



**UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE QUÍMICA**

**GUILHERME ZAINOTTI MIGUEL FAHUR BOTTINO**

**PREVISÃO DA ESTRUTURA SECUNDÁRIA DE PROTEÍNAS POR  
PROCESSAMENTO DE SEQUÊNCIAS EMPREGANDO UM MODELO DE  
LINGUAGEM ACESSÍVEL E ULTRA-PARALELIZADO**

**CAMPINAS**

**2024**

**GUILHERME ZAINOTTI MIGUEL FAHUR BOTTINO**

**PREVISÃO DA ESTRUTURA SECUNDÁRIA DE PROTEÍNAS POR  
PROCESSAMENTO DE SEQUÊNCIAS EMPREGANDO UM MODELO DE  
LINGUAGEM ACESSÍVEL E ULTRA-PARALELIZADO**

**Tese de Doutorado apresentada ao Instituto de Química  
da Universidade Estadual de Campinas como parte dos  
requisitos exigidos para a obtenção do título de Doutor  
em Ciências**

**Orientador: Prof. Dr. Leandro Martínez**

**O arquivo digital corresponde à versão final da Tese defendida pelo aluno Guilherme  
Zainotti Miguel Fahur Bottino e orientada pelo Prof. Dr. Leandro Martínez.**

**CAMPINAS**

**2024**

Ficha catalográfica  
Universidade Estadual de Campinas (UNICAMP)  
Biblioteca do Instituto de Química  
Camila Barleta Fullin - CRB 8462

B659p Bottino, Guilherme Zainotti Miguel Fahur, 1992-  
Previsão da estrutura secundária de proteínas por processamento de sequências empregando um modelo de linguagem acessível e ultra-paralelizado / Guilherme Zainotti Miguel Fahur Bottino. – Campinas, SP : [s.n.], 2024.

Orientador: Leandro Martínez.  
Tese (doutorado) – Universidade Estadual de Campinas (UNICAMP),  
Instituto de Química.

1. Bioinformática. 2. Processamento sequencial (Computação). 3. Proteínas - Estrutura. I. Martínez, Leandro, 1979-. II. Universidade Estadual de Campinas (UNICAMP). Instituto de Química. III. Título.

Informações Complementares

**Título em outro idioma:** Prediction of secondary structures of proteins by sequence processing employing an accessible and ultra-parallelized language model

**Palavras-chave em inglês:**

Bioinformatics

Sequence processing

Proteins - Structure

**Área de concentração:** Físico-Química

**Titulação:** Doutor em Ciências

**Banca examinadora:**

Leandro Martínez [Orientador]

Márcia Cristina Breitkreitz

Sandra Eliza Fontes de Avila

Laurent Emmanuel Dardenne

André Farias de Moura

**Data de defesa:** 27-05-2024

**Programa de Pós-Graduação:** Química

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0003-1953-1576>

- Currículo Lattes do autor: <http://lattes.cnpq.br/3423275260374279>

## **BANCA EXAMINADORA**

**Prof. Dr. Leandro Martínez (Orientador)**

**Profa. Dra. Sandra Eliza Fontes de Avila (Instituto de Computação - UNICAMP)**

**Profa. Dra. Marcia Cristina Breitkreitz (Instituto de Química - UNICAMP)**

**Dr. Laurent Emmanuel Dardenne (Laboratório Nacional de Computação Científica)**

**Prof. Dr. André Farias de Moura (Universidade Federal de São Carlos)**

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

Este exemplar corresponde à redação final da Tese de Doutorado defendida pelo aluno Guilherme Zainotti Miguel Fahur Bottino, aprovada pela Comissão Julgadora em 27 de Maio de 2024.

*That you are here — that life exists and identity,  
That the powerful play goes on, and you may contribute a verse.*

WHITMAN, Walt. *Leaves of grass.*

New York, NY: New American, c1954. 430 p.

# Agradecimentos

Antes de mais nada, agradeço a Deus. O processo de confecção desta Tese foi permeado por conversas com Ele, dos mais variados tipos e teores, com saldo absolutamente positivo. Viver uma vida a serviço das Ciências da Natureza fica menos frustrante, mais bonito e menos solitário na presença constante Dele, e embora eu não precise recorrer a Ele para apreciar os padrões do universo, apreciar sua fabulosa harmonia junto Dele é muito satisfatório.

Depois, preciso agradecer muito (**mesmo!**) ao meu orientador, Leandro Martínez. O Leandro acreditou em mim quando não tinha motivo nenhum pra isso, não precisava, e não foi apenas uma vez. Ele acreditou em mim quando eu quis escolher um projeto que não tinha muito a ver com o laboratório; quando eu escolhi soluções inusitadas para os problemas que encontrei; quando eu disse que estava trabalhando nos resultados, mesmo estando longe; quando eu precisei de um tempo para organizar a minha vida. O Leandro me educou pra ser cientista, e fez isso muito bem. Eu espero poder fazer pelo menos metade do que ele fez por mim por outras pessoas que vierem no meu caminho.

Sou muito grato também ao meu pai Sidney, minha mãe Luciana e minha irmã Luiza, que não desanimaram mesmo diante de várias das maiores provações enfrentadas por nos nas ultimas décadas, juntas. Esses últimos anos foram muito difíceis, mas também foram muito interessantes. A vocês, agradeço pelo pavimento firme que lançaram, do inicio ate o momento em que puderam; pela fundação bem assentada, as paredes altas e o teto rígido. Agradeço todo dia, por me dar excelentes motivos para ir e motivos melhores ainda para voltar. Vocês são verdadeiros heróis.

Agradeço aqueles que me aguentavam diariamente durante esse doutorado, a começar por aqueles que conviviam comigo primeiro durante os anos de trabalho presencial, dentro e fora do laboratório (Állan, Diego, Brenda, Adriano, Vinicius e Ander) e no pouco período depois da

pandemia, em que estávamos redescobrindo o instituto depois de quase dois anos longe (Pamella, Felipe e Eduard).

Agradeço também aos que conviviam comigo diariamente depois de chegar do trabalho, e aqui estou certamente falando da saudosa República do Patrão, minha família longe da família. Era sempre a eles que eu recorria quando precisava de algum apoio, e eles sempre estavam lá, até mesmo quando a pandemia de COVID-19 nos obrigou a nos distanciar. Inclusive foi dessa República e dos seus agregados que surgiu a minha principal rede de apoio, que sustentou minha sanidade durante o isolamento. Portanto, não poderia deixar de agradecer ao I.D.R.B., ao L.M.C., ao L.R.T.B., ao J.V.R.A., ao J.P.M., ao H.A.G.P., a B.V.C., ao E.B.S.M. e ao I.B.S.M. por terem sido meus companheiros (alguns deles, praticamente diários) durante os anos de pandemia. Finalizando o capítulo de Patrões, com certeza merecedor de uma menção honrosa, agradeço do fundo do coração ao meu glorioso amigo de quase 15 anos de IQ-UNICAMP Henrique Caracho, meu grande parceiro antes, durante e - tomara - depois de toda a pós-graduação, companheiro de noites viradas depurando código, cervejas comemorando resultados, partidas de videogames dos anos 2000, shows de rock no interior paulista, provas de disciplinas esquisitas e muitas, (*muitas*) caronas.

Agradeço pela oportunidade de ter me reencontrado com “os seis” - A.B.P., B.C.C., F.C.L, T.A.F.O., G.S.M. e V.O.B. - durante o doutorado, cuja fotografia alegra a escrivaninha onde escrevo essa Tese. Entre casamentos, shows, *newsletters*, praias e campos, saibam que esses encontros na reta final foram essenciais para terminar isso tudo.

Esse doutorado se tornou possível a partir de uma ideia absolutamente maluca e improvável, e a ideia, em si, foi fundamental. E ha agradecimentos a serem feitos nesse sentido. Eu preciso agradecer a G.S.S., por ter me aguentado enquanto eu não tinha encontrado a ideia ainda; a L.M.G., que me lembrou da importância de voltar a procurar a ideia quando eu tinha me esquecido; a M.T.A., que estava lá no momento exato (hora, minuto e segundo) em que eu tive a ideia; a N.U., que estava lá no dia que eu fiz o *release* no github.

Escrever a Tese, em si, também não foi fácil, e eu preciso agradecer muito, *muito* a um grupo de amigos improváveis carinhosamente apelidado de "o Boteco", que me deu forças entremeio a solidão e a inércia para compor esse trabalho escrito. Acho que, sem eles, teria sido diferente.

Agradeço à CAPES pelo apoio ao Programa de Pós-Graduação em Química do

Instituto de Química da UNICAMP. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Agradeço ao CNPq e ao Programa de Bolsas Institucionais do PPGQ pelo meu estipêndio de estudos. O trabalho contou com apoio financeiro do CNPq via Bolsa de Doutorado (Processo CNPq 140317/2019-8).

Por fim, esse projeto foi vinculado ao Cepid (Centro de Pesquisa, Inovação e Difusão) CCE&S - *Center for Computing in Engineering and Sciences*, apoiado financeiramente pela FAPESP (Processos 2010/16947-9, 2013/08293-7, 2016/13195-2, 2018/14274-9, 2018/24293-0, 2019/17007-4). O CCE&S é um centro de pesquisa humanamente, intelectualmente e instrumentalmente incrível, do qual espero participar e contribuir em outras oportunidades.

# Resumo

Grandes avanços foram feitos na bioinformática através da implementação de algoritmos de Inteligência Artificial (IA). Especialmente na previsão da estrutura de proteínas, modelos de linguagem extensivos podem ser usados para modelar estruturas nativas quando pouca ou nenhuma informação de homologia está disponível. No entanto, à medida que esses modelos se tornam maiores e mais sofisticados, sua acessibilidade a pesquisadores e usuários, bem como seus custos de treinamento e execução, tornam-se uma preocupação. Modelos de pequena escala que podem manter um nível competitivo de precisão com apenas uma fração do tamanho, tempo de execução e comprimento do código são, portanto, importantes.

Neste trabalho, apresentamos TintiNet.jl (<https://github.com/Hugemiler/TintiNet.jl>), uma abordagem contemporânea para a estimativa de propriedades estruturais de proteínas - estrutura secundária, ângulos  $\phi$  e  $\psi$  e acessibilidade ao solvente - utilizando apenas sua sequência como *input*. Usando arquiteturas de redes neurais contemporâneas, desenvolvemos um modelo de linguagem de tamanho reduzido, combinando os módulos Inception e Transformer, que é capaz de alcançar o mesmo desempenho – ou ligeiramente superior - nas tarefas de regressão e classificação, quando comparado aos algoritmos de ponta mais recentes, com apenas uma fração do número de parâmetros. Nosso modelo também alcança o menor tempo para gerar uma previsão por sequência entre todos os modelos avaliados. Sendo rápida e leve, nossa arquitetura pode ser executada em dispositivos simples com recursos limitados. Além disso, o a complexidade reduzida e os princípios de design permitem um acesso transparente aos parâmetros do modelo e aos pesos de atenção, viabilizando pequenos experimentos de interpretação da relação entre sequências e propriedades. Este trabalho tem o potencial de ampliar a acessibilidade da pesquisa de IA para Proteômica Estrutural.

# Abstract

Great advances have been made in bioinformatics through the implementation of Artificial Intelligence (AI) algorithms. Especially in protein structure prediction, extensive language models can be used to model native structures when little or no homology information is available. However, as these models become larger and more sophisticated, their accessibility to researchers and users, as well as their training and execution costs, become a concern. Small-scale models that can maintain a competitive level of accuracy with just a fraction of the size, runtime, and code length are therefore important.

In this work, we present TintiNet.jl (<https://github.com/Hugemiler/TintiNet.jl>), a fresh approach for estimating protein structural properties - secondary structure,  $\phi$  and  $\psi$  angles, and solvent accessibility - using only its primary sequence as *input*. Through contemporary neural network architectures, we developed a small-size language model, combining Inception and Transformer modules, that is capable of achieving the same – or slightly better – performance on regression and classification tasks, when compared to the latest state-of-the-art algorithms with just a fraction of the number of parameters. Our model also achieves the shortest time to generate a prediction per sequence among all the models evaluated. Being fast and lightweight, our architecture can run on simple devices with limited resources. Furthermore, the reduced complexity and design principles allow transparent access to model parameters and attention weights, enabling small experiments to interpret the relationship between protein sequences and their properties. This work has the potential to expand the accessibility of AI research for Structural Proteomics.

