

apresentam padrões de interação semelhantes na rede) e pares negativos (sem sobreposição de interações ou com topologias de rede significativamente diferentes). O objetivo é maximizar a similaridade entre pares positivos e minimizá-la para pares negativos, resultando em um espaço de *embeddings* onde amostras semelhantes estejam próximas umas das outras, enquanto amostras diferentes sejam afastadas [46].

Vale destacar que o aprendizado autossupervisionado é frequentemente utilizado como uma etapa de pré-treinamento para abordagens supervisionadas. Nesse cenário, as representações aprendidas de forma autossupervisionada são refinadas em tarefas específicas com dados rotulados, permitindo que o modelo aproveite tanto a riqueza de informações intrínsicas extraídas de grandes volumes de dados não rotulados quanto a precisão fornecida por anotações supervisionadas.

2.3 Trabalhos Correlatos

No Apêndice A, apresentamos os principais trabalhos correlatos à pesquisa em investigação, com ênfase em abordagens para anotação funcional de proteínas baseadas em redes de interações, além de outros estudos que se relacionam às estratégias de aprendizado profundo discutidas neste capítulo.

Capítulo 3

Material e Métodos

Este capítulo descreve a metodologia proposta, as bases de dados utilizadas, as métricas de avaliação e os recursos computacionais empregados no desenvolvimento do projeto.

3.1 Metodologia

Esta seção apresenta a metodologia proposta para a predição de funções de proteínas, utilizando três representações distintas dos dados de interações proteína-proteína (PPI): tabular, imagem e grafo. A proposta contempla tanto abordagens supervisionadas quanto estratégias que incorporam pré-treinamento autossupervisionado como etapa complementar. Cada domínio da ontologia (*Biological Process*, *Molecular Function* e *Cellular Component*) é tratado de forma independente, gerando um classificador específico para cada ontologia, pois os domínios apresentam características biológicas e estruturais distintas. Essa segmentação permite a seleção de características e a definição de arquiteturas de rede ou classificadores mais adequados a cada tipo de dado, facilitando o ajuste fino dos parâmetros e a integração dos resultados, além de possibilitar uma comparação consistente com métodos do estado da arte que também segmentam os dados de PPI.

3.1.1 Pré-processamento

Inicialmente, os dados brutos de interações são organizados em um vetor de pontuações, que representa as evidências de interação entre a proteína-alvo e as proteínas presentes na base de treinamento. Essa etapa pode envolver procedimentos básicos de normalização, remoção de ruído e padronização dos valores, de forma a garantir que as representações extraídas (tabular, imagem e grafo) sejam consistentes e compatíveis para as etapas subsequentes, como ilustrado na Figura 3.1.

3.1.2 Anotação Funcional

A etapa de anotação funcional consiste em atribuir termos da Ontologia Genética (GO) a cada proteína, tendo como ponto de partida os dados baseados nas pontuações de evidência de interação entre a proteína-alvo e as proteínas da base de treinamento. As anotações já existentes nas proteínas de treino são combinadas e, de forma inicial, essas pontuações servem para inferir os possíveis termos da proteína-alvo. Em seguida, cada termo é propagado aos seus ancestrais na hierarquia da GO, garantindo a coerência das funções atribuídas.

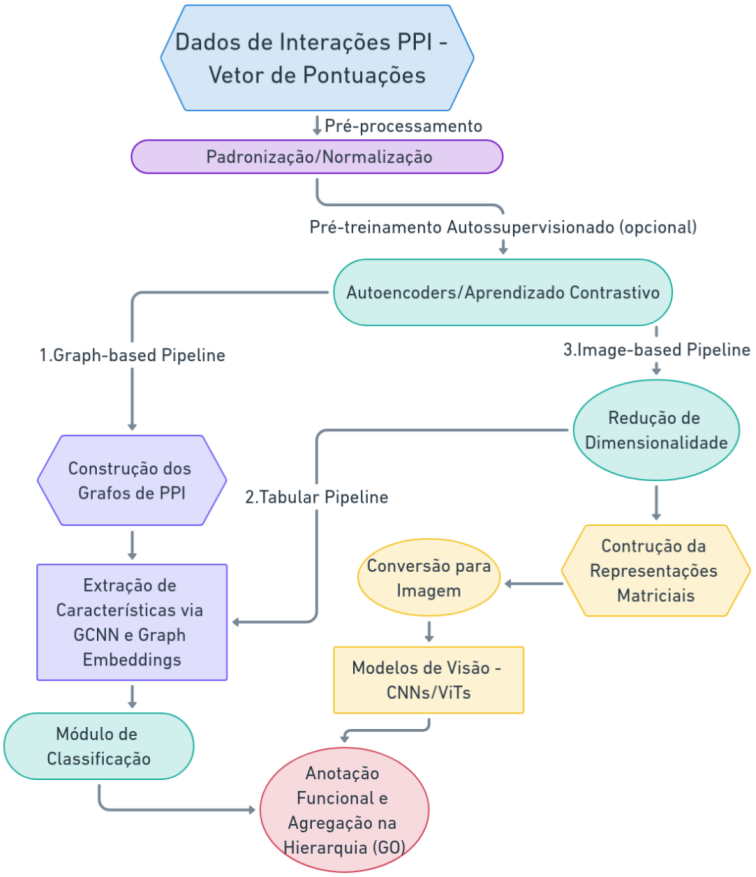

Figura 3.1: Fluxograma das abordagens para as diferentes representações de redes PPI. A etapa opcional de pré-treinamento autossupervisionado pode ser integrada às abordagens de grafos, imagens e dados tabulares.

3.1.3 Representação dos Dados

Todas as representações utilizadas neste trabalho são extraídas do vetor de pontuação de interações da proteína-alvo com as proteínas da base de treinamento. Serão investigados três formatos de representação:

- **Tabular:** Cada proteína é mapeada para um vetor numérico contendo as pontuações de interação com as proteínas do conjunto de treino. Esse vetor pode ser reduzido por técnicas de *pooling* ou seleção de atributos.

- **Imagem:** A partir do mesmo vetor de interações, são geradas matrizes — por exemplo, utilizando a diferença absoluta entre os componentes do vetor (após redução dimensional) e seu transposto — que podem ser interpretadas como imagens em tons de cinza e convertidas para tensores.

- **Grafo:** As proteínas são representadas como nós em uma rede PPI, em que as arestas são ponderadas pelas pontuações de interação, permitindo explorar explicitamente a estrutura topológica dos dados. Nesta representação, podem ser aplicadas técnicas de *graph embedding*, que convertem a estrutura do grafo em vetores que sintetizam as

características topológicas de forma adequada para o treinamento dos modelos. Iremos adotar também representações por proteína a partir de características biológicas, propagadas usando a estrutura de grafos.

3.1.4 Abordagens Supervisionadas

Nas abordagens supervisionadas, são utilizadas arquiteturas específicas para cada representação dos dados:

- **Representação Tabular:** São empregados modelos como **MLPs** ou **Random Forests**, treinados diretamente sobre os vetores numéricos correspondentes às pontuações de interação.
- **Representação em Imagem:** Utilizam-se **CNNs** e **Vision Transformers (ViTs)**, com possibilidade de *transfer learning* a partir de modelos pré-treinados, ajustando as camadas finais para a tarefa de anotação funcional (ou seja, classificação multirrótulo).
- **Representação em Grafo:** O treinamento é conduzido por meio de **Graph Convolutional Networks (GCNs)** nos dados obtidos com técnicas de *graph embedding*. Nesta *pipeline*, após a extração de *features* via GCNN, integra-se um módulo de classificação (por exemplo, um MLP ou outro modelo supervisionado) que utiliza essas features para a predição final. Esse fluxo está ilustrado no Fluxo de Processamento dos Dados em Grafos (Figura 3.2).

Além do treinamento individual de cada modelo supervisionado, exploramos a possibilidade de realizar técnicas de *ensemble* que combinem as abordagens supervisionadas com as representações provenientes do pré-treinamento autossupervisionado. Essa estratégia de integração é tecnicamente justificada, pois os atributos refinados obtidos via pré-treinamento podem complementar as informações extraídas diretamente dos dados, melhorando a robustez e a capacidade de generalização do classificador final.

3.1.5 Abordagem Autossupervisionada

Para enriquecer as representações e potencialmente melhorar os resultados, são investigadas estratégias de aprendizado autossupervisionado ou semi-supervisionado. Essas estratégias podem ser integradas a qualquer uma das três pipelines principais e incluem:

- **Autoencoders para Vetores de Interação:** Treina-se um autoencoder para reconstruir as pontuações de interação, de modo que a camada intermediária extraia um *embedding* de dimensão reduzida que capta a estrutura subjacente dos dados.
- **Graph Embeddings:** Técnicas como *Node2Vec* ou *Graph2Vec* geram vetores que sintetizam as características topológicas do grafo PPI e de informações biologicas das proteínas (nós) das redes.
- **Aprendizado Contrastivo:** Essa abordagem pode maximizar a similaridade entre pares de proteínas funcionalmente próximas (ou conectadas no grafo) e minimizar a similaridade entre pares sem relação aparente, incentivando a descoberta de estruturas latentes.

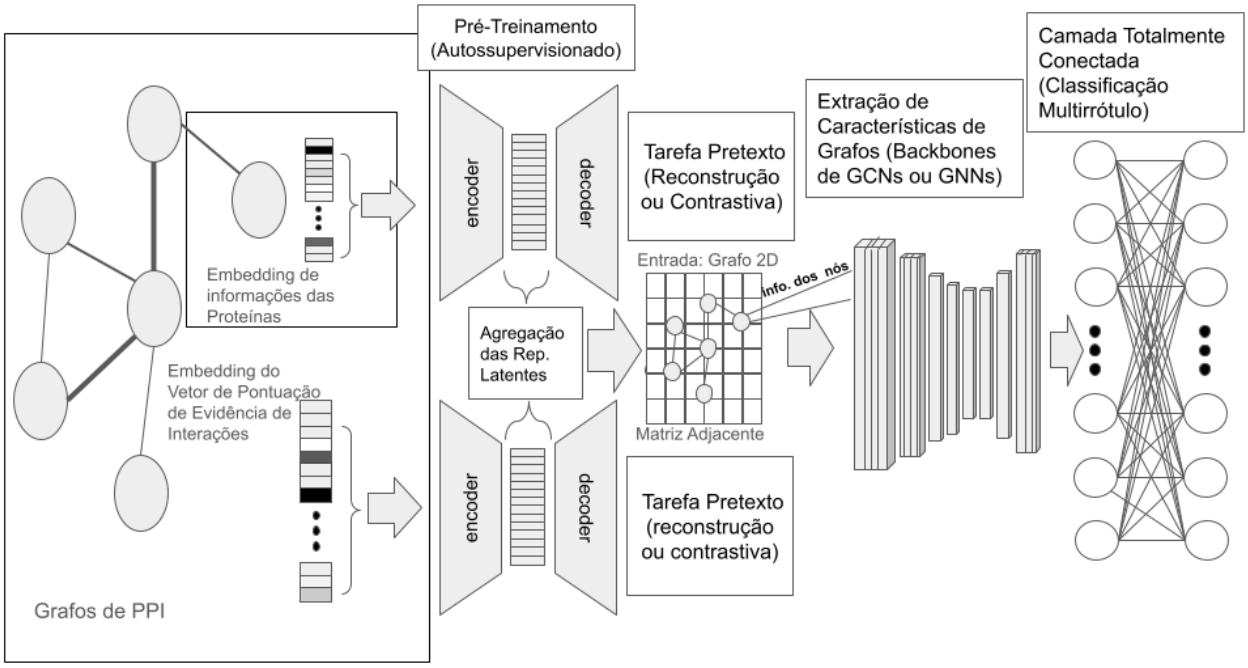

Figura 3.2: Fluxograma da Abordagem em Grafos: após a construção do grafo de interações proteína-proteína (PPI) e a obtenção dos embeddings adequados, são extraídas features utilizando modelos baseados em grafos, como GCNs. Por fim, um módulo de classificação, como um MLP, é aplicado para anotar as funções.

Essas técnicas podem ser utilizadas como etapa de pré-treinamento ou de forma complementar ao modelo supervisionado, fornecendo atributos adicionais que aprimoram a predição das funções. A Figura 3.2 ilustra o fluxo integrando a extração de *embeddings* autossupervisionados a abordagens supervisionadas.

3.1.6 Treinamento, Teste e Comparação

Após a obtenção das representações finais – seja por meio das abordagens supervisionadas ou pela integração das estratégias autossupervisionadas — cada ontologia (*BP*, *MF* e *CC*) é tratada de forma independente, com a construção de um classificador específico para a predição dos termos da GO em novas proteínas. Na fase de teste, esses classificadores são avaliados em um conjunto reservado (20% dos dados) e suas predições são comparadas com as anotações verdadeiras, considerando também a propagação dos termos aos seus ancestrais, conforme a hierarquia da GO.

Com isso, o objetivo é comparar as abordagens (tabular, imagem e grafo) e estratégias de aprendizado (supervisionadas versus aquelas com suporte autossupervisionado) em termos de desempenho e complexidade de implementação, permitindo identificar a metodologia que melhor equilibra esses aspectos. Os melhores resultados serão comparados com metodologias de ponta que exploram dados PPI e, posteriormente, o modelo com melhor desempenho será integrado com método baseado em sequência desenvolvido pelo grupo de pesquisa do laboratório [14].

3.2 Bases de Dados

Para avaliar e comparar os métodos propostos com os da literatura, utilizaremos as proteínas da coleção de dados fornecida pelo *Critical Assessment of Protein Function Annotation* (CAFA)[1], um desafio periódico focado na predição de funções proteicas. As anotações funcionais serão obtidas da *Gene Ontology* (GO) [5], considerando 500 termos de Processo Biológico (BP), 498 de Função Molecular (MF) e 498 de Componente Celular (CC). Além disso, as redes de interações proteína-proteína (PPI) extraídas da STRING permitirão integrar informações das proteínas da competição CAFA5 ao estudo. Detalhes dessas bases de dados, assim como uma breve análise exploratória, são apresentados no Apêndice B.

3.3 Métricas de Avaliação

Para mensurar o desempenho dos métodos e compará-los com trabalhos da literatura, adotamos as métricas $\mathbf{F}_{\max}$, $\mathbf{S}_{\min}$, **AuPRC** e **IAuPRC**, cujas definições e formulações estão detalhadas no Apêndice C.

3.4 Recursos Computacionais

A implementação deste projeto será realizada em Python, pela ampla disponibilidade e excelente documentação de bibliotecas científicas e de aprendizado de máquina. Faremos uso de pacotes para funções numéricas, aprendizado profundo, aprendizado de máquina tradicional e visualização de dados. Dentre as principais bibliotecas que utilizaremos, destacam-se NumPy[2], PyTorch[3], scikit-learn[4] e Matplotlib[5], que oferecem suporte abrangente para tarefas de modelagem, processamento e análise de dados.

Para a realização dos experimentos, adotaremos duas principais plataformas de execução. Primeiramente, utilizamos o Google Colab, que disponibiliza tanto CPUs convencionais quanto GPUs NVIDIA T4, viabilizando protótipos rápidos e experimentos em escala moderada. Além disso, pretendemos utilizar o CENAPAD[6] para a execução de experimentos. Assim, a combinação das duas infraestruturas fornece flexibilidade de recursos e escalabilidade durante o desenvolvimento e a execução deste projeto.

3.5 Resultados Preliminares

No Apêndice D, apresentamos os resultados preliminares obtidos na fase inicial de experimentos desta pesquisa, detalhando a execução dos testes e discutindo suas implicações, bem como possíveis adaptações futuras para conduzir novos experimentos.

[1] `https://biofunctionprediction.org/cafa`
[2] `https://numpy.org/doc/`
[3] `https://pytorch.org/docs/stable/`
[4] `https://scikit-learn.org/stable/documentation.html`
[5] `https://matplotlib.org/stable/contents.html`
[6] `https://www.cenapad.unicamp.br`

Capítulo 4

Plano de Trabalho e Cronograma de Execução

O plano de trabalho é composto pelas seguintes atividades:

1. Estudo e análise das principais técnicas e abordagens de aprendizado profundo disponíveis na literatura para anotação funcional de proteínas utilizando dados de redes PPI, incluindo métodos baseados em grafos, imagens e aprendizado autossupervisionado.
2. Preparação da base de dados de interações proteína-proteína (PPI), abrangendo a coleta, processamento e integração de dados provenientes de fontes públicas.
3. Desenvolvimento de abordagens supervisionadas para anotação funcional de proteínas, explorando diferentes representações de dados de interações PPI, com ênfase em abordagens baseadas em grafos.
4. Realização de testes e validação das abordagens supervisionadas, utilizando métricas apropriadas para avaliar o desempenho e a eficácia dos modelos propostos.
5. Desenvolvimento de abordagens utilizando pré-treinamento autossupervisionado para melhorar a anotação funcional de proteínas, explorando diferentes estratégias de aprendizado de representações.
6. Realização de testes e validação das abordagens com pré-treinamento autossupervisionado, comparando os resultados obtidos com as abordagens supervisionadas.
7. Comparação dos resultados obtidos com outros trabalhos disponíveis na literatura, visando identificar avanços e limitações das abordagens propostas.
8. Documentação e publicação dos resultados, incluindo a elaboração de artigos científicos para divulgação em periódicos e conferências da área.
9. Escrita do documento da dissertação, integrando todos os componentes do trabalho realizado.
10. Defesa da dissertação de mestrado.

O cronograma de execução das atividades propostas, divididas em 5 etapas, em um prazo de 24 meses, é apresentado na Tabela 4.1.

Atividades	**1º ano**						**2º ano**					
	1	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**
Etapa 1 - Preparação												
Estudo e análise das técnicas												
Preparação da base de dados												
Etapa 2 - Abordagens supervisionadas												
Desenvolvimento dos modelos												
Realização de testes												
Etapa 3 - Abordagens com Pré-Treinamento Autossupervisionado												
Desenvolvimento dos modelos												
Realização de testes												
Etapa 4 - Comparação e Análise												
Comparação dos resultados												
Etapa 5 - Conclusão												
Documentação e publicação												
Escrita da dissertação												
Defesa da dissertação												

Tabela 4.1: Cronograma de atividades dividido em bimestres.