# List of Figures

1.1	Representação ilustrativa da sequência primária de uma proteína. O <i>backbone</i> da proteína, sua cadeia principal, é conectado por ligações peptídicas. . . . .	23
1.2	Representação ilustrativa dos quatro níveis de estruturação da proteína, as estruturas primária, secundária, terciária e quaternária. . . . .	25
1.3	Representação ilustrativa do funil de Dill. No topo do funil, as diversas maneiras pelas quais a proteína pode estar desenovelada ocupam os estados de maior energia livre. Por meio de mecanismos convergentes determinados pelo formato do funil (que, por sua vez, é determinado pela sequência da proteína), o enovelamento procede com a diminuição do número de estados degenerados até que o estado nativo seja alcançado. . . . .	27
1.4	Detalhamento das propriedades estruturais citadas que foram abordadas neste trabalho. (1) Estrutura secundária categórica SS3. (2, 3) Angulos ( $\varphi, \psi$ ). (4) Área superficial acessível ao solvente (SASA). . . . .	37
4.1	Sumário visual da arquitetura completa do TintiNet.jl, mostrando as redes Classificadora e Regressora em paralelo. Destaque para a estrutura da cabeça de previsão, que é diferente em ambas as redes. As setas pontilhadas mostram o significado aproximado do conteúdo do vetor codificado em cada posição da sequência de entrada no modelo . . . . .	52
4.2	Estrutura detalhada do bloco InceptiGOR de convoluções largas e paralelas. Observe os 8 blocos paralelos, com as reduções de dimensionalidade aprendidas, processamento paralelo de kernels de resíduos crescentes e maxpooling. Em azul, o número de parâmetros aprendidos para cada operação, considerando uma entrada de 128 canais (d=128). . . . .	53

5.1	Curvas de aprendizado para o modelo de Classificação da TintiNet.jl. O monitoramento ocorreu durante 50 épocas. A esquerda, acompanha-se a função objetivo medida como Entropia Cruzada mascarada, nos conjuntos de treino e validação, para cada um dos 10 <i>folds</i> , onde o eixo vertical representa a soma da entropia cruzada, normalizada para o tamanho do <i>fold</i> de validação. A direita, a acurácia do modelo nos conjuntos de treino e validação, para cada um dos 10 <i>folds</i> . . . . .	64
5.2	Curvas de aprendizado para o modelo de Regressão da TintiNet.jl. O monitoramento ocorreu durante 100 épocas. A função objetivo foi o Erro quadrático médio mascarado, nos conjuntos de treino e validação, para cada um dos 10 <i>folds</i> . . . . .	65
5.3	Distribuição das figuras de mérito de Classificação (Q3 e SOV) para os quatro modelos avaliados no conjunto VF3162. As linhas horizontais representam os valores médios. . . . .	66
5.4	Matrizes de confusão multiclasse para o conjunto VF3162 em todos os 4 modelos avaliados. A diagonal principal (cor azul) indica as classificações corretas. Na configuração escolhida, as linhas da matriz devem somar para 100% em todos os casos. . . . .	68
5.5	Avaliação dos méritos derivados do domínio de NLP na arquitetura TintiNet.jl. Acurácia de classificação em função do comprimento da sequência para o conjunto VF. A linha vermelha mostra o resultado de uma regressão linear. . . . .	71
5.6	Avaliação dos méritos derivados do domínio de NLP na arquitetura TintiNet.jl. Acurácia de classificação em função de MSA Neff de HHBlits para o conjunto VF. A linha vermelha mostra o resultado de uma regressão linear. . . . .	72
5.7	Avaliação dos méritos derivados do domínio de NLP na arquitetura TintiNet.jl. Acurácia de classificação para cada estado SS alvo por identidade de aminoácidos, ordenados por abundância (barras cinzas). As linhas horizontais tracejadas mostram a acurácia média de classificação para cada estado, codificado por cores. . . . .	73
5.8	Distribuição proporcional entre os estados SS3, para os 20 aminoácidos naturais, para todas as proteínas da base de dados CATHS40. . . . .	74

5.9	Distribuição das figuras de mérito de Regressão (MAE e cMAE) para os quatro modelos avaliados no conjunto VF3162. As linhas horizontais representam os valores médios. . . . .	75
5.10	Gráfico de Ramachandran computado para o <i>fold</i> 10 de validação. Cores mais escuras representam regiões mais permitidas. Destaque para as regiões mais densas do gráfico: no topo e a esquerda, a região de fitas e folhas-beta; no centro, a esquerda, a alfa-hélice, e no centro a direita, a hélice canhota . . . . .	76
5.11	Gráfico de Ramachandran para cada um dos 20 aminoácidos naturais e para o aminoácido desconhecido (X), para todas as proteínas do fold 10 do conjunto VF3162. . . . .	78
5.12	<i>Boxplots</i> comparando a distribuição de referência e estimada das áreas acessíveis ao solvente, para cada um dos 20 aminoácidos naturais, no fold10/VF3162. . . . .	81
5.13	<i>Benchmark</i> computacional da arquitetura TintiNet.jl comparada aos outros preditores. Tempo real de execução para previsão em função do tamanho da amostra (em número de sequências) para todas as cinco replicatas. As linhas tracejadas são resultados de regressão linear. . . . .	82
5.14	Sinal de saída da última camada InceptiGOR8 da parte convolucional do modelo TintiNet.jl para o alvo 5CXOB01. As anotações na cor laranja ilustram as classes que seriam estimadas para cada bloco resultante da análise hierárquica de agrupamentos se a classificação fosse realizada apenas com a saída da camada convolucional. . . . .	86
5.15	Pesos de atenção médios recebidos por cada aminoácido da sequência do alvo CATH-1C75A01 em cada cabeça de detecção das camadas BERT em tempo de inferência de SS3 . . . . .	87
5.16	(A) Matriz de atenção da cabeça 6 da primeira camada Transformer do componente BERT da TintiNet.jl durante a inferência de estrutura secundária categórica (SS3) para o alvo CATH-1C75A01. (B) Detalhes estruturais do alvo CATH-1C75A01 evidenciando as duas conformações da LYS10. Imagem gerada com Mol* Viewer [90] . . . . .	89
5.17	Sinal de saída da última camada InceptiGOR8 da parte convolucional do modelo TintiNet.jl para o alvo 5CXOB01. . . . .	91

5.18	Pesos de atenção médios recebidos por cada aminoácido da sequência do alvo CATH-5CXOB01 em cada cabeça de detecção das camadas BERT em tempo de inferência de SS3 . . . . .	93
5.19	Matrizes de atenção das cabeças 6 (A) e 2 (B) da primeira camada Transformer do componente BERT da TintiNet.jl durante a inferência de estrutura secundária categórica (SS3) para o alvo CATH-5CXOB01. (C) Detalhe estrutural da interação entre ARG7 e outros resíduos com grande grau de separação na sequência primária. (D) Detalhe do núcleo hidrofóbico do alvo CATH-5CXOB01 e alguns de seus componentes com cadeias laterais aromáticas. Imagem gerada com Mol* Viewer [90] . . . . .	96

# **Lista de Tabelas**

5.1	Resumo das Figuras de Mérito para Classificação e Regressão para o modelo TintiNet.jl comparado com SPIDER3-single, ProteinUnet e SPOT1D-single . . . . .	67
5.2	Número aproximado de parâmetros totais dos modelos avaliados . . . . .	67
5.3	Figuras de mérito para a classificação binária “um-contra-todos” por categoria de estrutura terciária . . . . .	70
5.4	Tempo real de execução da previsão de estrutura secundária nos experimentos de <i>benchmark</i> . . . . .	83

# **Lista de Abreviaturas**

AA	Aminoácido
ACC	Notação alternativa para SASA utilizada no software DSSP (ver SASA)
BERT	Em inglês, Bidirectional Encoder for Transformers, ou Codificador Bidirecional para <i>Transformers</i> .
CASP	Em inglês, <i>Critical Assessment of Structure Prediction</i> , ou Avaliação Exigente de Predição Estrutural, uma competição internacional de modelagem biomolecular
CATH	Em inglês, <i>Class, Architecture, Topology/fold, Homologous superfamily</i> , uma base de dados de classificação estrutural de proteínas previamente caracterizadas.
cMAE	Em inglês, <i>Circular Mean Absolute Error</i> , ou Erro Absoluto Médio Circular (angular).
CryoEM	Crio-microscopia eletrônica.
DSSP	<i>Dictionary of Protein Secondary Structure</i> , software de cálculo de estruturas secundárias.
GOR	Garnier-Osguthorpe-Robson, método de estimativa da estrutura secundária proposto em 1970
GPU	Em inglês, <i>Graphics Processing Unit</i> , ou Unidade de processamento gráfico
HPC	Em inglês, <i>High Performance Computing</i> , ou Computação de Alto Desempenho
IA	Inteligência Artificial
ML	Em inglês, <i>Machine Learning</i> , ou Aprendizado de Máquina.
MSA	Em inglês, <i>Multiple Sequence Alignment</i> , ou Alinhamento de Múltiplas Sequências.

MAE	Em inglês, <i>Mean Absolute Error</i> , ou Erro Absoluto Médio.
MSE	Em inglês, <i>Mean Square Error</i> , ou Erro Quadrático Médio.
NLP	Em inglês, <i>Natural Language Processing</i> , ou Processamento de Linguagens Naturais
PDB	<i>Protein Data Bank</i>
Q3	Acurácia da classificação da estrutura secundária categórica em 3 estados
RMN	Ressonância Magnética Nuclear
SASA	Em inglês, <i>Solvent-Accessible Surface Area</i> , ou Área Superficial Acessível ao Solvente
SOV	Segment Overlap Score
SS3	Estrutura secundária categórica em 3 estados
TPU	Em inglês, <i>Tensor Processing Unit</i> , ou Unidade de processamento tensorial

# Sumário

<b>1</b>	<b>A nova era da Bioinformática Estrutural</b>	<b>21</b>
1.1	Muito além de uma fórmula molecular . . . . .	21
1.2	Estruturas de Estruturas . . . . .	22
1.3	O que determina o estado nativo de uma proteína? . . . . .	24
1.3.1	O Paradoxo de Levinthal . . . . .	25
1.3.2	A Hipótese de Anfinsen . . . . .	26
1.3.3	O Funil de Enovelamento de Dill . . . . .	26
1.3.4	O eixo Sequência-Estrutura-Função . . . . .	28
1.4	O problema da modelagem de proteínas . . . . .	28
1.4.1	Afinal, <i>para que</i> modelar de proteínas? . . . . .	29
1.4.2	Desenvolvimento do problema . . . . .	30
1.4.3	Problemas complexos exigem soluções sofisticadas . . . . .	31
1.4.4	IA: de coadjuvante a protagonista . . . . .	32
1.4.5	Como assim, <i>modelo de linguagem?</i> . . . . .	33
1.4.6	Breves considerações de ordem Filosófica . . . . .	34
1.5	Ensinar uma máquina não é barato! . . . . .	35
1.5.1	Existe modelagem além do <i>end-to-end?</i> . . . . .	36
1.5.2	Contra o fluxo . . . . .	39
1.6	Uma pequena brincadeira com ChatGPT . . . . .	39
<b>2</b>	<b>Objetivo</b>	<b>41</b>
<b>3</b>	<b>Abordagem</b>	<b>42</b>

3.1	Um modelo unidimensional, acessível em três dimensões . . . . .	43
3.2	Revisão bibliográfica . . . . .	43
<b>4</b>	<b>Metodologia Desenvolvida</b>	<b>46</b>
4.1	Obtenção e pré-processamento dos dados . . . . .	46
4.1.1	Seleção da fonte de dados e obtenção de sequências e estruturas .	46
4.1.2	Extração das estruturas secundárias e demais propriedades unidimensionais . . . . .	47
4.1.3	Construção da base de dados em formato universal de acesso aberto	49
4.2	Arquitetura e Implementação do modelo preditivo . . . . .	49
4.2.1	O que significa “arquitetura” de uma rede neural? . . . . .	49
4.2.2	Concepção da Arquitetura TintiNet.jl . . . . .	51
4.2.3	Treinamento da rede neural . . . . .	54
4.3	Métricas de análise dos resultados . . . . .	55
4.3.1	Conjuntos de dados para avaliação . . . . .	55
4.3.2	Métricas de avaliação de classificação . . . . .	57
4.3.3	Métricas de avaliação de regressão . . . . .	59
4.4	<i>Benchmark</i> computacional . . . . .	60
4.5	Interpretação do modelo . . . . .	60
4.5.1	Interpretando operações de atenção . . . . .	61
4.5.2	Protocolo geral de interpretação . . . . .	62
<b>5</b>	<b>Resultados e Discussões</b>	<b>63</b>
5.1	Treinamento do modelo . . . . .	63
5.1.1	Curvas de Aprendizado . . . . .	63
5.2	Resultados da validação cruzada . . . . .	64
5.3	Análise dos méritos do modelo Classificador . . . . .	65
5.3.1	Dependência da classificação com propriedades genéricas da sequência	70
5.4	Análise dos méritos do modelo Regressor . . . . .	74
5.4.1	Dependência de desempenho de regressão com propriedades de sequência . . . . .	75
5.4.2	<i>Benchmark</i> de tempo de execução . . . . .	82

5.5	Interpretação do modelo . . . . .	83
5.5.1	Estudo de caso 1: CATH-1C75A01 . . . . .	84
5.5.2	Estudo de caso 2: CATH-5CXOB01 . . . . .	89
5.5.3	Observações gerais sobre os estudos de caso . . . . .	95
5.6	Limitações . . . . .	98
<b>6</b>	<b>Considerações Finais</b>	<b>100</b>
6.1	Perspectivas . . . . .	100
	<b>Bibliografia</b>	<b>102</b>

## Capítulo 1

# A nova era da Bioinformática Estrutural

*"Can we predict how proteins will fold? Out of a near infinitude of possible ways to fold, a protein picks one in just tens of microseconds. The same task takes 30 years of computer time"[1]*

A qualquer momento, em todo tipo de organismo vivo, existem bilhões de pequenas moléculas trabalhando incessavelmente para mediar os processos biológicos. Elas são, cada uma com sua especialidade, responsáveis por promover a realização de reações químicas que afetam desde nossa capacidade de andar ou enxergar, até a interpretação do nosso código genético. A principal classe dessas moléculas, compostas por longas cadeias de aminoácidos conectados em uma ordem específica, recebe o nome de **proteínas**, e é por meio do esforço *internacional e multidisciplinar* de investigar seus formatos e os trabalhos que realizam, que surgem as grandes revoluções na medicina, no setor produtivo e na própria compreensão científica sobre o funcionamento da vida.

### 1.1 Muito além de uma fórmula molecular

Uma das definições mais primitivas do significado químico da palavra proteína poderia ser “um copolímero de L- $\alpha$ -aminoácidos naturais conjugados por ligações peptídicas”. Ela certamente serve ao propósito de apresentar proteínas como moléculas formadas pela associação de unidades monoméricas bem determinadas, com uma composição e conectividade conhecidas,

para cada exemplar. Apesar de simples, essa definição é convenientemente correta do ponto de vista da Química Orgânica, e por isso é encontrada - com pequenas modificações - em livros de Química Geral [2].

Partindo dessa definição, parece lógico que o passo inicial do estudo de qualquer proteína é descobrir quais são os L- $\alpha$ -aminoácidos que lhe dão origem. Contudo, conhecer apenas a composição elementar da cadeia não é suficiente, porque, para que uma proteína consiga desempenhar sua função biológica, essa cadeia polipeptídica precisa adquirir uma estrutura tridimensional *específica e resiliente* que viabilize os processos que a envolvem [3]. Essa estrutura - ou, mais precisamente, conjunto de estruturas - é denominada **estado nativo**, e é univocamente determinada não somente pela composição da proteína, mas também pela ordem dos aminoácidos na cadeia principal.

Por isso, saber a combinação de *composição e ordem* dos monômeros na cadeia polipeptídica que será sintetizada pelo ribossomo é sempre o primeiro passo para investigar uma proteína estruturalmente. Justamente por isso, essa **sequência de aminoácidos** recebe o nome de **estrutura primária**. Nela, podemos distinguir dois componentes estruturais relevantes: o esqueleto (do inglês **backbone**) da proteína, considerado a cadeia principal da molécula, que engloba os terminais da proteína e todas as **ligações peptídicas** entre os aminoácidos; e o conjunto de *cadeias laterais* dos aminoácidos, que se conectam aos carbonos-alfa do *backbone* e determinam, coletivamente, a estrutura e as propriedades da molécula (Figura 1.1).

Para entender a origem dessa estruturação que a proteína sofre em solução, é importante lembrar que o processo de síntese proteica ocorre de forma sequencial e ordenada, aminoácido por aminoácido, do início ao fim da fita de RNA-mensageiro. Conforme a proteína é construída, ela está sujeita às múltiplas interações moleculares que existem, naturalmente, no meio fisiológico. Essas interações, de diferentes tipos e intensidades, são a base do processo físico de **enovelamento** da proteína, por meio do qual a sequência primária da proteína obtém seu formato funcional [4].

## 1.2 Estruturas de Estruturas

Embora não seja objetivo deste trabalho realizar um tratado abrangente sobre o mecanismo de enovelamento de proteínas - a *cinética do enovelamento de proteínas* é uma disciplina extensa e a diversidade dos mecanismos de enovelamento é grande - descrevê-lo em

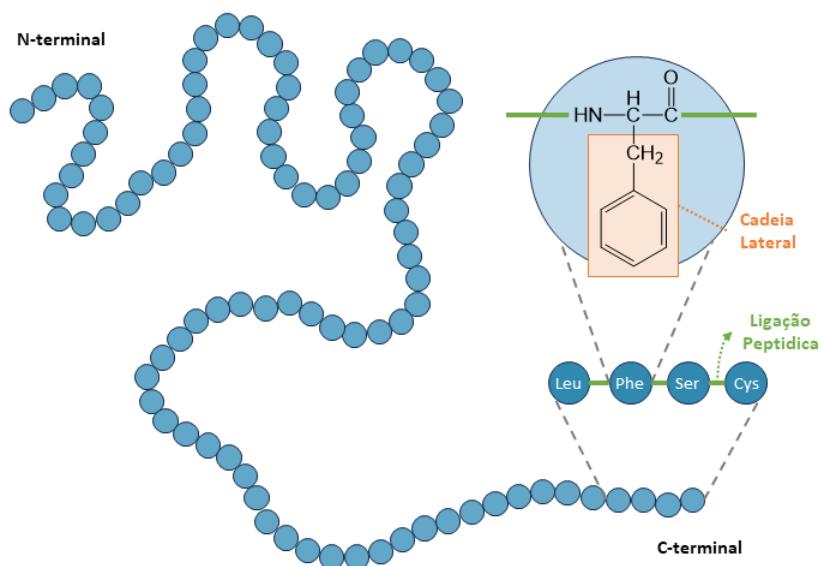


Figura 1.1: Representação ilustrativa da sequência primária de uma proteína. O *backbone* da proteína, sua cadeia principal, é conectado por ligações peptídicas.

linhas gerais é fundamental para as discussões a seguir. É razoável dividir esse processo em dois grandes subprocessos, que se diferenciam em relação (1) ao nível de escala, (2) grau de localidade na sequência, (3) componente estrutural envolvido e (4) o tipo de interação protagonista.

O primeiro grande subprocesso ocorre em segmentos da estrutura primária ainda durante o processo de síntese proteica. Ele é protagonizado por ligações de hidrogênio entre os componentes do *backbone* de aminoácidos com pequeno grau de separação na sequência primária. Esse processo tem como resultado pequenos **núcleos de enovelamento** localmente estruturados.

O segundo grande processo envolve o **agregamento** - também chamado de **colapso** ou **coalescência** - desses núcleos de enovelamento por meio da interação entre as cadeias laterais dos aminoácidos que os compõem. Aqui, tornam-se protagonistas as interações de Van der Waals entre as cadeias laterais de núcleos de enovelamento com maior grau de separação na sequência primária. A **proteína enovelada** é o produto desse colapso, no qual também ocorre algum refinamento da estrutura do *backbone* e a relaxação rotacional das cadeias laterais para minimizar efeitos estéricos.

Da mesma forma que se apresentou o enovelamento de proteínas dividido em dois grandes processos, o seu resultado final - ou seja, a estrutura proteína enovelada - pode também ser interpretado com dois níveis de detalhamento diferentes. O primeiro deles é a chamada **estrutura secundária** da proteína. Ela é largamente influenciada pelo primeiro processo de

enovelamento apresentado, dado que diz respeito à estruturação local do *backbone*. De maneira geral, existem duas formas primordiais de estrutura secundária, a  **$\alpha$ -hélice** e a **fita- $\beta$** .

A  $\alpha$ -hélice apresenta uma configuração em espiral, onde cada grupo N – H da estrutura cede uma ligação de hidrogênio ao grupo C = O da estrutura do aminoácido quatro unidades à frente na sequência. Já a fita- $\beta$  é um segmento da cadeia de polipeptídeos, geralmente compreendendo entre 3 a 10 aminoácidos, com sua estrutura básica quase completamente estendida. Duas ou mais cadeias de polipeptídeos de fita beta adjacentes, sejam paralelas ou antiparalelas, são estabilizadas por ligações de hidrogênio, podendo formar uma superestrutura denominada **folha- $\beta$** .

O segundo nível de apreciação da proteína enovelada é a sua própria estrutura tridimensional, que recebe o nome de **estrutura terciária** e é determinada pela coordenadas em 3D de cada átomo que a perfaz.

Essa estrutura - e por consequência, as relações espaciais entre cada aminoácido - é virtualmente única para cada sequência primária, e é a grande responsável por definir o funcionamento da proteína. Diferentemente da estrutura secundária, onde existem dois grandes representantes e algumas pequenas subdivisões, encontramos uma infinidade de motivos estruturais (do inglês, *structural motifs*) que abstraem o formato da proteína. Esses *motifs* recebem nomes interessantes que refletem a posição relativa dos segmentos de estrutura secundária coalescida, como “barril-beta”, “ferradura-alfa”, “concha” e “sanduíche-beta”. Isso reforça que, embora sejam conceitualmente distintas, as estruturas secundária e terciária da proteína estão intrinsecamente conectadas.

Vale comentar que existe ainda um estado acima da estrutura terciária, denominado estrutura quaternária, que representa a correta associação de diversas cadeias com suas estruturas terciárias próprias em um complexo proteico. Como elas dizem respeito a mais de uma proteína, transcendem o escopo deste projeto e não serão aprofundadas neste trabalho. A Figura 1.2 ilustra os níveis de estruturação mencionados nesta seção.

### 1.3 O que determina o estado nativo de uma proteína?

Dentre todos os cientistas que procuraram entender o fenômeno do enovelamento das proteínas em solução, devemos uma menção histórica aos esforços de Levinthal, Anfinsen e Dill.

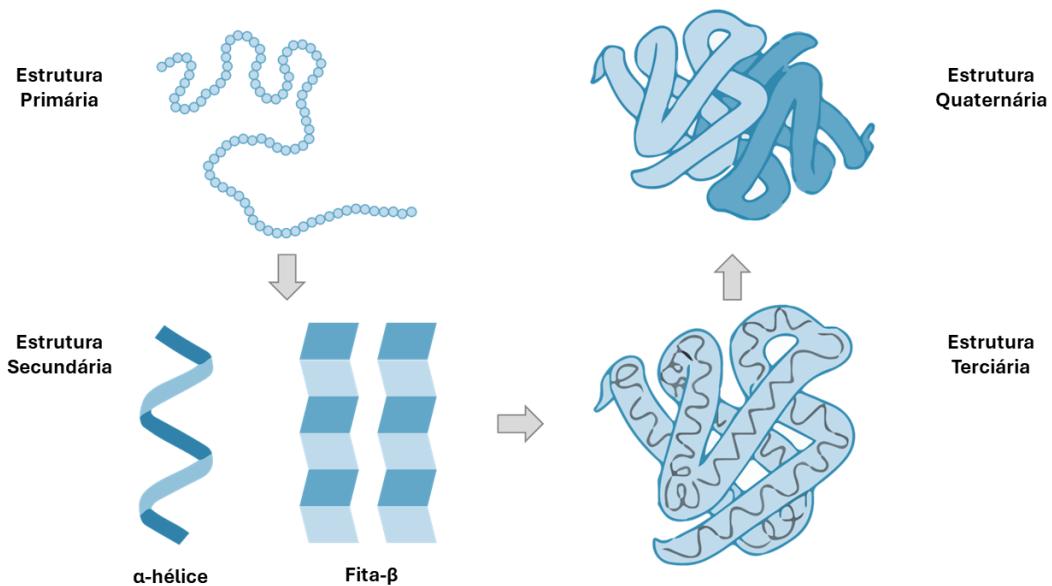


Figura 1.2: Representação ilustrativa dos quatro níveis de estruturação da proteína, as estruturas primária, secundária, terciária e quaternária.

### 1.3.1 O Paradoxo de Levinthal

Cronologicamente, o enunciado do Paradoxo de Levinthal, por Cyrus Levinthal em 1969 [5], foi um dos marcos iniciais para o entendimento do processo de enovelamento das proteínas. Trata-se de um conceito da biologia molecular que questiona como proteínas conseguem alcançar sua forma funcional enovelada na escala de tempo real em que esse fenômeno ocorre, apesar do enorme número de configurações possíveis para a molécula [6].

Para ilustrá-lo, considere uma proteína que possui cerca de 100 aminoácidos. A princípio, há várias ligações que têm liberdade rotacional no *backbone*, mas, por simplicidade, suponhamos que existam apenas duas conformações possíveis para cada aminoácido. Mesmo nesse caso *extremamente reducionista*, isso já resultaria em  $2^{100}$  possíveis conformações para a proteína como um todo - um número astronomicamente grande. Ainda que, nesse caso, cada transição conformacional levasse apenas um picosegundo ( $2 \cdot 10^{-12}$  segundos) para acontecer, ainda seriam necessários 40 bilhões de anos para visitar todas as possíveis conformações, numa busca aleatória equiprovável sem repetições.

Contudo, já na década de 1960, sabia-se que o enovelamento de proteínas desse tamanho ocorria em uma escala real de segundos a minutos (essa informação foi obtida por meio de experimentos de desnaturação térmica e renaturação realizados em laboratório [7]).

Constitui-se, portanto, um paradoxo entre a quantidade de estados possíveis na *hipótese equiprovável*, e o tempo real que o enovelamento leva para acontecer.

### 1.3.2 A Hipótese de Anfinsen

O próprio Levinthal já havia postulado que uma possível solução do paradoxo seria que o processo de dobramento de proteínas não é uma busca conformacional aleatória e equiprovável, mas sim guiado por vias determinísticas com probabilidades diferentes para cada transição, o que reduziria drasticamente o número de configurações que precisam ser exploradas.

Coube a Christian Anfinsen, que recebeu o prêmio Nobel em 1972 “*pelo seu trabalho sobre ribonuclease, especialmente no que diz respeito à conexão entre a sequência de aminoácidos e a conformação biologicamente ativa*”, oferecer uma explicação físico-química para as observações de Levinthal, por meio da sugestão da Hipótese Termodinâmica do enovelamento de proteínas [8]. Nela, afirma que a estrutura tridimensional de uma proteína nativa em seu meio fisiológico normal (solvente, pH, força iônica, presença de outros componentes como íons metálicos ou grupos prostéticos, temperatura, etc.) é aquela em que a energia livre de todo o sistema é mais baixa; isto é, a conformação nativa - ou o conjunto de conformações que a proteína pode assumir em seu estado nativo - em um ambiente é determinada pela totalidade das interações interatômicas dentro e ao redor da proteína, e, portanto, pela sequência de aminoácidos.

A conjectura final dessa definição é que, dado um conjunto definido de propriedades do ambiente da proteína - que em meio fisiológico é sempre garantido pela estabilidade homeostática do meio intracelular -, **é a sequência primária que determina univocamente o espaço conformacional termodinamicamente acessível e resiliente da proteína enovelada.**

### 1.3.3 O Funil de Enovelamento de Dill

O Funil de Enovelamento de Dill é uma resposta concreta ao Paradoxo de Levinthal, que incorpora a hipótese termodinâmica de Christian Anfinsen para explicar como as proteínas se enovelam em suas estruturas tridimensionais específicas.

Proposto por Ken Dill nos anos 1990 [9], este modelo é baseado em dois princípios fundamentais, sendo o primeiro a abstração de uma *paisagem energética afunilada*. Dill sugere

visualizar a distribuição dos estados ao longo do enovelamento protéico como uma paisagem energética concreta em forma de funil. O topo do funil representa a proteína desenovelada, um estado de energia livre superior ao estado nativo. Conforme a proteína começa a dobrar-se, ela desce no funil, onde o número de conformações possíveis e a energia livre do estado parcialmente enovelado vai progressivamente se reduzindo. O fundo do funil representa o estado nativo da proteína, em que o sistema alcança o estado de menor energia possível e a proteína, sua estrutura tridimensional enovelada.