Bibliografia

[1] David P. Clark and Nanette J. Pazdernik. *Molecular Biology*. 2012. 1

[2] Donald Voet, Judith G. Voet, and Charlotte W. Pratt. *Principles of Biochemistry*. 2008. 1, 4

[3] Nahid Safari-Alighiarloo, Mohammad Taghizadeh, Mostafa Rezaei-Tavirani, Bahram Goliaei, and Ali Asghar Peyvandi. Protein-Protein Interaction Networks (PPI) and Complex Diseases. *Gastroenterology and Hepatology from Bed to Bench*, 7(1):17–31, 2014. 1, 5, 6

[4] Alexei Vazquez, Alessandro Flammini, Amos Maritan, and Alessandro Vespignani. Global Protein Function Prediction from Protein-Protein Interaction Networks. *Nature Biotechnology*, 21(6):697–700, 2003. 1

[5] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25(1):25–29, 2000. 1, 5, 18

[6] Christophe Dessimoz, Nives Škunca, and Paul D. Thomas. CAFA and the Open World of Protein Function Predictions. *Trends in Genetics*, 29(11):609–610, 2013. 1, 5

[7] Gavin C. K. W. Koh, Pablo Porras, Bruno Aranda, Henning Hermjakob, and Sandra E. Orchard. Analyzing Protein-Protein Interaction Networks. *Journal of Proteome Research*, 11(4):2014–2031, 2012. 2, 7

[8] Loïc Royer, Matthias Reimann, Bill Andreopoulos, and Michael Schroeder. Unraveling Protein Networks with Power Graph Analysis. *PLoS Computational Biology*, 4(7):e1000108, 2008. 2

[9] Ana Paula S Dantas, Gabriel Bianchin de Oliveira, Daiane Mendes de Oliveira, Helio Pedrini, Cid C. de Souza, and Zanoni Dias. Algorithmic Fairness Applied to the Multi-Label Classification Problem. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, pages 737–744, 2023. 2

[10] Naihui Zhou, Yuxiang Jiang, Timothy R. Bergquist, Alexandra J. Lee, Balint Z. Kacsoh, Alex W. Crocker, Kimberley A. Lewis, George Georghiou, Huy N. Nguyen, Md Nafiz Hamid, Larry Davis, Tunca Dogan, Volkan Atalay, Ahmet S. Rifaioglu, Alperen Dalkıran, Rengul Cetin Atalay, Chengxin Zhang, Rebecca L. Hurto, Peter L. Freddolino, Yang Zhang, Prajwal Bhat, Fran Supek, José M. Fernández, Branislava Gemovic, Vladimir R. Perovic, Radoslav S. Davidović, Neven Sumonja, Nevena Veljkovic, Ehsaneddin Asgari, Mohammad R.K. Mofrad, Giuseppe Profiti, Castrense Savojardo, Pier Luigi Martelli, Rita Casadio, Florian Boecker, Heiko Schoof, Indika Kahanda, Natalie Thurlby, Alice C. McHardy, Alexandre Renaux, Rabie Saidi, Julian Gough, Alex A. Freitas, Magdalena Antczak, Fabio Fabris, Mark N. Wass, Jie Hou, Jianlin Cheng, Zheng Wang, Alfonso E. Romero, Alberto Paccanaro, Haixuan Yang, Tatyana Goldberg, Chenguang Zhao, Liisa Holm, Petri Törönen, Alan J. Medlar, Elaine Zosa, Itamar Borukhov, Ilya Novikov, Angela Wilkins, Olivier Lichtarge, Po-Han Chi, Wei-Cheng Tseng, Michal Linial, Peter W. Rose, Christophe Dessimoz, Vedrana Vidulin, Saso Dzeroski, Ian Sillitoe, Sayoni Das, Jonathan Gill Lees, David T. Jones, Cen Wan, Domenico Cozzetto, Rui Fa, Mateo Torres, Alex Warwick Vesztrocy, Jose Manuel Rodriguez, Michael L. Tress, Marco Frasca, Marco Notaro, Giuliano Grossi, Alessandro Petrini, Matteo Re, Giorgio Valentini, Marco Mesiti, Daniel B. Roche, Jonas Reeb, David W. Ritchie, Sabeur Aridhi, Seyed Ziaeddin Alborzi, Marie-Dominique Devignes, Da Chen Emily Koo, Richard Bonneau, Vladimir Gligorijević, Meet Barot, Hai Fang, Stefano Toppo, Enrico Lavezzo, Marco Falda, Michele Berselli, Silvio C.E. Tosatto, Marco Carraro, Damiano Piovesan, Hafeez Ur Rehman, Qizhong Mao, Shanshan Zhang, Slobodan Vucetic, Gage S. Black, Dane Jo, Erica Suh, Jonathan B. Dayton, Dallas J. Larsen, Ashton R. Omdahl, Liam J. McGuffin, Danielle A. Brackenridge, Patricia C. Babbitt, Jeffrey M. Yunes, Paolo Fontana, Feng Zhang, Shanfeng Zhu, Ronghui You, Zihan Zhang, Suyang Dai, Shuwei Yao, Weidong Tian, Renzhi Cao, Caleb Chandler, Miguel Amezola, Devon Johnson, Jia-Ming Chang, Wen-Hung Liao, Yi-Wei Liu, Stefano Pascarelli, Yotam Frank, Robert Hoehndorf, Maxat Kulmanov, Imane Boudellioua, Gianfranco Politano, Stefano Di Carlo, Alfredo Benso, Kai Hakala, Filip Ginter, Farrokh Mehryary, Suwisa Kaewphan, Jari Björne, Hans Moen, Martti E.E. Tolvanen, Tapio Salakoski, Daisuke Kihara, Aashish Jain, Tomislav Šmuc, Adrian Altenhoff, Asa Ben-Hur, Burkhard Rost, Steven E. Brenner, Christine A. Orengo, Constance J. Jeffery, Giovanni Bosco, Deborah A. Hogan, Maria J. Martin, Claire O'Donovan, Sean D. Mooney, Casey S. Greene, Predrag Radivojac, and Iddo Friedberg. The CAFA Challenge Reports Improved Protein Function Prediction and New Functional Annotations for Hundreds of Genes Through Experimental Screens. *Genome Biology*, 20(1):1–23, 2019. 2, 33

[11] Maxat Kulmanov and Robert Hoehndorf. DeepGOPlus: Improved Protein Function Prediction from Sequence. *Bioinformatics*, 36(2):422–429, 2020. 2

[12] Ronghui You, Shuwei Yao, Hiroshi Mamitsuka, and Shanfeng Zhu. DeepGraphGO: Graph Neural Network for Large-Scale, Multispecies Protein Function Prediction. *Bioinformatics*, 37:i262–i271, 2021. 2, 29

[13] Haris Hasic, Emir Buza, and Amila Akagic. A Hybrid Method for Prediction of Protein Secondary Structure Based on Multiple Artificial Neural Networks. In *International*

Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pages 1195–1200, 2017. 2

[14] Gabriel Bianchin de Oliveira, Helio Pedrini, and Zanoni Dias. SUPERMAGO: Protein Function Prediction Based on Transformer Embeddings. *Proteins: Structure, Function, and Bioinformatics*, Early View:1–22, 2024. 2, 17

[15] U Satyanarayana. *Biochemistry*. 2013. 4

[16] Christian B. Anfinsen. Principles That Govern the Folding of Protein Chains. *Science*, 181(4096):223–230, 1973. 4

[17] Gabriel B. Oliveira, Helio Pedrini, and Zanoni Dias. TEMPROT: Protein Function Annotation using Transformers Embeddings and Homology Search. *BMC Bioinformatics*, 24(1):242, 2023. 5

[18] Giorgio Valentini. True Path Rule Hierarchical Ensembles for Genome-Wide Gene Function Prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(3):832–847, 2010. 5

[19] Cinzia Volonté, Nadia D'Ambrosi, and Susanna Amadio. Protein Cooperation: from Neurons to Networks. *Progress in Neurobiology*, 86(2):61–71, 2008. 6

[20] Tamás Nepusz, Haiyuan Yu, and Alberto Paccanaro. Detecting Overlapping Protein Complexes in Protein-Protein Interaction Networks. *Nature Methods*, 9(5):471–472, 2012. 6

[21] Stanley Fields and Ok-kyu Song. A Novel Genetic System to Detect Protein-Protein Interactions. *Nature*, 340(6230):245–246, 1989. 6

[22] Karthik Raman. Construction and Analysis of Protein-Protein Interaction Networks. *Automated Experimentation*, 2:1–11, 2010. 6

[23] Barbara Kaboord and Maria Perr. Isolation of Proteins and Protein Complexes by Immunoprecipitation. In *2D PAGE: Sample Preparation and Fractionation*, pages 349–364. 2008. 7

[24] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-Based Prediction of Protein Function. *Molecular Systems Biology*, 3(1):88, 2007. 7

[25] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. 2006. 7

[26] Grigorios Tsoumakas and Ioannis Katakis. Multi-Label Classification: An Overview. *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*, 3(3):64–74, 2008. 8

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, volume 1. 2016. 8, 10

[28] David E. Rumelhart, Geoffrey E. Hinton, and R. J. Williams. Learning Representations by Back-Propagating Errors. *Nature*, 323(6088):533–536, 1986. 9

[29] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, pages 1–13, 2015. 10

[30] Dongpo Xu, Shengdong Zhang, Huisheng Zhang, and Danilo P. Mandic. Convergence of the RMSProp Deep Learning Method with Penalty for Nonconvex Optimization. *Neural Networks*, 139:17–23, 2021. 10

[31] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad Stepsizes: Sharp Convergence over Nonconvex Landscapes. *Journal of Machine Learning Research*, 21(219):1–30, 2020. 10

[32] Waseem Rawat and Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9):2352–2449, 2017. 10

[33] Long Wen, Xinyu Li, and Liang Gao. A Transfer Convolutional Neural Network for Fault Diagnosis Based on ResNet-50. *Neural Computing and Applications*, 32(10):6111–6124, 2020. 11

[34] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1D Convolutional Neural Networks and Applications: A Survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021. 11

[35] 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, 2016. 11

[36] Farsana Salim, Faisal Saeed, Shadi Basurra, Sultan Noman Qasem, and Tawfik Al-Hadhrami. DenseNet-201 and Xception Pre-Trained Deep Learning Models for Fruit Recognition. *Electronics*, 12(14):3132, 2023. 11

[37] Debjyoti Sinha and Mohamed El-Sharkawy. Thin MobileNet: An Enhanced MobileNet Architecture. In *IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 280–285, 2019. 11