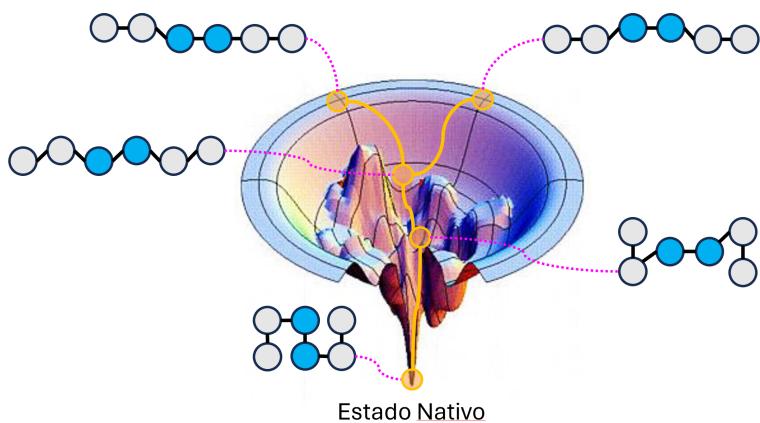


Figura 1.3: Representação ilustrativa do funil de Dill. No topo do funil, as diversas maneiras pelas quais a proteína pode estar desenovelada ocupam os estados de maior energia livre. Por meio de mecanismos convergentes determinados pelo formato do funil (que, por sua vez, é determinado pela sequência da proteína), o enovelamento procede com a diminuição do número de estados degenerados até que o estado nativo seja alcançado.

O segundo princípio dita que a superfície do funil não precisa ser necessariamente lisa. Há diversas rugosidades que representam transições de estado, devido às quais, ao contrário de um processo sequencial e único, o dobramento de proteínas pode potencialmente ocorrer através de múltiplos caminhos, que não são necessariamente idênticos do ponto de vista mecanístico, mas que alcançam o mesmo resultado. Existem duas consequências disso: a primeira, mais evidente, é que nem todos os caminhos de enovelamento são igualmente viáveis em meio fisiológico, ou seja, o enovelamento não acontece "de qualquer jeito"; contudo, é igualmente verdade que as proteínas podem descer o funil por mais de uma trajetória, contanto que os caminhos pela superfície de energia livre sejam suaves e tenham uma cinética acessível, eventualmente convergindo para intermediários parecidos e a mesma estrutura nativa. Isso significa que o processo de dobramento é robusto e eficiente, pois não depende de uma única sequência de eventos, o que - do ponto de vista biológico -

pode inclusive ser considerado um mecanismo elegante de resistência funcional. Uma representação ilustrativa do Funil de enovelamento de Dill pode ser apreciada na Figura 1.3.

### 1.3.4 O eixo Sequência-Estrutura-Função

A progressão do paradoxo de Levinthal para o dogma de Anfinsen e finalmente para o modelo do Funil de Dill ilustra uma cadeia contínua de ideias, onde a complexidade do enovelamento de proteínas é destilada num modelo determinístico, impulsionado pela sequência primária, alinhado com o imperativo termodinâmico da biologia molecular.

O fio que conecta as conclusões desses três trabalhos expostos foi, sozinho, responsável por revolucionar a maneira como a proteômica estrutural se desenvolveu e se desenvolve até hoje: é **a sequência de aminoácidos de uma proteína que codifica a topologia de toda a paisagem energética multidimensional que os aminoácidos inevitavelmente percorrerão durante o processo de enovelamento.** Essa topologia inclui, evidentemente, o estado nativo, sem limitar-se exclusivamente a ele. Por conta própria, a sequência de aminoácidos determina o formato e a rugosidade do funil de enovelamento (conforme os postulados de Dill), incluindo as estruturas secundária e terciária do estado de menor energia livre dentro desse funil (evocado por Anfinsen), e os múltiplos caminhos que levam a esse estado nativo por mecanismos bem definidos (e não por deslocamentos aleatórios, respondendo ao paradoxo de Levinthal).

## 1.4 O problema da modelagem de proteínas

A combinação dos princípios expostos culminou na enunciação formal do problema de modelagem computacional de proteínas. Em 2012, o próprio Ken Dill apresentou, em uma entrevista a revista Science [10], uma formulação em retrospecto simples - mas inevitavelmente precisa - desse problema que perdura mais de 50 anos, e o caminho lógico para sua solução. De acordo com ele,

*...podemos entender que esse problema se refere a três perguntas amplas:*

1. *Qual é o código Físico pelo qual uma sequência de aminoácidos dita o estado nativo de uma estrutura?*
2. *Como podem as proteínas se enovelar tão rápido?*
3. *É possível desenvolver um algoritmo computacional para prever estruturas de*

*proteínas a partir de suas sequências?*

A própria forma como o problema da modelagem de proteínas é enunciado pressupõe a validade dos desdobramentos dos trabalhos de Levinthal, Anfinsen e do próprio Dill. A modelagem de proteínas, sob essa ótica, passa portanto a ser primordialmente um problema de ponta a ponta, onde o processamento de sequências unidimensionais levaria, por processos arbitrariamente complexos, a estruturas tridimensionais.

### 1.4.1 Afinal, para que modelar de proteínas?

Antes de começar a falar do problema especificamente, precisamos entender o porque dele ser um problema de fato. Afinal, desde muito antes de Dill publicar suas ideias, John Kendrew e Max Perutz já haviam recebido o Prêmio Nobel [11, 12], em 1962 pela estrutura da mioglobina determinada por cristalografia de raios-X, e em 1992 o Brookhaven Data Bank - que se tornaria futuramente o Protein Data Bank - já tinha quase 1000 estruturas cristalográficas de proteínas depositadas em alta resolução [13]. Se existe então uma maneira de determinar experimentalmente a estrutura de uma proteína, para que precisamos de um programa de computador para tentar estimá-la?

Verdadeiramente, se existem condições viáveis para a realização de um experimento de cristalografia para uma determinada proteína-alvo, ele deve ser realizado. No entanto, isso é mais complexo e desafiador do que parece. As proteínas precisam ser purificadas em concentrações elevadas (o que já elimina diversos alvos a depender da sua fonte de obtenção) e, em seguida, submetidas a uma série de condições de cristalização que não são necessariamente conhecidas *a priori*, que envolvem variáveis como pH, temperatura, concentração de sal, e a presença de agentes precipitantes. A formação de cristais adequados para análise difratométrica é muitas vezes imprevisível e pode requerer meses *ou até anos* de tentativas, dado que cada proteína possui requisitos únicos de cristalização.

Uma vez obtidos os cristais, eles são irradiados com raios X em um sincrotron ou fonte similar, gerando padrões de difração que são analisados. A própria interpretação desses padrões para deduzir a estrutura atômica da proteína é um processo matematicamente intensivo, que envolve a solução do chamado “problema de fase” [14], que surge porque a técnica de difração de raios X fornece informações apenas sobre a magnitude dos fatores de estrutura, mas não sobre suas fases,

que são cruciais para reconstruir a imagem tridimensional da densidade eletrônica da proteína.

Isso sem mencionar a preocupação cada vez maior que a cristalografia de raios X fornece uma imagem estática da proteína, que pode não refletir adequadamente a gama de conformações que uma proteína pode adotar em solução, principalmente quando a cristalização é realizada com um agente precipitante ocupando um sítio no seu interior. As proteínas são moléculas dinâmicas, e mesmo no seu estado nativo, estão em constante movimento. Assim, enquanto a cristalografia de raios X pode revelar detalhes atônicos da estrutura proteica, ela também pode mascarar aspectos importantes da flexibilidade e dinâmica proteica, limitando a compreensão completa de seus mecanismos funcionais.

Em complemento, podemos mencionar que a escala de descobrimento de novas sequências de proteínas, principalmente por técnicas de sequenciamento metagenômico e *design* de proteínas, já ultrapassa muito a capacidade da pesquisa em cristalografia de proteínas, mesmo numa situação idealizada em que seria possível obter e cristalizar todas aquelas com sequências já conhecidas. Portanto, mesmo com a existência da cristalografia de proteínas e muitas outras técnicas utilizadas para fins similares como RMN e CryoEM, resolver o problema da modelagem de proteínas é a única saída escalável para fornecer de maneira imediata, para qualquer alvo, uma estrutura tridimensional razoável.

### 1.4.2 Desenvolvimento do problema

Embora as condições de contorno do problema da modelagem de proteínas sejam amplamente conhecidas há muitos anos e tenham provocado diversos avanços na proteômica estrutural, o desenvolvimento de abordagens centradas ao redor de compreender primordialmente a estrutura primária das proteínas como único *input* é relativamente recente. Traçando um breve panorama histórico, fica evidente que os esforços iniciais para estudar estruturas proteicas computacionalmente foram em sua maioria focados nas propriedades físicas e químicas dos aminoácidos, e a forma como eles se comportavam quando submetidos a simulações de dinâmica molecular em campos de forças clássicos (como nos trabalhos de Levitt, no final dos anos 80 [15, 16, 17, 18]).

Conforme as técnicas baseadas em campos de forças clássicos se desenvolveram, diversas tentativas de aperfeiçoamento foram na direção de customizar a forma funcional dos potenciais,

adicionando algumas vezes termos derivados de observações empíricas [19]. Quando as proteínas ficaram grandes demais e o tempo computacional deixou de ser prático para simulações longas, os métodos de modelagem baseada em conhecimento foram desenvolvidos. Nesse momento as bases de dados de sequências de proteínas haviam crescido substancialmente, e as ferramentas de busca eficiente haviam se desenvolvido a ponto de permitir buscas extensivas e alinhamentos múltiplos de sequências de proteínas (MSAs) em escalas de tempo de minutos ou horas. O paradigma não era “processar a sequência de aminoácidos”, mas sim “utilizar a sequência para obter o máximo de informação útil, e então processar essa informação”.

Todo tipo de estratégia era válida, sendo alguns exemplos os mapas de contato derivados de coevolução, fragmentos de proteínas conhecidas, estimativas iniciais de estrutura secundária, segmentos modelados por homologia com proteínas já caracterizadas [20, 21, 22, 23, 24]. Contudo, nenhum sistema havia obtido sucesso processando primordialmente a sequência de aminoácidos, sem o uso de dados auxiliares, para obter informação estrutural. Simplesmente não havia, naquele momento, um paradigma escalável, eficiente e universalizável de processamento de sequências capaz de aprender os tão complexos padrões por trás da geração de sequências de proteínas.

### 1.4.3 Problemas complexos exigem soluções sofisticadas

Padrões complexos são a regra no estado do conhecimento contemporâneo do século XXI: os fenômenos geradores dos observáveis no mundo real estão ficando cada vez mais multidimensionais e difíceis de descontaminar de ruídos. Essa realidade não se limita apenas aos padrões visíveis ou físicos, mas se estende ao domínio abstrato dos dados e, em última instância, a representação simbólica de dados da química e da biologia, como as fórmulas moleculares, níveis taxonômicos e sequências biológicas.

Conforme sublinhado em retrospectiva por Sergei Ovchinnikov (organizador do CASP), em 2021 [25], a inteligencia artificial começa a permear o problema da modelagem de proteínas de forma notável a partir do CASP12 (2016), substituindo técnicas estatísticas clássicas de processamento de dados nos diversos passos do protocolo de modelagem. Inicialmente, substituíram algoritmos como DCA [21] e EVCouplings [26] no processamento de MSAs para encontrar acoplamentos evolutivos que informam restrições na modelagem[27]. Posteriormente, trabalhando sobre o mesmo tipo de dado (MSAs), aperfeiçoaram as iterações anteriores dos

estimadores de estrutura secundária PSIPRED [28] e JPRED [29]. Também chegaram a ser utilizadas para refinamento e hibridização dos fragmentos obtidos por homologia para a posterior modelagem *ab initio*. Contudo, até antes do CASP14 (2020), ainda estavam limitadas ao processamento dos dados que saíam da busca em bases de dados.

#### 1.4.4 IA: de coadjuvante a protagonista

Foi apenas no ano de 2020 que a Inteligência Artificial proporcionou o que foi provavelmente o salto mais relevante na previsão de estruturas de proteínas, substituindo os motores de busca nas bases de dados, atuando em uma camada intermediária entre as sequências e os MSAs. Essa abordagem, juntamente com outros componentes, culminou na solução mundialmente famosa da empresa *Deepmind*, batizada de **AlphaFold2**, que efetivamente avançou de forma significativa o estado do problema de modelagem de proteínas.

Não é objetivo deste trabalho fazer uma análise aprofundada do algoritmo AlphaFold2, nem fazer um ensaio crítico sobre o entorno do seu desenvolvimento. Fundamental aqui é mencionar, contudo, que do ponto de vista puramente técnico, existem três condições que tornaram possível a criação deste modelo:

1. A experiência do AlphaFold1 [30], que, operando ainda sob o paradigma de IA-sobre-MSA, saturou o nível de performance alcançável neste paradigma;
2. O *boom* generalizado de experiências e conhecimentos adquiridos em tarefas sequenciais, como o desenvolvimento de tradutores, oriundo do campo do NLP;
3. O desenvolvimento da arquitetura Transformer de processamento de linguagens, por Vaswani et al. , em 2017 [31], que permitiu operações *paralelizáveis* e *escaláveis* de interação **global** entre todos os componentes de uma sequência de entrada.

De fato, foi justamente a incorporação de técnicas e arquiteturas de modelos de IA do domínio do processamento de linguagem na previsão da estrutura de proteínas que permitiu um salto tão significativo neste campo.

*Mas, um momento. o que proteínas tem a ver com linguagem? E qual é o sentido de aplicar modelos que funcionam bem como tradutores ou como classificadores de sentimentos a sequências de aminoácidos?*

### 1.4.5 Como assim, *modelo de linguagem?*

Modelos de linguagem são sistemas computacionais projetados para entender, interpretar e gerar linguagem [32]. Esses modelos ocupam o núcleo da área de Processamento de Linguagem Natural (NLP), permitindo que computadores realizem tarefas como tradução automática, resumo de textos, reconhecimento de fala, e até mesmo a geração de textos coerentes e contextuais como respostas a perguntas. Eles funcionam ao analisar grandes quantidades de dados textuais para aprender padrões linguísticos, estruturas gramaticais e o uso de palavras em diversos contextos.

As primeiras abordagens de NLP eram baseadas em regras gramaticais e listas de palavras, mas mostraram-se limitadas devido à complexidade e variabilidade da linguagem humana. Com o desenvolvimento do *deep learning*, os modelos de linguagem alcançaram um novo patamar, principalmente após o desenvolvimento da arquitetura Transformer, capaz de entender o contexto de palavras em sentenças muito longas.

Embora as sequências de proteínas não sejam construções da linguagem humana, elas ainda exibem certas características que podem ser processadas por esses modelos. Por exemplo, uma sequência de proteínas é, primordialmente, um arranjo sequencial de um vocabulário de aminoácidos. Além disso, certos aminoácidos têm maior probabilidade de aparecer juntos do que outros, assim como certas palavras têm maior probabilidade de aparecerem juntas em uma frase. Na realidade, existem diversos critérios de viabilidade para aplicação de modelos de linguagem que conectam esses dois domínios do conhecimento. Trazendo alguns elementos de trabalhos recentes na fronteira da modelagem biológica linguisticamente inspirada [33, 34], podemos delimitar algumas propriedades interessantes para a hipótese de tratar o processamento de sequência primárias de proteínas como um problema de linguagem. Alguns exemplos são:

- Linguagens são sistemas de sinais simbólicos, discretos e complexos, onde cada sinal pode ser identificado com alta confiabilidade;
- Os elementos simbólicos principais e inseparáveis das linguagens são as palavras (morfemas), e eles podem ser codificados ou representados de diferentes formas para comunicar seu significado;
- Linguagens diferentes são estruturadas de formas distintas por regras gramaticais específicas.
- Algumas palavras em diferentes linguagens possuem múltiplos significados (polissêmicas),

podendo esses significados, às vezes, serem de naturezas opostas (auto-antônimos). Há conjuntos de palavras que têm o mesmo significado (sinônimos);

- Algumas palavras se comportam de forma diferente quando são utilizadas em contextos diferentes ou com funções diferentes, principalmente no que diz respeito ao seu contexto e posição na frase. Além disso, existem combinações idiomáticas de palavras que são caracterizadas por um significado diferente daquele das palavras isoladas;
- De forma geral, o significado de um conjunto de palavras pode conter ambiguidades lexicais (relativas a uma mesma palavra), sintáticas (relativas à função de cada palavra), semânticas (relativas a um conjunto de palavras) ou anafóricas (relativas a palavras mencionadas anteriormente).

Se entendermos os L- $\alpha$ -aminoácidos como palavras ou morfemas, as sequências de proteínas ou seus domínios como conjuntos dessas palavras, e os significados como estruturas funcionais em meio fisiológico, constitui-se uma massa crítica de hipóteses que sugerem o uso de paradigmas de NLP para construir modelos de proteômica estrutural.

#### 1.4.6 Breves considerações de ordem Filosófica

Embora possa parecer inusitado à primeira vista, utilizar técnicas de NLP para explorar estruturas de proteínas pode ser uma abordagem promissora, apesar de requerer uma análise cuidadosa da sua fundamentação teórica, e uma reflexão de “*por que*” esse uso faz sentido. Por exemplo, houve situações em que o campo do NLP evocou conceitos como a teoria da Gramática Universal de Chomsky [35, 36] (*um conceito por si só complexo, controverso e que o autor não pretende explicar nem tampouco posicionar-se sobre neste trabalho*). Nesse caso, *de maneira simplista* pressupondo que os seres humanos têm uma predisposição inata para a linguagem, uma base cognitiva universal para sua construção, isso implicaria a existência de padrões universais subjacentes na forma como a linguagem é percebida e processada, explicando o que a IA estaria capturando no aprendizado [37].

Decerto que parece bonito e poético que estejamos tentando então desvendar “*a verdadeira gramática com a qual a natureza redige as sequências de proteínas*”. De forma prática, no entanto, ao considerar sequências de proteínas, é claro que elas não são produtos da cognição

humana, mas sim determinadas por leis naturais e princípios da biologia molecular. Surge então a questão: *como é que modelos de linguagem, concebidos para simular e compreender as nuances da linguagem humana, podem ser relevantes para sequências naturais como as proteínas?*

A chave para esta questão reside na própria natureza dos modelos de Inteligência Artificial. Antes de mais nada, precisamos estabelecer que modelos de IA nada mais são do que ferramentas sofisticadas de fazer cálculos. Modelos como os Seq2Seq, por exemplo, realizam operações matriciais para converter sequências de entrada em sequências de saída. Essencialmente, eles não “compreendem” as sequências que processam, seja em linguagem natural ou em cadeias de aminoácidos, mas sim identificam padrões nas representações dessas sequências em bits e bytes.

Quando uma sequência de proteína é convertida em uma série de vetores numéricos, cada um representando um aminoácido específico, um modelo de NLP pode processar esses dados usando os mesmos princípios de álgebra linear aplicados a frases codificadas de forma semelhante. Essa habilidade de identificar padrões não se baseia em uma compreensão cognitiva universal, mas sim no reconhecimento de estruturas matemáticas ou estatísticas nos dados.

Portanto, a aplicação de modelos de NLP a sequências de proteínas não depende de uma base cognitiva compartilhada entre linguagem e biologia, mas na capacidade desses modelos de detectar e aprender padrões estatísticos. Isso destaca a versatilidade e a capacidade de abstração dos modelos de linguagem, permitindo que sejam empregados em áreas tão distintas quanto a linguística e a biologia molecular. O uso dessas ferramentas para analisar sequências de proteínas exemplifica como técnicas desenvolvidas para um campo podem oferecer *insights* valiosos em outro, através da adaptação de problemas similares reduzidos a seus princípios fundamentais.

## 1.5 Ensinar uma máquina não é barato!

Fica claro que, a partir dos modelos de linguagem, a inteligência artificial inaugura uma nova era na modelagem biomolecular [38]. Nesse momento, computadores já são capazes de produzir estruturas terciárias de alta qualidade, mesmo quando uma proteína-alvo exibe um baixo grau de homologia remota. Simultaneamente, alguns grupos de pesquisa estão ensaiando os primeiros passos na direção de processar apenas a sequência de aminoácidos idiomaticamente, em uma tentativa - de resultados ainda modestos - de traduzir sequências em estruturas sem utilizar nenhuma consulta a bases de dados de homologia, como é o caso do software RGN2 [39].

Desde que os recursos necessários estejam disponíveis, todo o código-fonte seja aberto e os conjuntos de treinamento sejam acessíveis, algoritmos inspirados no AlphaFold2, como o RoseTTAFold [40], podem a princípio ser desenvolvidos e implementados de forma local - mesmo que com um número reduzido de parâmetros treináveis - por grupos de pesquisa em todo o mundo. Atender aos requisitos dessa implementação, no entanto, não é uma tarefa simples, uma vez que o poder computacional bruto mínimo necessário pode ocupar uma parcela significativa dos ecossistemas de computação de alto desempenho. De fato, avançar na vanguarda dos modelos de estrutura de proteínas de ponta a ponta (“*end-to-end*”, ou da sequência à estrutura terciária) naturalmente vem com um grande crescimento no custo dos recursos de computação [41] e, apesar da disponibilidade de dispositivos GPU e TPU gratuitos em serviços como o Google Colab, muitos grupos ao redor do mundo são efetivamente excluídos da participação nesse avanço.

### 1.5.1 Existe modelagem além do *end-to-end*?

O problema da modelagem biomolecular, no entanto, não precisa ser resolvido integralmente de ponta a ponta por todos os algoritmos para gerar progresso científico. Como um problema de múltiplas escalas, naturalmente surgem como alvo de inferência as propriedades químicas e biológicas das proteínas e dos próprios aminoácidos que as formam. Nesse ínterim, surge uma categoria secundária de modelos de previsão de estrutura de proteínas que, embora recentemente ofuscados por seus irmãos *end-to-end*, permanecem disponíveis para implementação em ambientes de recursos limitados. Trata-se dos chamados estimadores de *propriedades estruturais* das proteínas, que preveem características de cada um dos resíduos de uma sequência refletidas na estrutura do estado nativo, como a estrutura secundária, acessibilidade ao solvente, desordem intrínseca, ângulos torcionais contínuos e número de contatos, entre outras propriedades.

Ainda que menos informativas por definição, as previsões geradas por esses estimadores de propriedades, quando suficientemente precisas, podem fornecer subterfúgio para definir a classe estrutural [42] de um alvo de modelagem, e as mesmas previsões podem ser usadas como estados de referência para avaliação e avaliação da qualidade de um modelo de proteína [43]. Eles também são usados como geradores do estado inicial para simulações de dinâmica molecular (MD) e modelagem baseada em estrutura (SBM) [44, 45]. Mesmo no caso em que uma pequena quantidade de elementos de estrutura secundária é prevista, como é o caso de proteínas

intrinsecamente desordenadas (IDPs), essas previsões podem orientar a busca de sítios funcionais nas respectivas regiões não estruturadas [46] ou até indicam transições entre estados enovelados (do inglês, *fold changes*) [47].

Quando os estimadores de propriedade evoluíram em um certo ponto para prever valores contínuos para propriedades de resíduos, como ângulos de torção, eles também se tornaram úteis para construir potenciais personalizados para cada alvo de modelagem com base na distribuição de von Mises, uma abordagem usada pela iteração original do AlphaFold. Diante dessa diversidade de propriedades estruturais, escolhemos concentrar este trabalho em três propriedades que além de serem muito prevalentes na literatura, costumam ser estimadas em conjunto pelos modelos: estrutura secundária categórica em três estados, ângulos ( $\varphi, \psi$ ) do *backbone* proteico e área superficial acessível ao solvente. A Figura 1.4 traz um sumário visual dessas propriedades.

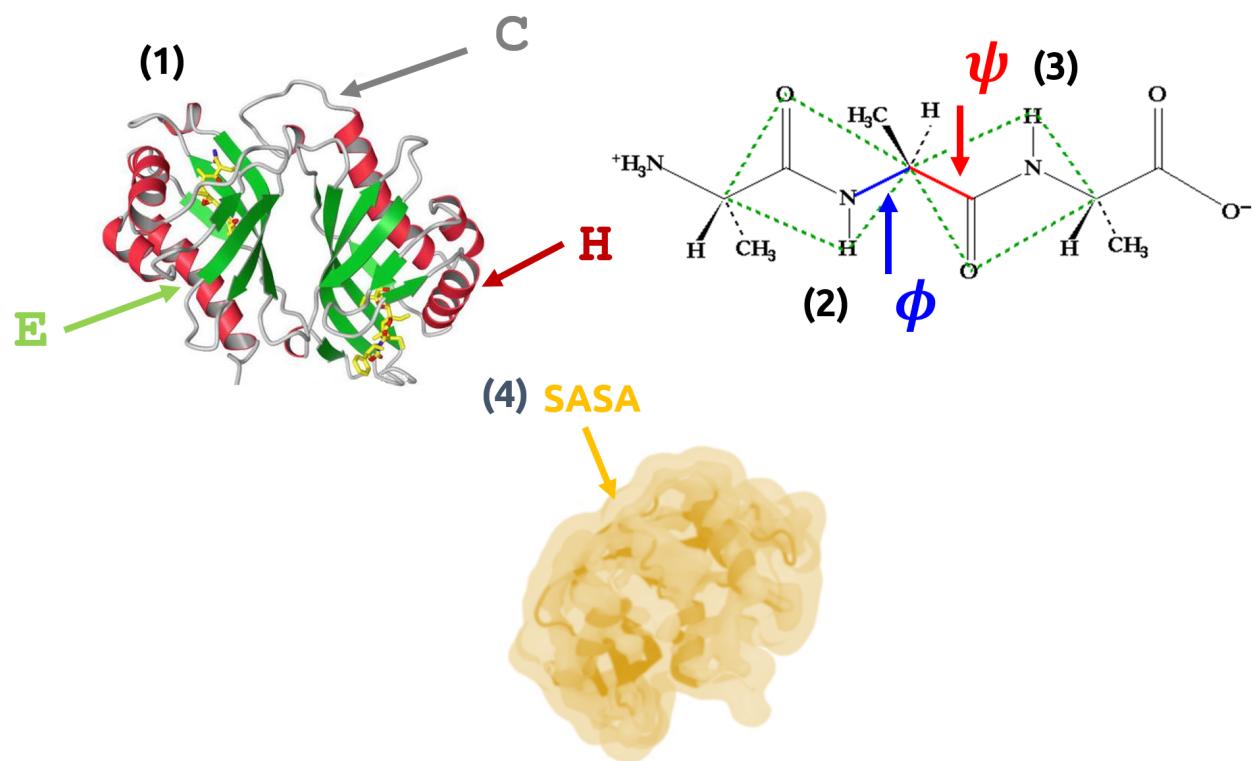


Figura 1.4: Detalhamento das propriedades estruturais citadas que foram abordadas neste trabalho. (1) Estrutura secundária categórica SS3. (2, 3) Angulos ( $\varphi, \psi$ ). (4) Área superficial acessível ao solvente (SASA).

Os estimadores de propriedades unidimensionais, como muitos outros algoritmos de determinação de estrutura de proteínas, experimentaram uma progressão histórica da qualidade

das previsões geradas que sofreu um incremento muito agudo após o surgimento de algoritmos baseados em *Deep Learning*, e gradualmente se aproximaram de seu limite teórico de precisão de classificação [48]. Por exemplo, a determinação da estrutura secundária de proteínas começou com métodos heurísticos, onde um dos pioneiros foi o método de Chou e Fasman [49, 50], desenvolvido na década de 1970. Este método baseava-se na propensão de cada aminoácido em formar estruturas secundárias como hélices alfa e folhas beta, usando dados empíricos extraídos de estruturas conhecidas. Os cálculos envolviam médias móveis e contagem de resíduos, mas a precisão era limitada pela dependência em dados limitados e a simplificação dos contextos estruturais. Seguindo a linha de avanços heurísticos, o método GOR (Garnier-Osguthorpe-Robson) [51, 52] foi introduzido em 1978. Ele calculava as probabilidades baseando-se não em aminoácidos individuais, mas em janelas deslizantes de fragmentos de sequências para prever a estrutura secundária, como uma forma de levar em consideração o ambiente local dos resíduos. Posteriormente, métodos como PSIPRED e JPRED representaram uma evolução significativa.