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 11

[39] Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, and Kurt Keutzer. ImageNet Training in Minutes. In *International Conference on Parallel Processing (ICPP)*, pages 1–10, 2018. 12

[40] Bin Li and Dimas Lima. Facial Expression Recognition via ResNet-50. *International Journal of Cognitive Computing in Engineering*, 2:57–64, 2021. 12

[41] Mingxing Tan and Quoc Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019. 12

[42] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. 12

[43] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv*, pages 1–14, 2016. 12

[44] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021. 12

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30:1–11, 2017. 12

[46] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020. 13

[47] Longlong Jing and Yingli Tian. Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058, 2020. 12

[48] Frederick Sanger. The Chemistry of Insulin: Nobel Lecture, December 11, 1958. *Nobel Lectures, Chemistry 1942–1962*, 1958. 28

[49] Max F. Perutz, Michael G. Rossmann, Andrew F. Cullis, Hugh Muirhead, George Will, and Arthur C. T. North. Structure of Haemoglobin: A Three-Dimensional Fourier Synthesis at 5.5-Å Resolution, Obtained by X-Ray Analysis. *Nature*, 185:416–422, 1960. 28

[50] John B. Fenn, Matthias Mann, Chin Kai Meng, Shek Fu Wong, and Craig M. Whitehouse. Electrospray Ionization for Mass Spectrometry of Large Biomolecules. *Science*, 246(4926):64–71, 1989. 28

[51] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. 28

[52] Damian Szklarczyk, Annika L. Gable, Katerina C. Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T. Doncheva, Marc Legeay, Tao Fang, Peer Bork, Lars J. Jensen, and Christian von Mering. The STRING Database in 2021: Customizable Protein-Protein Networks, and Functional Characterization of User-Uploaded Gene/Measurement Sets. *Nucleic Acids Research*, 49(D1):D605–D612, 2021. 28, 32

[53] Zhuoyang Chen and Qiong Luo. DualNetGO: A Dual Network Model for Protein Function Prediction via Effective Feature Selection. *Bioinformatics*, 40(7):btae437, 2024. 29

[54] Kunjie Fan, Yuanfang Guan, and Yan Zhang. Graph2GO: A Multi-Modal Attributed Network Embedding Method for Inferring Protein Functions. *GigaScience*, 9(8):giaa081, 2020. 29, 30

[55] Zhourun Wu, Mingyue Guo, Xiaopeng Jin, Junjie Chen, and Bin Liu. CFAGO: Cross-Fusion of Network and Attributes Based on Attention Mechanism for Protein Function Prediction. *Bioinformatics*, 39(3):btad123, 2023. 29

[56] Jiajie Peng, Hansheng Xue, Zhongyu Wei, Idil Tuncali, Jianye Hao, and Xuequn Shang. Integrating Multi-Network Topology for Gene Function Prediction Using Deep Neural Networks. *Briefings in Bioinformatics*, 22(2):2096–2105, 2020. 29

[57] Yangyue Fang, Yinzuo Zhou, and Jianping Huang. SE3Graph-PPI: Multiscale Protein-Protein Interaction Prediction Model based on Graph Neural Networks. In *Chinese Control Conference (CCC)*, pages 1024–1029, 2024. 29

[58] Vladimir Gligorijevic, Meet Barot, and Richard Bonneau. deepNF: Deep Network Fusion for Protein Function Prediction. *Bioinformatics*, 34(22):3873–3881, 2018. 29

[59] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 855–864, 2016. 30

[60] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning Distributed Representations of Graphs. *arXiv*, pages 1–8, 2017. 30

[61] Baohui Lin, Xiaoling Luo, Yumeng Liu, and Xiaopeng Jin. A Comprehensive Review and Comparison of Existing Computational Methods for Protein Function Prediction. *Briefings in Bioinformatics*, 25(4):bbae289, 2024. 30

[62] Chengshuai Zhao, Shuai Liu, Feng Huang, Shichao Liu, and Wen Zhang. CSGNN: Contrastive Self-Supervised Graph Neural Network for Molecular Interaction Prediction. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3756–3763, 2021. 30

[63] Mingzhi Yuan, Ao Shen, Kexue Fu, Jiaming Guan, Yingfan Ma, Qin Qiao, and Manning Wang. ProteinMAE: Masked Autoencoder for Protein Surface Self-Supervised Learning. *Bioinformatics*, 39(12):btad724, 2023. 30

[64] Xuehua Bi, Weiyang Liang, Qichang Zhao, and Jianxin Wang. SSLpheno: Self-Supervised Learning for Gene-Phenotype Association Prediction. *Bioinformatics*, 39(11):btad662, 2023. 30

[65] Yuan Zhang, Mingyuan Dong, Junsheng Deng, Jiafeng Wu, Qiuye Zhao, Xieping Gao, and Dapeng Xiong. Graph Masked Self-Distillation Learning for Prediction of Mutation Impact on Protein-Protein Interactions. *Communications Biology*, 7(1):1400, 2024. 30

[66] Michael Doron, Théo Moutakanni, Zitong S. Chen, Nikita Moshkov, Mathilde Caron, Hugo Touvron, Piotr Bojanowsi, Wolfgang M. Pernice, and Juan C. Caicedo. Unbiased Single-Cell Morphology with Self-Supervised Vision Transformers. *bioRxiv*, pages 1–33, 2023. 31

[67] Xueying Xie, Lin Gui, Baixue Qiao, Guohua Wang, Shan Huang, Yuming Zhao, and Shanwen Sun. Deep Learning in Template-Free De Novo Biosynthetic Pathway Design of Natural Products. *Briefings in Bioinformatics*, 25(6):bbae495, 2024. 31

[68] Lingyan Zheng, Shuiyang Shi, Mingkun Lu, Pan Fang, Ziqi Pan, Hongning Zhang, Zhimeng Zhou, Hanyu Zhang, Minjie Mou, Shijie Huang, Lin Tao, Weiqi Xia, Honglin Li, Zhenyu Zeng, Shun Zhang, Yuzong Chen, Zhaorong Li, and Feng Zhu. AnnoPRO: A Strategy for Protein Function Annotation Based on Multi-Scale Protein Representation and a Hybrid Deep Learning of Dual-Path Encoding. *Genome Biology*, 25(1):41, 2024. 31

[69] Xiaoyan Jiang, Zuojin Hu, Shuihua Wang, and Yudong Zhang. Deep Learning for Medical Image-Based Cancer Diagnosis. *Cancers*, 15(14):3608, 2023. 31

[70] Damian Szklarczyk, Katerina Nastou, Mikaela Koutrouli, Rebecca Kirsch, Farrokh Mehryary, Radja Hachilif, Dewei Hu, Matteo E. Peluso, Qingyao Huang, Tao Fang, Nadezhda T. Doncheva, Sampo Pyysalo, Peer Bork, Lars J. Jensen, and Christian von Mering. The STRING Database in 2025: Protein Networks with Directionality of Regulation. *Nucleic Acids Research*, 53(D1):D730–D737, 2025. 32

Apêndice A

Principais Trabalhos Correlatos

Neste apêndice, iremos discutir os principais trabalhos relacionados à anotação e predição de funções proteicas, além de trabalhos que utilizem técnicas que possam auxiliar e se relacionar com a metodologia proposta em nossa pesquisa.

A proteômica, o estudo abrangente das proteínas, incluindo suas estruturas, funções e interações, tem moldado nossa compreensão da biologia celular e molecular. Essa evolução é refletida em mais de dez Prêmios Nobel relacionados ao campo. Frederick Sanger, em 1958, deu o primeiro passo ao sequenciar a insulina, revelando a ordem específica dos aminoácidos em uma proteína [48]. Max Perutz e John Kendrew, em 1962, determinaram as estruturas tridimensionais da hemoglobina e da mioglobina por cristalografia de raios X, marcando o início da compreensão da relação entre estrutura e função proteica [49]. O avanço continuou com Aaron Ciechanover, Avram Hershko e Irwin Rose, que, em 2004, elucidaram o sistema de ubiquitinação, essencial para a regulação proteica celular.

Técnicas revolucionárias, como a espectrometria de massa — originalmente desenvolvida no início do século XX por cientistas como J.J. Thomson e Francis Aston para analisar a composição de íons — passaram por avanços significativos ao longo das décadas. Um desses avanços foi a introdução da ionização por eletrospray (ESI), desenvolvida por John B. Fenn, e a ionização por dessorção a laser suave (SLD), por Koichi Tanaka. Essas contribuições foram reconhecidas com o Prêmio Nobel de Química em 2002, por possibilitarem a análise eficaz de macromoléculas como proteínas, consolidando a espectrometria de massa como uma técnica essencial para a proteômica em larga escala [50]. Mais recentemente, em 2024, Demis Hassabis, John Jumper e David Baker foram reconhecidos pelo impacto do AlphaFold na previsão de estruturas proteicas e no *design* de biomoléculas, conectando diretamente a estrutura ao estudo das funções biológicas [51].

O estudo das redes de interações proteína-proteína (PPI) representa um avanço crucial na predição funcional, pois permite inferir informações funcionais a partir da conectividade entre elas. Essas redes oferecem um contexto funcional detalhado ao descrever as interações entre proteínas nos sistemas biológicos. Bases de dados como a STRING [52] desempenham um papel fundamental nesse campo, integrando múltiplas fontes de evidências, como coexpressão, fusões genéticas e dados experimentais, para atribuir pontuações de confiança às interações proteicas. Essa abordagem possibilita a análise de padrões relacionais e a exploração de propriedades topológicas de grafos de interações, sendo especialmente útil para tarefas de predição funcional.