De forma similar aos modelos para estruturas terciárias, em determinado momento os preditores de propriedades estruturais mais competitivos começaram a incorporar MSAs ou resultados de análise de coevolução sobre esses alinhamentos. Embora ainda fossem anteriores ao uso intenso de *deep learning*, PSIPRED[28] utilizava redes neurais simples treinadas em estatísticas sobre MSAs e bancos de dados de estruturas conhecidas, melhorando a precisão das previsões. JPRED[29], por sua vez, combinava o uso de MSAs com as janelas deslizantes de fragmentos de aminoácidos, e combinava essas previsões com outros métodos heurísticos por meio de um abordagem de *ensemble*. Esses métodos pavimentaram o caminho para os avanços subsequentes na bioinformática estrutural, estabelecendo as bases para a era de ouro do *machine learning*. A vantagem dessa abordagem foi que a informação evolutiva melhorou fortemente a precisão da previsão, assim como aconteceu com os modelos geradores de estruturas terciárias. No entanto, rapidamente percebeu-se uma deterioração severa na precisão desses algoritmos baseados em informação evolutiva, mais proeminente nos casos de MSAs rasos ou proteínas sem homologia identificável, que são a regra nos alvos de modelagem contemporâneos [53].

Esse fato, juntamente com o entendimento de que, do ponto de vista fisiológico, as proteínas se dobram em suas estruturas secundárias apenas por determinação da sequência primária [54], promoveu o desenvolvimento de abordagens de processamento de sequência única.

Os representantes mais notáveis e recentes desta categoria são SPIDER3-single [55], ProteinUnet [56] e SPOT1D-single [57]. Essas arquiteturas de redes neurais foram finalmente elaboradas com os mesmos modelos de linguagem que figuram nos modelos *end-to-end*, e foram crescendo em precisão e complexidade conforme a série temporal do seu desenvolvimento.

### 1.5.2 Contra o fluxo

Essa tendência, embora utilitariamente positiva, gera um problema intrínseco do processo de melhoramento dos algoritmos: conforme eles vão ficando cada vez mais performáticos, mais parâmetros treináveis e mais operações complexas precisam ser realizadas para alcançar incrementos cada vez mais marginais de desempenho. Esse fenômeno culmina no desenvolvimento de efetivas “caixas-pretas”, que realizam operações essencialmente incompreensíveis, suprimindo qualquer esforço de utilizar o algoritmo treinado para elucidar os padrões que estão sendo abstraídos. Observando esse fenômeno, encontramos motivação para empreender no sentido contrário desse crescimento desenfreado de complexidade.

**Este é, portanto, um trabalho rebelde.** Não se trata de um esforço para desenvolver o próximo modelo que vai ganhar 1% de desempenho dobrando o número de parâmetros. Nem faz sentido fazê-lo, quando já é possível fazer o download e executar um dos diversos modelos pré-treinados derivados do AlphaFold2, utilizando-o simplesmente no modo de inferência. Este trabalho não tem como único objetivo apenas acertar mais que seus predecessores (embora seja importante, aqui, não errar demais), mas interrogar cientificamente - molecularmente - por meio de arquiteturas algébricas específicas para este fim, a maneira com que a sequência de aminoácidos é numericamente manipulada e transformada em um conjunto de números que, juntos, traduzem a estrutura de uma proteína natural.

## 1.6 Uma pequena brincadeira com ChatGPT

Autor: “ChatGPT, can you give me examples for applications of seq2seq models that are outside the linguistic domain? do they exist?”

Reposta: “Seq2seq models, or sequence-to-sequence models, have indeed been applied outside the linguistic domain. Although originally designed for tasks such as machine translation, they've proven to be versatile and useful in a variety of different fields. Here

are a few examples:

Time Series Forecasting: Seq2seq models can be used for predicting future values of a time series, given its history. This is relevant in numerous fields such as finance (predicting stock prices), meteorology (forecasting weather), and many more.

**Bioinformatics:** In this domain, seq2seq models can be used for tasks such as predicting protein structures based on their amino acid sequences.

Music Generation: Seq2seq models can be trained to generate music. The input can be a sequence of notes or chords, and the output is the continuation of the music piece.

Image Captioning: Though primarily a linguistic task, the seq2seq model here needs to interpret image data (visual sequence) and generate a corresponding textual description (text sequence).

Autonomous Driving: Seq2seq models can be used to predict the sequence of future actions or maneuvers of a vehicle based on the sequence of past and current sensor inputs.

Video Processing and Analysis: In video event prediction, a seq2seq model can predict what will happen next in a video, given the current sequence of frames.

Medical Data Analysis: Seq2seq models can be used to predict future medical events from a sequence of past health-related events in electronic health records.

Speech Recognition and Synthesis: These are considered non-linguistic in that they involve sequences of audio data. A seq2seq model can transcribe speech (audio-to-text) or generate speech from text (text-to-speech).

Remember that sequence-to-sequence models are all about mapping one sequence to another, and it doesn't necessarily have to involve text. It can be any kind of sequence, as long as there's some temporal or ordered aspect to the data."

ChatGPT v3.5. Seq2Seq Beyond Linguistics. Personal communication with Guilherme Fahur Bottino, 21 de Junho de 2023. Disponível em: <https://chat.openai.com/share/d45c3232-ba1f-4b1f-a730-ebf498a9ce08>. Acesso em: 21 de Junho de 2023.

## Capítulo 2

# Objetivo

O motivo estrutural de uma proteína enovelada é caracterizado por fragmentos de estrutura secundária e suas posições relativas coalescidas por interações de médio e longo alcance na sequência primária, e essa estrutura secundária é determinada pela sequência de aminoácidos, que apresenta características que viabilizam seu processamento por meio de algoritmos de NLP.

Considerando o exposto, o **Objetivo Principal** deste trabalho é desenvolver um modelo de linguagem baseado em *Machine Learning*, empregando redes neurais, em código aberto, capaz de receber como única entrada a sequência primária da proteína-alvo em questão e prever as seguintes propriedades estruturais: estrutura secundária categórica em três estados (SS3); ângulos torcionais ( $\varphi, \psi$ ) do *backbone* proteico; e área superficial acessível ao solvente (SASA) de cada resíduo na estrutura enovelada.

Considerando também que arquiteturas de redes neurais comumente utilizadas em modelos de linguagem contemporâneos, como convoluções escaláveis e Transformers, podem ter seus pesos internos extraídos para iluminar o percurso dos dados de ponta a ponta do algoritmo, o **Objetivo Secundário** deste trabalho é investigar a emergência de padrões sequenciais e estruturais relacionados, por meio de visualizações desses pesos extraídos em estudos de caso.

## Capítulo 3

# Abordagem

Nossa abordagem do problema de modelagem começa com uma motivação em construir uma arquitetura de *Machine Learning* mais enxuta, que surge como ideia durante a pandemia de COVID-19. Essa informação parece irrelevante, mas ela tem uma importância ideológica tremenda na concepção do projeto, pelo menos do lado do autor. Durante o período mais crítico de isolamento da pandemia, surgiu uma ideia de criar um modelo de IA que, ao mesmo tempo que alcançasse um desempenho digno da fronteira de conhecimento da estimativa de propriedades 1D, também fosse simples o suficiente a ponto de permitir que qualquer pessoa submetida ao isolamento pudesse estudar, programar, desenvolver, treinar, interpretar e executar utilizando hardware comumente disponível para o mercado de consumo. Em outras palavras, um modelo que seria executável em computadores domésticos, sem requisitar hardware especializado de grandes sistemas de HPC.

Essa abordagem não é inédita no campo da modelagem de proteínas. No passado, lançando mão do fato de que modelos menores funcionam em uma variedade maior de dispositivos e podem ser implementados mesmo em ambientes amplamente virtualizados, e aliando essa propriedade à ideia de computação distribuída, extensos experimentos de modelagem distribuída de proteínas foram realizados, a exemplo de iniciativas como FoldIt [58] e Folding@Home [59].

Por trás dessa ideia, talvez até mais importante, vem um ideal de que um modelo pequeno e competitivo amplia a **acessibilidade** do estado-da-arte de previsão de propriedades 1D a grupos de pesquisa que foram marginalizados da pesquisa contemporânea em IA por conta da pouca disponibilidade de recursos computacionais, contribuindo para **democratizar** o acesso a essa linha de pesquisa e integrando mais cientistas no esforço global do seu progresso.

### 3.1 Um modelo unidimensional, acessível em três dimensões

A palavra-chave **acessibilidade** permeia completamente o desenvolvimento deste projeto.

Evidentemente, conforme a narrativa anterior, a primeira dimensão de acessibilidade aqui postulada é a dimensão de custo, que no universo dos algoritmos de IA, é paralela ou colinear com a dimensão do próprio poder computacional alocado e investido. Posicionamos este trabalho no sentido que o permite ser executado com menos recursos e ser treinado em menos tempo.

A segunda dimensão de acessibilidade é aquela da reprodutibilidade. O trabalho foi concebido desde o início para ser compartilhado publicamente, com código e *dados* amplamente disponíveis em formato aberto.

A terceira e última dimensão de acessibilidade, que retoma a última ideia da Introdução, é aquela da transparência em relação aos mecanismos internos do modelo, onde em um extremo existem as “caixas-pretas” completamente impermeáveis, e no outro extremo posicionamos este trabalho, onde um dos objetivos principais é justamente acessar os mecanismos internos do modelo e procurar entender de que forma, em que proporção, e em que momento, propriedades estruturais estão sendo abstraídas da sequência primária.

### 3.2 Revisão bibliográfica

Fora do domínio da modelagem biomolecular, a redução do tamanho de modelos de ML para viabilizar implementações “*mobile, on-device*” [60] já é uma prática em amplo desenvolvimento. Artigos de revisão desses outros domínios de conhecimentos mostram que o processo de simplificação de modelos iniciou historicamente com uma simples poda de parâmetros e camadas em modelos já treinados, evoluindo para tornar-se uma filosofia de uso de operações e arquiteturas que são naturalmente econômicas e têm um alto potencial de escalabilidade [61].

Pensando nisso, nossa abordagem para este trabalho foi justamente trazer essa mesma filosofia e estudar profundamente as implementações já existentes e avançar sobre elas por meio da substituição de seus componentes por paradigmas mais atualizados, de modo que a redução no tamanho dos modelos pudesse ser compensada pelo aumento da utilidade dos parâmetros coletivos. Os modelos escolhidos para esse estudo foram os já mencionados modelos recentes de estimadores

de propriedades de sequência única (SPIDER3-single, ProteinUnet e SPOT1D-single).

SPIDER3-single é o mais antigo, e faz uso da teoria mais clássica do aprendizado sequencial com redes neurais recorrentes (RNNs), consistindo em uma sucessão de camadas bidirecionais de Long Short-Term Memory (LSTM) [62] e camadas lineares totalmente conectadas, como em sua versão baseada em MSA, SPIDER3 [63]. Aqui, empregou-se um estado anterior de conhecimento em modelagem de linguagem, utilizando células recorrentes para acompanhar o contexto - passado e futuro, graças à sua bidirecionalidade - para um determinado componente de uma sentença [64, 65]. A principal desvantagem das arquiteturas recorrentes é o fato de que uma sequência deve ser processada em sua ordem específica - direta ou reversa -, o que gera um grau inevitável de serialização que escala com o comprimento da sequência.

Conforme já exposto, uma grande melhoria em relação às RNNs foi introduzida pela arquitetura Transformer, baseada nas chamadas operações de auto atenção (do inglês, “*self-attention*”), trivialmente paralelizáveis, realizadas sobre sentenças codificadas por posição. Essa arquitetura alcançou grandes avanços em velocidade e previsibilidade em tarefas baseadas em sequência, tornando-se rapidamente o estado da arte contemporâneo em modelos de linguagem [66], e chegando às aplicações mais modernas em modelagem de proteínas, como as camadas EvoFormer do AlphaFold2 ou o AminoBERT do RGN2. Baseado nisso, decidimos que alguma forma de codificação baseada em Transformer poderia ser um bloco de construção fundamental do nosso trabalho e poderia melhorar a ideia de modelagem seq2seq.

O ProteinUnet, segundo modelo do qual nos inspiramos, empregou uma rede onde blocos simétricos convolucionais e deconvolucionais - uma arquitetura UNet [67] - operando sobre janelas de aminoácidos foram usados para gerar estimativas. As operações de convolução para análise de sentenças já eram uma realidade estabelecida na época da publicação [68], e a ideia parece natural porque o *kernel* de convolução é capaz de representar e aprender relações espaciais na entrada, permitindo que propriedades sequenciais (como a ordem dos resíduos) sejam contabilizadas no modelo. Blocos consecutivos permitem que a rede aumente seu campo receptivo [69], o que se traduz na consideração de um grande fragmento peptídico para a estimativa da estrutura em um único aminoácido.

Uma das desvantagens dessa abordagem é que, com um campo receptivo grande o suficiente, algumas das relações de longo alcance acabam sendo espúrias, prejudicando o desempenho

- e a utilidade efetiva - dos parâmetros. Em nosso trabalho, resgatamos da ProteinUnet a ideia de utilizar camadas convolucionais como um componente adicional do nosso bloco codificador, implementando convoluções mais rasas e mais largas e priorizando reduzir o custo de memória.

O último modelo que usamos como referência para este trabalho foi o SPOT1D-single. Foi construído em torno da ideia de agrupar (“*ensemble*”) previsões de três redes neurais que já eram de última geração, especificamente uma rede recorrente, uma rede convolucional profunda e uma RNN-CNN final mista. A ideia de prever com um conjunto de resultados de vários estimadores é justificada por um aumento na precisão das potencialidades de amostragem dos componentes individuais descorrelacionados e é a base da versão baseada em MSA do Spot1D [70].

As técnicas de ensemble, no entanto, geralmente apresentam a desvantagem de que todos os modelos subjacentes devem ser implementados e executados antes da etapa final - a estimativa real da propriedade. A consequência é que, embora o SPOT1D-single seja um dos modelos contemporâneos de estimativa de propriedades com melhor desempenho, o conjunto de parâmetros para os três modelos, combinados com a cabeça de detecção final, somam mais de 80 milhões de parâmetros, que representam quase duas ordens de grandeza a mais do que o tamanho do SPIDER3-single e do ProteinUnet. Em nosso trabalho, o SPOT1D-single não foi uma inspiração arquitetônica (uma vez que ele já é, por si só, um ensemble de outros modelos), mas nos inspirou a encontrar uma maneira mais inteligente de unir potencialidades dos outros modelos estudados, além de oferecer uma referência para o nível de precisão na previsão que queríamos alcançar como alvo.

Essa inspiração em combinar convoluções e Transformers de forma inteligente culminou no componente final de nossa abordagem: o fluxo real de informações através da rede, para o qual nos inspiramos na arquitetura DETR [71], onde convoluções, codificadores e decodificadores de transformadores e cabeças de previsão foram usados sucessivamente, não de forma descorrelacionada em um ensemble de redes independentes, mas como diferentes etapas de processamento de dados em um só estimador. O produto dessa abordagem, nossa arquitetura final, é descrito em detalhes na seção 4.2

## Capítulo 4

# Metodologia Desenvolvida

### 4.1 Obtenção e pré-processamento dos dados

O primeiro passo do desenvolvimento de um modelo de *Machine Learning* para estimar propriedades estruturais a partir de sequências de proteínas é obter um conjunto de pares sequência-estrutura de referência, que será utilizado para treinar o modelo. Naturalmente, para esse fim, é necessário trabalhar com as proteínas cuja estrutura tridimensional já foi investigada e determinada por alguma técnica de caracterização estrutural capaz de fornecer resultados com resolução atômica. Conforme já mencionado, a técnica mais comum para esse fim é a **cristalografia** empregando difração de raios-X em monocrystalais de proteína. Essa técnica responde pelo maior número de estruturas depositadas em bases de dados, e fornece as melhores resoluções, sendo também a mais fácil de processar, então evidentemente é a escolha natural aqui.

#### 4.1.1 Seleção da fonte de dados e obtenção de sequências e estruturas

Existem alguns bancos de dados contendo estruturas cristalográficas de proteínas, e muitos subconjuntos e seleções desses bancos que servem a fins diferentes, e coube a nós escolher o conjunto adequado para este projeto. Por princípio, o conjunto de dados de referência deve refletir a distribuição do espaço latente sequencial e estrutural compartilhado por toda a população de possíveis proteínas. Esse requisito sempre foi considerado um grande obstáculo à **generalizabilidade** (ou seja, capacidade de gerar previsões para proteínas que não foram vistas durante a etapa de treinamento) e **perplexidade** (ou seja, capacidade de gerar previsões plausíveis do ponto de vista sequencial ao encontrar uma sequência “surpreendente”, sem padrões inapropriados - ex: CHEHEHEC) dos modelos de *Machine Learning* aplicados a estrutura de proteínas. A razão disso já mencionada,

é que o número de proteínas já caracterizadas é muito pequeno frente ao volume de sequências de proteínas. Como consequência - e também agravante -, o espaço estrutural já acessado é fortemente enviesado às vizinhanças dos representantes mais fáceis de cristalizar e caracterizar, e seus homólogos.

Portanto, a escolha de uma base de dados *representativa* foi um passo fundamental do projeto. Selecioneamos o banco de dados CATHS40 [72] como a principal fonte de dados para nosso trabalho. O CATHS40 armazena sequências e estruturas de **domínios** (fragmentos estruturalmente independentes de uma mesma sequência) de proteínas **não-redundantes**. Ele contém mais de 30.000 domínios de proteína segregados do Protein Data Bank [73], filtrados de modo que não mais de 40% de identidade de sequência seja exibida por quaisquer duas entradas.

As estruturas foram obtidas em formato mmCIF (*macromolecular Crystallographic Information File*), que desde 2018 substituiu como padrão o antigo formato PDB, originalmente criado para padronização do Protein Data Bank. O formato mmCIF emprega uma notação de objeto hierárquica, na qual é possível obter coordenadas espaciais (x, y e z) para cada átomo resolvido da estrutura da proteína, conforme exemplificado no Código 1.

#### 4.1.2 Extração das estruturas secundárias e demais propriedades unidimensionais

O software DSSP (*Dictionary of Protein Secondary Structure*) [74, 75] é o software mais comumente utilizado para calcular estruturas secundárias a partir de estruturas PDB. Nesse trabalho, ele foi utilizado no conjunto de coordenadas de cada domínio para obter estruturas secundárias, acessibilidades de solventes e ângulos  $\varphi$  e  $\psi$  de cada resíduo. Esse software *não* é um classificador preditivo baseado em sequências, uma vez que exige um conjunto já existente de coordenadas tridimensionais com resolução atômica para funcionar. Seu funcionamento é meramente de calcular, dadas as posições de cada átomo dos aminoácidos, um conjunto de ângulos entre resíduos e ligações de hidrogênio realizadas por cada resíduo e, a partir desses dados, classificar cada um deles segundo critérios predefinidos, extensivamente formalizados (inclusive do ponto de vista geométrico) na publicação original de Sander e Kabsch.

**Código 1.** Exemplo de coordenadas atômicas em um arquivo mmCIF. Cada linha corresponde a um átomo de cada aminoácido (os átomos de hidrogênio geralmente não são exibidos). Nas colunas centrais, observam-se

coordenadas cartesianas para os três primeiros aminoácidos da proteína de exemplo (PDB-5CXO-B).

ATOM	1	N N	. MET A	1 1	?	-5.392	-4.777	-7.145	1.00	53.69	?	1	MET A N	1
ATOM	2	C CA	. MET A	1 1	?	-5.472	-6.236	-6.858	1.00	53.50	?	1	MET A CA	1
ATOM	3	C C	. MET A	1 1	?	-4.094	-6.793	-6.690	1.00	61.29	?	1	MET A C	1
ATOM	4	O O	. MET A	1 1	?	-3.124	-6.181	-7.129	1.00	65.90	?	1	MET A O	1
ATOM	5	C CB	. MET A	1 1	?	-6.122	-6.940	-8.007	1.00	59.08	?	1	MET A CB	1
ATOM	6	C CG	. MET A	1 1	?	-7.428	-6.329	-8.360	1.00	58.39	?	1	MET A CG	1
ATOM	7	S SD	. MET A	1 1	?	-8.779	-7.357	-7.765	1.00	83.25	?	1	MET A SD	1
ATOM	8	C CE	. MET A	1 1	?	-8.460	-7.571	-6.021	1.00	80.32	?	1	MET A CE	1
ATOM	9	N N	. GLN A	1 2	?	-3.986	-7.965	-6.073	1.00	50.37	?	2	GLN A N	1
ATOM	10	C CA	. GLN A	1 2	?	-2.666	-8.567	-5.945	1.00	51.81	?	2	GLN A CA	1
ATOM	11	C C	. GLN A	1 2	?	-2.354	-9.590	-7.023	1.00	46.39	?	2	GLN A C	1
ATOM	12	O O	. GLN A	1 2	?	-1.207	-9.910	-7.248	1.00	52.61	?	2	GLN A O	1
ATOM	13	C CB	. GLN A	1 2	?	-2.472	-9.149	-4.552	1.00	61.50	?	2	GLN A CB	1
ATOM	14	C CG	. GLN A	1 2	?	-2.047	-8.082	-3.544	1.00	67.84	?	2	GLN A CG	1
ATOM	15	C CD	. GLN A	1 2	?	-2.313	-8.482	-2.111	1.00	76.33	?	2	GLN A CD	1
ATOM	16	O OE1	. GLN A	1 2	?	-3.448	-8.821	-1.744	1.00	72.16	?	2	GLN A OE1	1
ATOM	17	N NE2	. GLN A	1 2	?	-1.271	-8.423	-1.285	1.00	75.92	?	2	GLN A NE2	1
ATOM	18	N N	. ASP A	1 3	?	-3.399	-10.127	-7.634	1.00	43.79	?	3	ASP A N	1
ATOM	19	C CA	. ASP A	1 3	?	-3.273	-10.905	-8.826	1.00	41.90	?	3	ASP A CA	1
ATOM	20	C C	. ASP A	1 3	?	-2.835	-9.935	-9.984	1.00	40.34	?	3	ASP A C	1
ATOM	21	O O	. ASP A	1 3	?	-3.541	-9.017	-10.298	1.00	33.64	?	3	ASP A O	1
ATOM	22	C CB	. ASP A	1 3	?	-4.592	-11.491	-9.218	1.00	43.13	?	3	ASP A CB	1
ATOM	23	C CG	. ASP A	1 3	?	-4.465	-12.282	-10.446	1.00	55.21	?	3	ASP A CG	1
ATOM	24	O OD1	. ASP A	1 3	?	-5.185	-11.993	-11.427	1.00	57.14	?	3	ASP A OD1	1
ATOM	25	O OD2	. ASP A	1 3	?	-3.578	-13.167	-10.437	1.00	51.22	?	3	ASP A OD2	1

Por padrão, o software DSSP é capaz de classificar a estrutura secundária entre 9 classes diferentes possíveis (SS9, uma atualização recente das antigas 8 classes, SS8), mas o problema clássico de classificação da estrutura secundária apenas diferencia entre as 3 classes principais (SS3: hélices, fitas e segmentos sem estrutura secundária definida). Portanto, construímos um mapeamento entre a saída do DSSP (SS8/SS9) e a saída desejada pelo problema que estávamos atacando (SS3), da seguinte forma:

- S, T, C e regiões sem atribuição são mapeadas para C
- G, H, P e I (estados de hélice) para H
- B e E (estados de fita) para E.

Apos processamento com o software DSSP, realizamos uma análise confirmatória com o software STRIDE [76], outro padrão-ouro para extração de propriedades estruturais a partir de estruturas PDB. Nosso objetivo era remover da análise qualquer estrutura em que não houvesse concordância em mais de 90% das atribuições bem-sucedidas, pois nesse caso poderia haver algum tipo de ambiguidade que dificultaria o trabalho do modelo. Contudo, não houve casos em que isso ocorreu.

### 4.1.3 Construção da base de dados em formato universal de acesso aberto

Todas essas saídas são combinadas com o domínio original FASTA para produzir pares de entrada-saída de cada aminoácido em uma sequência. Devido à descontinuidade nas coordenadas do *backbone* exibidas em uma grande parte das amostras, a sequência FASTA original é alinhada rapidamente à sequência que de fato está no arquivo PDB e que não falhou no DSSP usando a biblioteca BioAlignments.jl [77] e um algoritmo de *affine gap scoring*, e esse alinhamento é utilizado para garantir que as previsões correspondam às posições corretas, preenchendo as propriedades que faltam com o tradicional símbolo de lacuna ('-'). Finalmente, as amostras contendo menos de 32 aminoácidos são descartadas e as amostras maiores que 192 são cortadas para esse comprimento máximo. Isso foi necessário para permitir que o tamanho de um lote de amostras, combinado com o próprio modelo e seus gradientes, não ocupasse um tamanho superior ao que a GPU que tínhamos disponível poderia comportar (mais detalhes sobre recursos de computação na seção 4.5).

Toda a base de dados de treino construída para esse projeto foi organizada e armazenada em arquivos JSON, em formato aberto, e está disponível para download no repositório público do projeto (<https://github.com/Hugemiler/TintiNet.jl>)

## 4.2 Arquitetura e Implementação do modelo preditivo

### 4.2.1 O que significa “arquitetura” de uma rede neural?

A **arquitetura** de uma rede neural refere-se ao *design* e à estrutura da rede, que inclui vários aspectos fundamentais que determinam como a rede funciona e se comporta durante o treinamento e a inferência. A terminologia utilizada no domínio de conhecimento das redes neurais é extensa, e, embora não seja objetivo dessa seção (ou deste trabalho) explicar cada termo utilizado aqui, cada componente será acompanhado de alguns exemplos em terminologia usual:

- **Número de Camadas (Layers)**: redes neurais são construídas em camadas. Portanto, o componente mais básico da arquitetura é o número de camadas que a rede possui, incluindo camadas de entrada, camadas ocultas e camada de saída.
- **Tipos de Camadas (Layer Types)**: Os diferentes tipos de camadas utilizadas na rede,

como camadas densas (*fully connected*), convolucionais (CNN), recorrentes (RNN), camadas de normalização, camadas de *dropout*, camadas Transformer, etc.

- **Complexidade (quantidade de parâmetros) por Camada (Layer Complexity):** a forma de medir essa complexidade varia entre diferentes tipos de camadas. Em uma camada densa (completamente conectada), por exemplo, pode ser medida pelo número de neurônios (ou unidades) existentes na camada. Em uma camada convolucional, vai depender da dimensionalidade, do número de filtros e do tamanho desses filtros. Geralmente, sabendo o tipo de camada e a complexidade, é possível derivar diversos outros fatores importantes, como por exemplo o formato adequado dos dados antes e depois de passar pela camada. Em linhas gerais, uma camada mais complexa armazena mais parâmetros ajustáveis durante o aprendizado, e, portanto, isso impacta na quantidade de memória alocada para o modelo e, dentro do mesmo tipo de camada, na quantidade de operações que a camada realiza.
- **Funções de Ativação (Activation Functions):** As funções matemáticas aplicadas à saída de cada neurônio, como ReLU, sigmoid, tanh, softmax, entre outras, que introduzem não-linearidades na rede.