A.1 Predição Funcional Baseada em Redes PPI e Aprendizado de Máquina

As abordagens computacionais para anotação funcional de proteínas podem ser classificadas em unimodais, que utilizam apenas uma única fonte de informação, ou multimodais, que integram múltiplas modalidades para maior robustez [12]. A incorporação de redes PPI tem sido um fator-chave para melhorar a predição funcional, complementando abordagens baseadas em sequências e estruturas.

DeepGraphGO [12] é um modelo baseado em *Graph Neural Networks* (GNNs) que utiliza exclusivamente redes PPI para a predição funcional de proteínas. Ele constrói uma representação baseada na conectividade entre proteínas na rede, usando *embeddings* de grafos como entrada para a rede neural. A principal vantagem desse método é sua capacidade de capturar relações contextuais complexas entre proteínas em múltiplas espécies, permitindo um aprendizado robusto das interações biológicas.

Já o *DualNetGO* [53] emprega uma abordagem multimodal, combinando *embeddings* derivados de redes PPI com características adicionais das proteínas, como localização subcelular e domínios proteicos. O modelo utiliza um seletor de características baseado em aprendizado profundo, permitindo que diferentes aspectos da proteína sejam analisados separadamente antes de serem combinados para a predição final. Essa abordagem melhora a interpretabilidade e reduz a interferência de informações irrelevantes, otimizando o desempenho do modelo.

O *Graph2GO* [54] utiliza *embeddings* de grafos para integrar múltiplas fontes de informação. Diferentemente dos modelos tradicionais que tratam redes PPI como simples estruturas relacionais, o *Graph2GO* emprega *Graph Attention Networks* (GATs) para ponderar a relevância de diferentes interações e calcular *embeddings* atribuídos. Assim, ele pode distinguir interações biológicas mais relevantes e reduzir o impacto de ruídos presentes nos dados experimentais.

O *CFAGO* [55] incorpora mecanismos de atenção baseados em *transformers* para modelar redes PPI. Ao contrário das GNNs tradicionais, que utilizam convoluções sobre grafos, este modelo adota um mecanismo de atenção cruzada, que permite a fusão de atributos proteicos e conectividade da rede. Ele aprende relações dinâmicas entre proteínas e suas conexões, destacando interações relevantes para a inferência funcional.

As redes convolucionais em grafos (GCNNs) têm se destacado como uma abordagem promissora para extrair representações de redes PPI [56]. Um aspecto essencial ao aplicar GCNNs para anotação funcional é a necessidade de representar cada nó (proteína) com um vetor de características, enquanto as arestas representam as interações extraídas via STRING. Esse vetor pode ser gerado a partir de *embeddings* de sequências utilizando modelos baseados em *transformers*, como o ProtBERT (para capturar informações de sequência), ou a partir de informações biológicas extraídas de bancos de dados experimentais. Outro exemplo é o *SE3Graph-PPI* [57] que emprega GCNNs com invariança espacial para prever interações proteína-proteína e inferir funções biológicas. Ele se diferencia ao modelar a conectividade das redes PPI em diferentes escalas estruturais, melhorando a capacidade do modelo de lidar com interações de longo alcance.

Por fim, *DeepNF* [58] é um modelo baseado em autoencoders multimodais para predição funcional utilizando exclusivamente redes PPI. Ele transforma múltiplas redes biológicas em um espaço latente unificado, capturando padrões de interação em diferentes bases de dados experimentais. O modelo usa autoencoders baseados em *deep learning* para aprender repre-

sentações compactas de redes PPI e aplica redes neurais profundas para integrar informações heterogêneas. A vantagem do *DeepNF* está na capacidade de combinar múltiplas redes PPI e inferir funções proteicas a partir de padrões estruturais.

Métodos tradicionais, como *Node2Vec* [59], utilizam caminhadas aleatórias enviesadas para gerar *embeddings* de nós, permitindo que algoritmos supervisionados usem essas representações como entrada. O *Graph2Vec* [60] expande essa abordagem ao representar subgrafos inteiros, extraindo representações vetoriais que capturam módulos funcionais da rede PPI.

A.2 Aprendizado Autossupervisionado em Redes PPI

A predição funcional baseada exclusivamente em redes PPI enfrenta desafios devido ao ruído presente nos dados experimentais [61]. Estratégias de aprendizado autossupervisionado surgem como soluções promissoras para extrair representações latentes dessas redes sem a necessidade de rótulos explícitos.

O aprendizado autossupervisionado permite que modelos aprendam representações ricas das redes PPI explorando características inerentes à estrutura do grafo. Essas abordagens podem ser classificadas em métodos baseados em reconstrução de nós e arestas, aprendizado contrastivo e aprendizado mascarado.

Contrastive Self-Supervised Learning (CSSL) tem sido aplicado para otimizar a extração de características a partir de redes PPI, onde o modelo aprende a maximizar a similaridade entre proteínas funcionalmente próximas e minimizar a similaridade entre proteínas sem relação funcional. Métodos como o CSGNN (*Contrastive Self-Supervised Graph Neural Network*) [62] utilizam essa abordagem para melhorar a predição de interações moleculares em redes biológicas, incluindo redes de interações proteína-proteína. O CSGNN emprega contrastes entre pares de proteínas para aprender representações que refletem relações funcionais.

Outro método contrastivo relevante é o *ProteinMAE* (*Protein Masked Autoencoder*) [63], que aplica um autoencoder mascarado para aprender representações estruturais de proteínas a partir de redes PPI. Esse modelo esconde partes do grafo (como nós ou arestas) e treina a rede para reconstruí-las, garantindo que as representações finais capturem padrões relacionais e estruturais essenciais.

Abordagens de aprendizado autossupervisionado podem ser utilizadas para pré-treinamento de modelos que posteriormente serão refinados para tarefas específicas de anotação funcional. O pré-treinamento pode ser realizado inicialmente por meio da extração de *embeddings* usando um modelo de *Graph Neural Network* (GNN) treinado em uma grande rede PPI sem supervisão explícita, empregando aprendizado contrastivo ou autoencoders mascarados. Após esse estágio inicial, o modelo pode ser refinado com um conjunto de dados anotado, aplicando aprendizado supervisionado para associar as representações aprendidas a funções específicas de proteínas. Finalmente, o modelo treinado pode ser transferido para outras redes PPI, permitindo que as representações aprendidas sejam reutilizadas para predições em contextos biológicos distintos.

Modelos como o *SSLpheno* [64] aplicam aprendizado autossupervisionado para gerar representações ricas de proteínas e prever associações gene-fenótipo com base em redes de interações. Da mesma forma, o *Graph2GO* [54] utiliza um pré-treinamento autossupervisionado para gerar *embeddings* que combinam informações de redes PPI e dados de atributos proteicos. Abordagens híbridas que combinam aprendizado autossupervisionado com técnicas de *masking learning* (aprendizado mascarado) têm demonstrado grande potencial. O método *Graph Masked Self-Distillation Learning* [65] explora essa técnica para inferir impac-

tos de mutações em interações PPI, mostrando que essas abordagens podem ser utilizadas para estudar variantes genéticas e sua influência nas funções das proteínas.

Portanto, a aplicação de aprendizado autossupervisionado a redes PPI abre novas possibilidades para gerar *embeddings* ricos e transferíveis, reduzindo a necessidade de grandes conjuntos de dados anotados e permitindo que modelos sejam generalizáveis a diferentes espécies e contextos biológicos. Além disso, a utilização de técnicas como aprendizado contrastivo, autoencoders mascarados e pré-treinamento transferível tem demonstrado ser uma abordagem eficaz para otimizar modelos de anotação funcional de proteínas.

A.3 Abordagens Visuais para Predição Funcional

Métodos baseados em *transfer learning* com CNNs e *Visual Transformers* têm sido explorados para representar proteínas em imagens e melhorar a predição funcional [66, 67].

O *AnnoPRO* [68] explora representações visuais de redes PPI, como matrizes de adjacência convertidas em mapas de similaridade. Esses mapas permitem que CNNs ou *Transformers* capturem padrões estruturais da conectividade proteica, auxiliando na predição funcional.

Técnicas de *transfer learning* têm sido aplicadas a modelos visuais para pré-treinar redes neurais em grandes bases de dados de proteínas e refiná-las posteriormente para tarefas específicas de anotação funcional [69]. Essa estratégia tem demonstrado sucesso em domínios como a classificação de sublocalização proteica e a predição de interações proteína-proteína.

Modelos como *Vision Transformers (ViTs)* e *Self-Supervised Learning* em imagens de proteínas mostraram grande potencial para extração de informações estruturais, sendo explorados em pesquisas recentes para capturar padrões morfológicos e relacionamentos entre proteínas [66].

A.4 Conclusão

Em síntese, a predição funcional de proteínas evoluiu significativamente com redes de interações proteína-proteína (PPI) e técnicas de *deep learning*. Métodos baseados em grafos e aprendizado autossupervisionado destacam-se por capturar relações complexas e propriedades topológicas das redes PPI, reduzindo a dependência de dados rotulados. Abordagens visuais, por sua vez, exploram novas representações estruturais e relacionais dessas redes. Essas estratégias não apenas oferecem soluções robustas para anotação funcional em sistemas biológicos complexos, mas também visam superar desafios como granularidade e ruídos em dados experimentais, impulsionando modelos mais generalizáveis.

Apêndice B

Bases de Dados

Neste apêndice, apresentamos as bases de dados utilizadas nos experimentos iniciais, além de uma breve análise exploratória dessas bases.

B.1 STRING

A base de dados STRING (*Search Tool for the Retrieval of Interacting Genes/Proteins*) é uma ferramenta amplamente utilizada para armazenar e organizar informações sobre interações proteína-proteína (PPI). Em sua versão 11.5 [52], a STRING integra dados de diversas fontes, incluindo experimentos laboratoriais, análises computacionais, mineração de textos e curadoria manual, com o objetivo de fornecer uma visão abrangente das interações funcionais entre proteínas.

As interações na STRING são representadas como um grafo, em que cada nó corresponde a uma proteína e cada aresta indica uma interação [52, 70]. Essas interações são sustentadas por sete tipos principais de evidências: vizinhança no genoma, fusão gênica, coocorrência em genomas distintos, coexpressão gênica, experimentos laboratoriais, informações de bases de dados curadas manualmente e mineração de textos científicos. Cada categoria de evidência recebe uma pontuação específica, de acordo com sua confiabilidade e tipo de suporte subjacente. Além disso, a STRING fornece uma pontuação combinada que unifica todas as evidências disponíveis, permitindo uma análise integrada e eficiente. No presente trabalho, utilizamos essa pontuação combinada como critério principal para priorizar as interações mais relevantes.

A STRING é amplamente empregada para relacionar proteínas a funções biológicas [52, 70]. A Figura B.1 ilustra um exemplo de rede PPI gerada na STRING para a proteína INSR (*Insulin Receptor Subunit Alpha*), um receptor tirosina quinase essencial na mediação dos efeitos pleiotrópicos da insulina. No grafo, os nós representam proteínas que interagem com INSR, e a espessura das arestas indica o nível de confiança das evidências de interação. As conexões mais espessas refletem interações mais bem documentadas e suportadas por múltiplos tipos de evidências. Proteínas como IRS1, IRS2, IGF1R e SHC1 aparecem conectadas ao INSR, refletindo seu papel central na sinalização da insulina e seus efeitos metabólicos e mitogênicos.

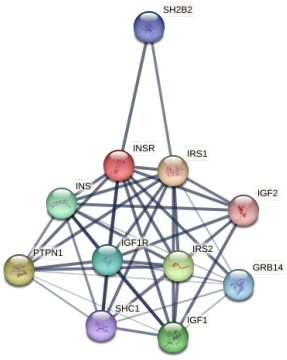

Figura B.1: Rede de interações proteína-proteína (PPI) gerada na ferramenta STRING para a proteína INSR em humanos (*Homo sapiens*). Os nós representam proteínas interagentes e a espessura das arestas indica o nível de confiança das evidências de interação. As conexões mais espessas refletem interações com maior suporte experimental e computacional.

B.2 CAFA5

Para os experimentos iniciais, empregamos uma base adaptada do CAFA5 [10], cujo quinto desafio foi realizado em 2023. Os dados de treinamento foram disponibilizados pela organização na plataforma Kaggle[1], mas o conjunto de teste oficial permanece oculto. Dessa forma, dividimos manualmente o conjunto de treinamento em 80% (treinamento), 10% (validação) e 10% (teste).

No que diz respeito aos termos, selecionamos os mais frequentes em cada ontologia. Inicialmente, definimos 500 termos como limite, mas, em Componente Celular (CC) e Função Molecular (MF), ocorreu um empate próximo a esse valor. Para evitar arbitrariedades, optamos pelo maior número abaixo do empate, resultando em 498 termos para CC e 499 termos para MF. A Tabela B.1 apresenta a quantidade de proteínas em cada subconjunto, bem como o número de termos.

Tabela B.1: Número de proteínas e termos na base CAFA5 adaptada.

Ontologia	**Treinamento**	**Validação**	**Teste**	**Termos**
Componente Celular (CC)	74.328	9.292	9.292	498
Função Molecular (MF)	62.909	7.864	7.864	499
Processo Biológico (BP)	73.768	9.221	9.221	500

B.2.1 Análise Exploratória das Interações

Na análise exploratória dos dados de interação, avaliamos a frequência de interações por proteína em cada domínio da *Gene Ontology* (*Biological Process*, *Molecular Function* e *Cellular Component*). A Tabela B.2 apresenta estatísticas descritivas, incluindo média, mediana, desvio-padrão, quartis, limites de outliers e quantidade total de proteínas em cada ontologia.

[1] `https://kaggle.com/competitions/cafa-5-protein-function-prediction`

Tabela B.2: Estatísticas descritivas do número de interações por proteína em cada domínio da *Gene Ontology*.

Estatística	**BP**	**CC**	**MF**
Média	41,06	44,44	45,86
Mediana	11,00	13,00	15,00
Desvio Padrão (*DP*)	82,60	86,20	88,40
Valor Mínimo	0,00	0,00	0,00
Valor Máximo	2794,00	2794,00	2794,00
1º Quartil (*Q1*)	0,00	0,00	0,00
3º Quartil (*Q3*)	49,00	53,00	55,00
Limite *Outlier* (sup.)	122,50	132,50	137,50
Número de *outliers*	7503,00	7455,00	6090,00

Observa-se que as distribuições são bastante assimétricas: as médias (41,06 em BP, 44,44 em CC e 45,86 em MF) superam significativamente as medianas (11, 13 e 15, respectivamente). Essa discrepância indica a existência de proteínas com grau de conectividade muito elevado, o que eleva a média de forma desproporcional. Os limites superiores para a definição de *outliers* (122,50 em BP, 132,50 em CC e 137,50 em MF) foram estabelecidos adotando-se o critério clássico de *boxplot*, ou seja, valores acima de $\mathrm{Q3} + 1{,}5 \times \mathrm{IQR}$. Dessa forma, proteínas cujo número de interações ultrapassa esses limiares são consideradas fora da faixa esperada, refletindo a influência de casos com muitas conexões.

Ademais, a quantidade expressiva de *outliers* — em torno de 10% das proteínas (7.503 em BP, 7.455 em CC e 6.090 em MF) — evidencia alta variabilidade e sugere a necessidade de análises mais específicas. Esses casos extremos podem afetar o desempenho dos algoritmos de aprendizado, requerendo estratégias apropriadas, como técnicas de normalização, amostragem ou modelos mais robustos a distribuições altamente desequilibradas.

Adicionalmente, como passo futuro, seria relevante conduzir uma análise exploratória da frequência de cada função (termo) anotada nas bases, buscando identificar possíveis desequilíbrios de classe. Quando há classes com poucos exemplos em comparação às demais, os modelos de aprendizado tendem a priorizar as funções majoritárias, comprometendo o desempenho global. Caso esse desequilíbrio seja detectado, podem ser necessárias estratégias como reamostragem, ponderação de classes ou abordagens específicas de aprendizado para dados desbalanceados, de modo a mitigar o viés e melhorar capacidade de generalização dos classificadores.

Apêndice C

Métricas

Este apêndice apresenta as definições formais das métricas de avaliação ($\mathbf{F}_{\max}$, $\mathbf{S}_{\min}$, **AuPRC**, **IAuPRC**).

C.1 $F_{\max}$

A métrica $F_{\max}$ avalia, ao longo de diferentes limiares τ (de 0,01 a 1,0 em incrementos de 0,01), o ponto em que a média harmônica de *precisão* (pr) e *revocação* (rc) atinge seu valor máximo:

$$F_{\max} = \max_{\tau} \left\{ \frac{2 \cdot pr(\tau) \cdot rc(\tau)}{pr(\tau) + rc(\tau)} \right\}.$$

onde:

$$pr(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{\left| P_i(\tau) \cap T_i \right|}{\left| P_i(\tau) \right|}, \quad rc(\tau) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| P_i(\tau) \cap T_i \right|}{\left| T_i \right|}.$$

Aqui, $P_i(\tau)$ é o conjunto de termos cuja pontuação é $\geq \tau$ para a proteína i, T_i é o conjunto de termos verdadeiros para a proteína i, $m(\tau)$ é o número de proteínas que têm ao menos um termo predito com pontuação $\geq \tau$, e n é o número total de proteínas no conjunto.

C.2 $S_{\min}$

A métrica $S_{\min}$ mede a discrepância semântica entre predições e anotações verdadeiras, combinando o custo de falsos negativos (RU) e falsos positivos (MI). Ela é definida como:

$$S(\tau) = \sqrt{\frac{\mathrm{RU}(\tau)^2 + \mathrm{MI}(\tau)^2}{2}},$$

onde:

- $\mathrm{RU}(\tau)$ (*Remaining Uncertainty*) é o custo semântico médio dos termos verdadeiros não preditos.
- $\mathrm{MI}(\tau)$ (*Misinformation*) é o custo semântico médio dos termos preditos incorretamente.

O valor de $S_{\min}$ é obtido pela minimização de $S(\tau)$ sobre todos os limiares τ:

$$S_{\min} = \min_{\tau} S(\tau).$$

C.3 AuPRC

A *Area Under the Precision-Recall Curve* (AuPRC) reflete o desempenho global do classificador conforme se variam os limiares de decisão. A curva é construída traçando-se *precisão* (ordenada) contra *revocação* (abscissa) para diferentes valores de τ durante o calculo do $F_{\max}$. A AuPRC pode ser calculada por integração numérica:

$$\text{AuPRC} = \sum_{k=1}^{K-1} \big(r_{k+1} - r_k\big) \cdot p_{k+1},$$

onde r_k e p_k são, respectivamente, a revocação e a precisão no ponto k. Valores mais altos de AuPRC indicam melhor desempenho no *trade-off* entre precisão e revocação.

C.4 IAuPRC

A *Interpolated AuPRC* (IAuPRC) é uma variação da AuPRC que realiza interpolação nos pontos de revocação, assumindo que a melhor precisão obtida em qualquer valor de revocação maior ou igual ao ponto atual deve ser usada nesse ponto. Dessa forma, métodos que não tenham predições em regiões de baixa revocação, mas possuam boa precisão em faixas intermediárias ou altas, não são penalizados de maneira desproporcional.

Formalmente, se definirmos um conjunto de pontos $\{(r_k, p_k)\}$ para $k = 1, 2, \ldots, K$, com $r_1 < r_2 < \cdots < r_K$, podemos interpolar:

$$p_{\text{interp}}(r) = \max_{r_k \geq r} p_k,$$

e então integrar a curva $\big(r,\, p_{\text{interp}}(r)\big)$ para obter a IAuPRC.

Apêndice D

Experimentos Iniciais

Este apêndice descreve os experimentos iniciais realizados para a classificação multirrótulo de proteínas nas três ontologias principais da *Gene Ontology* (GO), a saber: *Biological Process* (BP), *Molecular Function* (MF) e *Cellular Component* (CC). Cada ontologia dispõe de um classificador independente, e as funções filhas previstas para uma proteína são propagadas aos ancestrais na GO, garantindo coerência hierárquica. Nesta etapa, investigamos duas estratégias de representação dos dados de interações proteína-proteína (PPI) fornecidos pela base STRING (como mencionado no Apêndice B): uma abordagem tabular, na qual cada proteína é convertida em um vetor numérico, e uma abordagem por imagem, em que esse vetor é transformado em uma matriz (imagem sintética) para processamento por redes convolucionais.

Todas as avaliações (tanto na validação quanto no teste) fazem uso das métricas descritas no Apêndice C. Além disso, foi implementado um método que gera a ontologia agregando hierarquicamente as anotações funcionais para os nós ancestrais na ontologia GO, a fim de garantir a consistência e eficiência das anotações.

D.1 Construção do Vetor de Pontuação de Evidências

Para cada proteína-alvo, avaliamos a relação (interação) dela com as D proteínas de treinamento, onde D corresponde ao número de proteínas do conjunto de treino que possuem anotações funcionais conhecidas. Cada relação é armazenada em um vetor $\mathbf{v} \in \mathbb{R}^D$, em que a i-ésima posição ($\mathbf{v}_i$) indica a confiança de interação entre a proteína-alvo e a proteína i do treinamento. Esses valores variam de 0 a 1 e são extraídos da base STRING, que consolida múltiplas evidências (coexpressão gênica, experimentos, dados curados, etc.).

No caso específico em que a proteína-alvo também faz parte do conjunto de treinamento, ocorre apenas uma auto-interação: nessa posição do vetor, atribuímos o valor 1. Por outro lado, se não houver nenhuma evidência de interação entre a proteína-alvo e uma proteína i do treino, definimos $\mathbf{v}_i = 0$. Assim, cada proteína é descrita por um vetor de dimensão D que sintetiza as pontuações de interação com todas as proteínas de treinamento.

D.2 Média Ponderada das Anotações

Para estabelecer uma linha de base (*baseline*), consideramos uma média ponderada simples das anotações dos vizinhos. Nesse procedimento, normalizamos cada $\mathbf{v}_i$ por sua soma total

de modo a transformá-lo em um peso w_i, de forma que a soma de todos os pesos para uma proteína seja igual a 1. Em seguida, esses pesos são aplicados sobre as proteínas vizinhas (isto é, as proteínas que tem uma pontuação de evidência de interação alta no vetor de pontuações) que possuam aquele termo GO anotado, resultando em uma probabilidade de atribuição para cada termo. Formalmente,

$$P(\text{termo} \mid \text{alvo}) \;=\; \sum_{i=1}^{D} \Big(w_i \times \mathbf{1}_{\{\text{termo em } i\}} \Big).$$

Por não envolver aprendizado de parâmetros, essa abordagem não sofre *overfitting*, mas tende a apresentar resultados inferiores sempre que existirem padrões mais complexos a serem capturados nas interações.

D.3 Abordagem Tabular

Na abordagem tabular, a proteína é descrita como um vetor de dimensão fixa, o que possibilita a aplicação de modelos clássicos de aprendizado de máquina ou redes neurais densas. Três configurações distintas de pré-processamento foram avaliadas:

- **Vetor Inteiro (*raw*)**: mantém toda a informação de interações no vetor $\mathbf{v}$ de dimensão D. Embora preserve nuances importantes, sofre com a "maldição da dimensionalidade" e pode demandar redes profundas ou regularização intensiva para evitar *overfitting*.
- ***Max Pooling***: agrupa janelas de valores e retém apenas o máximo em cada janela, reduzindo o vetor para $\mathbf{v}^{\max}$. Essa compactação é agressiva e ignora valores intermediários, mas gera uma dimensionalidade menor, o que facilita o treinamento.
- ***Average Pooling***: análogo ao *max pooling*, porém retendo a média em cada janela, o que tende a preservar mais informações globais, suavizar picos e, em alguns casos, ajudar na estabilidade do treinamento.

Modelo e Treinamento

Optamos por uma arquitetura de MLP (*Multi-Layer Perceptron*) composta de duas camadas ocultas (128 e 64 neurônios, ambas com ativação ReLU), seguidas de uma camada de saída cujo número de neurônios corresponde à quantidade de termos GO em cada ontologia. Utilizamos a função de perda entropia binária cruzada para lidar com a tarefa multirrótulo, o otimizador Adam e *early stopping* baseado na perda de validação. O mesmo *pipeline* de treinamento e testes foi aplicado, garantindo a consistência entre as diferentes configurações de *pooling*.

Resultados e Discussões

A Tabela D.1 apresenta os resultados de F_{max}, S_{min}, AuPRC e IAuPRC, comparando a baseline e as três configurações (vetor *raw*, *max pooling* e *average pooling*). Observando especificamente a métrica F_{max}, percebemos:

- **Biological Process (BP):** a linha de base (*baseline*) obteve F_{max} de 0,4368, superando as três variações com MLP (cujos valores ficaram entre 0,3558 e 0,3969). Esse resultado indica que, nessa ontologia, o uso direto das ponderações dos vizinhos (baseline) funcionou melhor do que as redes densas — possivelmente porque as interações em BP sejam mais complexas ou distribuídas, dificultando o aprendizado de padrões sem um pré-processamento mais refinado.

- **Cellular Component (CC):** houve uma melhora significativa ao aplicar MLP com *max pooling*, que chegou a F_{max} de 0,6620, superando a baseline (0,5759) e as demais configurações. A MLP com *avg pooling* também ficou acima da baseline, enquanto o vetor *raw* praticamente se manteve no mesmo patamar do baseline (0,5745).

- **Molecular Function (MF):** todas as variações de MLP (*raw*, *max pool*, *avg pool*) superaram a baseline de 0,5324, com destaque para *max pooling* atingindo 0,6017.

Os resultados sugerem que MLP (*max pool*) é a configuração mais promissora em CC e MF, mas não em BP. Já a abordagem *raw* mostrou-se pouco competitiva em CC e BP, embora tenha um incremento moderado em MF (em relação ao baseline). Isso contradiz a hipótese inicial de que manter toda a dimensionalidade (vetor *raw*) traria os melhores desempenhos, pois, neste conjunto de resultados, o *max pooling* se destacou em duas das três ontologias.

A análise de $\mathbf{S}_{min}$ (custo semântico) evidencia que, em BP, a *max pooling* obteve a menor discrepância (23,4926), seguida por *raw* (23,8144) e *avg pool* (24,7065). Em CC, o menor S_{min} foi do *raw* (11,6893), enquanto *max pool* e *avg pool* ficaram próximos (em torno de 11,7 e 12,0, respectivamente). Já em MF, o *max pooling* (8,9388) obteve a melhor coerência hierárquica, contra 9,3573 (*avg*) e 10,4853 (*raw*). Assim, não há um padrão único entre as ontologias: *max pooling* traz melhor F_{max} em CC e MF e, em geral, bons valores de S_{min}, mas o vetor *raw* ainda aparece como opção competitiva para CC no quesito custo semântico, embora com F_{max} similar ou ligeiramente menor que a baseline.

Por fim, os valores de **AuPRC** e **IAuPRC** confirmam tendência semelhante: no caso de CC e MF, as configurações com *pooling* mostram-se mais robustas a cenários desbalanceados, enquanto em BP há queda acentuada — sugerindo que a representação simplificada (*pooling*) pode descartar interações intermediárias importantes nesta ontologia específica.

Os gráficos de perda indicam que, na abordagem utilizando o vetor inteiro de pontuação, o erro de treinamento decai rapidamente; contudo, essa abordagem apresenta um elevado *overfitting*, evidenciado pela perda de validação significativamente maior que a de treinamento. Em contraste, a estratégia com *average pooling* demonstra boa capacidade de generalização, enquanto a abordagem com *max pooling* alcança os melhores resultados, evidenciando um equilíbrio adequado entre ajuste e generalização. Esses comportamentos estão ilustrados nos gráficos comparativos das três abordagens para *Molecular Function* na Figura D.1.

D.4 Abordagem por Imagem

Nesta abordagem, o vetor de evidências de interação $\mathbf{v} \in \mathbb{R}^D$ é reduzido (via *max* ou *average pooling*) para uma dimensão d e então transformado em uma matriz $A \in \mathbb{R}^{d \times d}$, cujos elementos são as diferenças absolutas entre componentes do vetor, isto é, $A_{i,j} = |v_i - v_j|$, gerando imagens como as da Figura D.2. Se as componentes de $\mathbf{v}$ já estiverem em [0,1], a

Tabela D.1: Resultados para a abordagem Tabular, comparando vetor inteiro (*raw*), *max pooling*, *average pooling* e a baseline de média ponderada. Os valores em negrito apresentam a melhor pontuação e os sublinhados a segunda posição.

Método	**Ontologia**	**Fmax**	**Smin**	**AuPRC**	**IAuPRC**
Baseline	BP	**0,4368**	**21,9111**	**0,3977**	0,3651
MLP (raw)	BP	0,3969	23,8144	0,3935	0,3921
MLP (max pool)	BP	0,3944	23,4926	0,3773	**0,3951**
MLP (avg pool)	BP	0,3558	24,7065	0,2775	0,3401
Baseline	CC	0,5759	**10,9067**	0,5550	0,5029
MLP (raw)	CC	0,5745	11,6893	0,5812	0,6204
MLP (max pool)	CC	**0,6620**	11,7706	**0,6700**	**0,7056**
MLP (avg pool)	CC	0,5997	12,0358	0,5903	0,6352
Baseline	MF	0,5324	**8,4366**	0,4979	0,4656
MLP (raw)	MF	0,5631	10,4853	0,4789	0,5126
MLP (max pool)	MF	**0,6017**	8,9388	**0,5106**	0,5018
MLP (avg pool)	MF	0,5834	9,3573	0,4230	**0,5424**

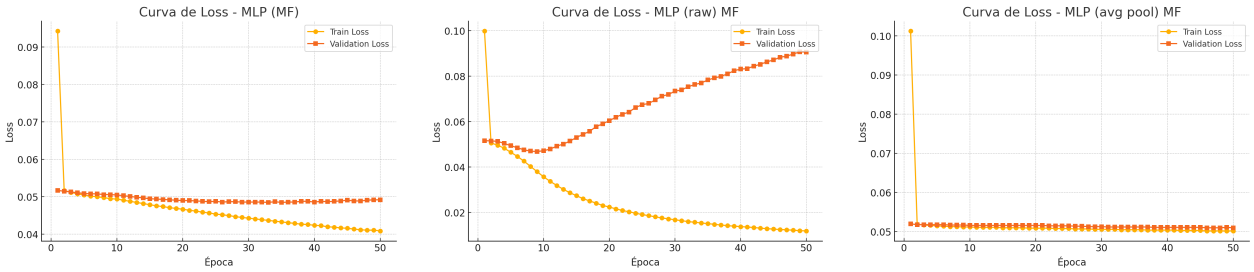

Figura D.1: Gráficos de perda de treinamento e validação comparando as abordagens de MLP com *max pooling*, *avarage pooling* e *raw* (isto é, vetor de pontuações inteiro).

matriz A mantém valores nesse intervalo. Para tornar a entrada compatível com redes pré-treinadas em imagens *RGB*, replicamos esse canal em três canais, resultando em um *tensor* de dimensões $(3, d, d)$.

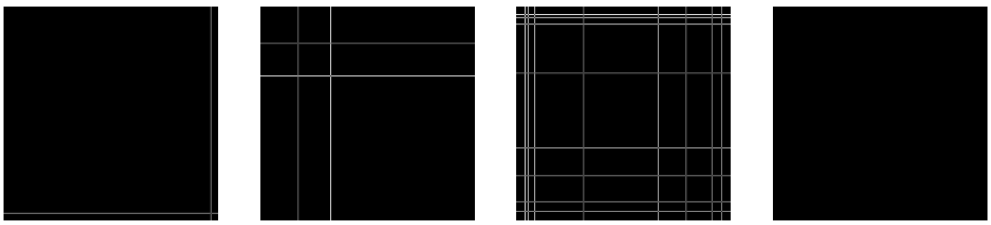

Figura D.2: Imagens obtidas a partir da diferença absoluta entre componentes do vetor de pontuação de interações (proteína alvo com proteínas do treinamento) e seu transposto reduzidos por *max pooling*.

D.4.1 Modelos Avaliados e Configuração de Treinamento

A partir das imagens, empregamos duas arquiteturas conhecidas de *transfer learning*: ResNet50 e EfficientNetB0. O processo de *fine-tuning* de ambas é semelhante. Inicialmente, congelamos os parâmetros de todas as camadas convolucionais, preservando os pesos pré-treinados na base ImageNet. Em seguida, descongelamos as dez últimas camadas convolucionais, permitindo sua adaptação às imagens sintéticas de interações PPI. A camada de saída original é substituída por uma camada densa que produz a quantidade de saídas multirrótulo correspondente ao número de termos GO naquela ontologia.

O treinamento segue por 50 épocas e utiliza o otimizador AdamW com taxa de aprendizado inicial de 10^{-4}. A função de perda é a entropia binária cruzada, apropriada para problemas de classificação multirrótulo. Adotamos também *early stopping*, interrompendo o treinamento quando a perda de validação não melhora por um número pré-estabelecido de épocas.

Resultados e Discussões

A Tabela D.2 exibe F_{max}, S_{min}, AuPRC e IAuPRC para ResNet50 e EfficientNetB0, em comparação com a baseline de média ponderada. Observamos que:

- Em *Biological Process* (BP), tanto ResNet50 quanto EfficientNetB0 ficaram abaixo da baseline em F_{max}, sugerindo que a conversão em imagens e posterior *fine-tuning* não capturou os padrões de interação de forma eficaz nessa ontologia específica.
- Em *Cellular Component* (CC) e *Molecular Function* (MF), ambos os modelos superaram a baseline, com destaque para a ResNet50 em CC (F_{max} de 0,6360) e MF (0,5903). No entanto, esses resultados ainda não ultrapassam o melhor desempenho obtido pelas MLPs tabulares (por exemplo, 0,6620 em CC e 0,6017 em MF, ambos com *max pooling*).

O valor de $\mathbf{S}_{min}$ em BP, por exemplo, mostra-se relativamente alto (em torno de 24,7) para as duas arquiteturas de visão, indicando maior discrepância semântica na ontologia GO, o que reforça a dificuldade de capturar a complexidade de BP a partir de imagens sintéticas. Em contrapartida, para CC e MF, os valores de AuPRC e IAuPRC são razoáveis e consistentemente acima da baseline, embora ainda fiquem abaixo dos melhores modelos tabulares.

Esses resultados sugerem que, apesar de as arquiteturas de visão (ResNet50, EfficientNetB0) proporcionarem ganhos em relação à baseline em CC e MF, há espaço para melhorar o *fine-tuning* — por exemplo, descongelando mais camadas ou adotando arquiteturas mais especializadas para imagens artificiais. Ainda assim, é notável a influência que a conversão do vetor em uma matriz de diferenças pode ter na perda de detalhes, principalmente em ontologias mais complexas (caso de BP).

D.5 Conclusões Parciais e Próximos Passos

Esses experimentos iniciais evidenciam que MLPs (principalmente com *max pooling*) e redes convolucionais (ResNet50 e EfficientNetB0) podem superar a *baseline* em métricas importantes nas ontologias CC e MF. Contudo, em BP, a baseline ainda se destaca, indicando

Tabela D.2: Resultados para a abordagem por Imagem, comparando ResNet50 e EfficientNetB0 com a *baseline* de média ponderada. Os valores em negrito apresentam a melhor pontuação e os sublinhados a segunda posição.

Método	**Ontologia**	**Fmax**	**Smin**	**AuPRC**	**IAuPRC**
Baseline	BP	**0,4368**	**21,9111**	**0,3977**	**0,3651**
ResNet50	BP	0,3582	24,7360	0,3538	0,3453
EfficientNetB0	BP	0,3562	24,7878	0,3519	0,3430
Baseline	CC	0,5759	**10,9067**	0,5550	0,5029
ResNet50	CC	**0,6360**	12,6971	**0,6476**	**0,6679**
EfficientNetB0	CC	0,6351	12,7187	0,5753	0,6657
Baseline	MF	0,5324	**8,4366**	0,4979	0,4656
ResNet50	MF	**0,5903**	9,2395	0,5248	**0,5604**
EfficientNetB0	MF	0,5873	9,3204	**0,5583**	0,5562

que a média ponderada simples das anotações dos vizinhos pode ser mais eficaz nesse caso específico. Dessa forma, os resultados possuem nuances importantes:

- **Modelos Tabulares**: o *max pooling* alcançou o melhor desempenho em CC ($F_{max} = 0,6620$) e MF ($F_{max} = 0,6017$), superando tanto a baseline quanto outras configurações de *pooling* ou mesmo o vetor *raw*. Entretanto, em BP, houve queda significativa em relação à baseline.
- **Modelos por Imagem**: em CC e MF, as CNNs ajustadas (ResNet50 e EfficientNetB0) obtiveram F_{max} superiores à baseline, mas ainda inferiores às melhores MLPs tabulares. Em BP, ficaram abaixo até mesmo da baseline.

No que diz respeito às métricas, F_{max} é útil para avaliar o melhor ponto de equilíbrio entre precisão e revocação, enquanto S_{min} revela se as predições se alinham semântica e hierarquicamente aos termos verdadeiros. Já as áreas sob as curvas (AuPRC e IAuPRC) destacam a qualidade global do classificador em cenários desbalanceados de classificação multirrótulo; em geral, as configurações de *pooling* (*max* ou *avg*) para MLP tiveram bom comportamento em CC e MF, embora tenham sofrido perdas em BP.

Para dar continuidade a este trabalho, pretendemos investigar métodos mais seletivos de redução de dimensionalidade (*autoencoders*, seleção de atributos, projeções não lineares) que possam preservar os sinais mais relevantes enquanto mitigam os efeitos da alta dimensionalidade. Planejamos também explorar *Graph Neural Networks* (GCNs e variações), que têm a capacidade de operar diretamente sobre a topologia da rede PPI, dispensando a necessidade de vetores fixos ou imagens sintéticas. Adicionalmente, será analisada a possibilidade de realizar um ajuste fino (*fine-tuning*) mais detalhado das redes de visão ao domínio específico abordado, avaliando arquiteturas mais adequadas às particularidades das imagens geradas, além de investigar técnicas de pré-processamento que realcem os seus atributos mais relevantes. Em suma, os resultados iniciais fornecem pistas valiosas sobre o comportamento dos modelos e indicam caminhos promissores para uma anotação funcional mais robusta e escalável.