A maioria das bibliotecas de Machine Learning das várias linguagens de programação tem funções específicas para construir as camadas das redes neurais, e os argumentos passados para essas funções são responsáveis por ditar a complexidade da camada e a função de ativação. Por exemplo, na biblioteca Flux.jl utilizada neste trabalho, a linha de código

```
Conv((5,5), 8 => 4, σ="relu")
```

Declara uma camada convolucional com um filtro 2D 5x5, que recebe um tensor em que cada amostra possui 8 canais bidimensionais, e que produzirá um tensor em que cada amostra possui 4 canais bidimensionais, com uma função de ativação do tipo *relu* (do inglês, *rectified linear unit*, uma função não-linear que tem valor nulo para argumentos negativos e valor de identidade para argumentos não-negativos)

- **Esquema de Conectividade (Connectivity Scheme):** Como os neurônios em uma camada estão conectados aos neurônios na próxima camada. Por exemplo, em uma camada totalmente

conectada, cada neurônio está conectado a todos os neurônios da camada anterior.

- **Algoritmos de Otimização (*Optimization Algorithms*)**: Os diferentes algoritmos usados para ajustar os pesos da rede durante o treinamento de acordo com os gradientes da função objetivo, como descenso estocástico no gradiente (SGD), Adam, RMSprop, entre outros.
- **Função Objetivo ou Função Perda (*Objective Function ou Loss Function*)**: A função que mede o quanto bem a rede está se saindo durante o treinamento, que precisa ser minimizada. Exemplos incluem erro quadrático médio (MSE), entropia cruzada, etc.
- **Hiperparâmetros (*Hyperparameters*)**: Parâmetros globais que influenciam o processo de treinamento da rede como um todo, como a taxa de aprendizado, tamanho do lote de amostras (*batch size*), número de épocas (*epochs*), taxa de *dropout*, etc.
- **Regularização (*Regularization*)**: Métodos utilizados para evitar o sobreajuste da rede, como *dropout*, regularização L1/L2, data augmentation, entre outros.

A arquitetura de uma rede neural é crucial porque influencia diretamente a capacidade da rede de aprender padrões dos dados, sua eficiência computacional, e sua capacidade de generalizar para novos dados. Projetar uma boa arquitetura geralmente requer experimentação e um entendimento profundo tanto do problema quanto das técnicas de rede neural disponíveis.

#### 4.2.2 Concepção da Arquitetura TintiNet.jl

Projetamos duas redes neurais (uma de classificação, outra de regressão) que seguem, cada uma, a mesma estrutura geral de codificador-decodificador, utilizando uma combinação de convoluções e Transformers para processar cada sequência de aminoácidos categórica com comprimento  $L$ ,  $32 < L < 192$ . No caso da classificação, a saída é um vetor de comprimento  $L$  contendo, em cada posição, uma distribuição de probabilidade para classes de estrutura secundária SS3; no caso da regressão, a saída é um vetor de comprimento  $L$  contendo, em cada posição, um outro vetor de valores reais contínuos correspondentes às propriedades estimadas.

A fim de processar múltiplas sequências nesse paradigma, desenvolvemos uma arquitetura na qual a entrada da rede neural é uma matriz de sequências  $L_{\max} \times N_{\text{sequências}}$  que contém uma sequência por coluna e um aminoácido por linha, representados por valores inteiros que mapeiam para chaves de um vocabulário FASTA. Além dos 20 aminoácidos naturais, existem duas entradas

adicionais correspondentes a um elemento de preenchimento neutro “-” e o aminoácido desconhecido “X”, que representa um total de 22 entradas possíveis. A Figura 4.1 apresenta um sumário visual da arquitetura, que será descrita a seguir.

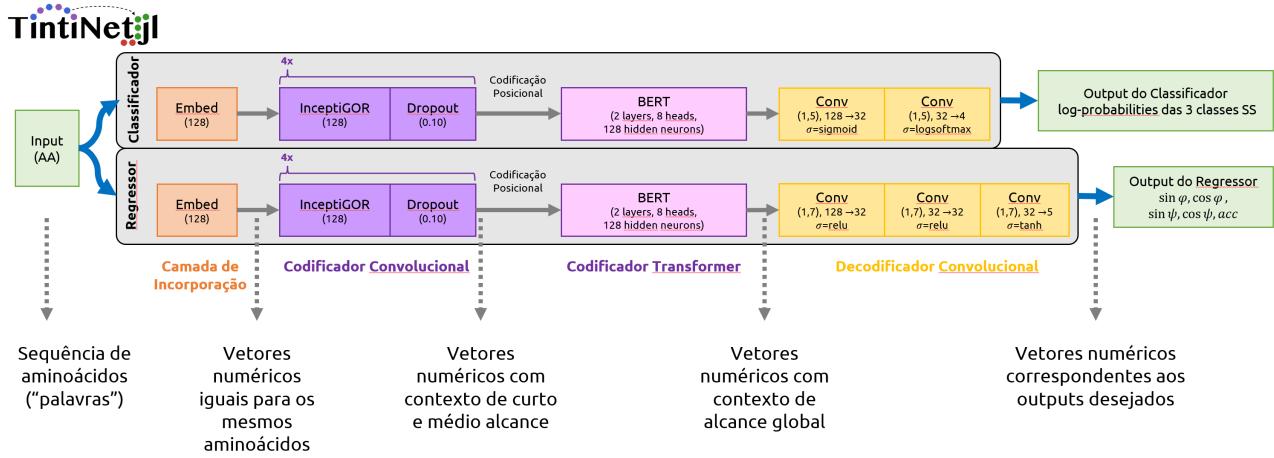


Figura 4.1: Sumário visual da arquitetura completa do TintiNet.jl, mostrando as redes Classificadora e Regressora em paralelo. Destaque para a estrutura da cabeça de previsão, que é diferente em ambas as redes. As setas pontilhadas mostram o significado aproximado do conteúdo do vetor codificado em cada posição da sequência de entrada no modelo

O primeiro elemento do bloco codificador, conforme é comum no processamento de linguagem natural, é uma camada de incorporação (*embedding*), responsável por codificar cada elemento de entrada com um vetor de comprimento  $d$  (por padrão neste projeto,  $d = 128$ ). Essas representações incorporadas são aprendidas e constituem o primeiro conjunto de parâmetros treináveis para o modelo. O valor  $d$ , que é a dimensionalidade da representação aprendida, é um hiperparâmetro de modelo e é conservado em toda a rede do codificador. Ele é o grande responsável por ditar o tamanho do modelo e a quantidade de relacionamentos que podem ser representados por uma única incorporação e aprendidos por cada camada. A saída da camada de incorporação é um tensor  $d \times L_{\max} \times N_{\text{sequências}}$ .

A camada de incorporação é seguida por um conjunto de quatro blocos convolucionais do tipo Inception [78, 79] modificados, que processam cada sequência por meio de uma série de convoluções 1D paralelas com tamanhos de janela variados. Essa ideia é parcialmente inovadora, e foi inspirada nos primeiros trabalhos de predição de estruturas secundárias que deram origem ao método GOR [51]. Desenvolvemos um bloco de operações composto por oito caminhos de processamento paralelos. Como no Inception V1 original, o primeiro caminho é uma convolução de elemento único com uma diminuição de dimensionalidade (no nosso caso, 8:1). Os próximos seis caminhos têm uma

estrutura semelhante: uma redução de dimensionalidade aprendida seguida por uma convolução 1D com um tamanho de *kernel* (fragmento de sequência considerado) ascendente - 3, 5, 7, 9, 11 e 13 aminoácidos. O caminho final dentro desse bloco é uma operação MaxPool de zero parâmetros. Concatenar os resultados de todos os 8 caminhos gera uma saída com a mesma dimensionalidade da entrada (controlada pelo hiperparâmetro  $d$ ), exceto que após passar pela camada, os resíduos são contaminados por informações sobre seus vizinhos em graus variados de separação. Essa camada foi batizada de InceptiGOR8 em referência ao seu processo de criação, e sua estrutura pode ser apreciada na Figura 4.2.

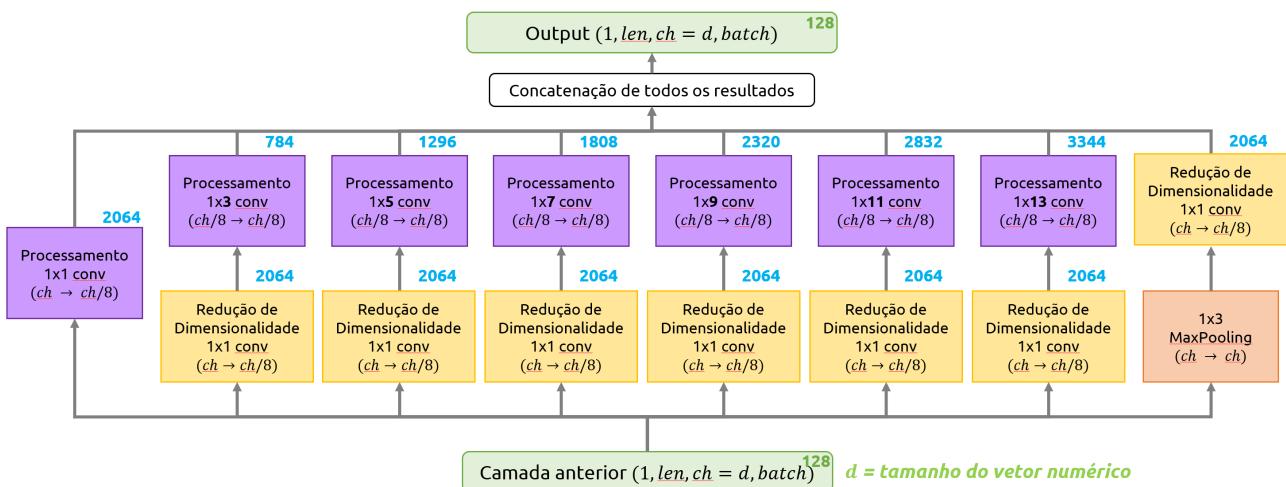


Figura 4.2: Estrutura detalhada do bloco InceptiGOR de convoluções largas e paralelas. Observe os 8 blocos paralelos, com as reduções de dimensionalidade aprendidas, processamento paralelo de kernels de resíduos crescentes e maxpooling. Em azul, o número de parâmetros aprendidos para cada operação, considerando uma entrada de 128 canais ( $d=128$ ).

O último elemento do bloco codificador é um par de camadas de codificador Transformer bidirecionais, organizadas da mesma maneira que no núcleo das arquiteturas BERT [80]. Cada camada é composta por 8 cabeças de atenção, com um tamanho oculto de 16, encimado por uma camada *positionwise* densa de tamanho oculto 128, que novamente gera um tensor com a mesma dimensionalidade da entrada. A codificação posicional é aplicada à sequência entre os blocos convolucionais e Transformer.

É importante mencionar que nós entendemos que o estado da arte da arquitetura BERT para classificação de *tokens* em uma sentença envolve um extenso pré-treinamento em uma tarefa de linguagem mascarada, com posterior ajuste fino. Não é o caso aqui: optamos deliberadamente por utilizar um bloco BERT puramente como um componente de codificação bidirecional de sequência

inteira, para ser treinado do zero com todo o modelo, já que o pré-treinamento de um codificador BERT por si próprio, de tamanho razoável (em relação ao número de parâmetros e amostras) exigiria muito mais recursos de computação.

O componente final, o único que difere entre as redes classificadora e regressora, é a cabeça de previsão. Para as redes classificadoras, ele é composto por dois *kernels* de convolução sucessiva de 5 resíduos, com funções de ativação respectivamente *sigmoid* e *logsoftmax*. Para regressão, existem três camadas de convolução consecutivas com *kernels* de 7 resíduos, cujas funções de ativação são *relu*, *relu* e *tanh*.

Optamos por implementar o modelo utilizando a linguagem Julia [81], utilizando o framework *Flux.jl* [82] e as arquiteturas contribuídas ao Flux pelo pacote *Transformers.jl*. Julia é uma nova linguagem orientada para computação numérica e científica, cuja combinação de simplicidade e velocidade já gerou implementações populares de vários algoritmos de IA, como exemplificado pelo *AlphaZero.jl* [83].

Batizamos o modelo de **TintiNet**: **T**opological **I**nference by **N**eural **T**ransformer-**I**nceptiGOR **N**etwork. O projeto é 100% open source e open data. Todo o código-fonte, com exemplos de rede regressora e classificadora treináveis, juntamente com *scripts* de previsão, pode ser encontrado em <https://github.com/hugemiler/TintiNet.jl>.

### 4.2.3 Treinamento da rede neural

As sequências da base de dados CATHS40, devidamente mapeadas para as classes SS3, filtradas e aparadas para comprimento máximo de 192 aminoácidos, foram divididas em 10 *folds* para treinamento com validação cruzada das redes de regressão e classificação. Em ambos os casos, as entradas foram fornecidas em lotes aleatórios de 128 sequências sem repetição ao longo da mesma época, independente da distribuição de comprimento. Sequências mais curtas foram preenchidas com o elemento neutro '-' para corresponder ao comprimento da maior proteína do lote e, consequentemente, tanto a função objetivo (*loss*) de treinamento, quanto as métricas de desempenho (mais detalhes na seção 4.4) foram mascaradas para esses resíduos. O mascaramento também foi aplicado em resíduos para os quais o DSSP não foi capaz de produzir uma saída.

Para classificação, foi empregada uma função objetivo de log-entropia-cruzada

mascarada, enquanto para regressão a função objetivo foi erro quadrático médio mascarado. Nenhuma abordagem de regularização relacionada aos valores dos parâmetros (L1/L2) foi empregada, ao que optamos por utilizar regularização do tipo dropout entre as camadas convolucionais. Ambas as redes foram treinadas com o otimizador Adam, com lotes de 128 sequências e taxas de aprendizado inicial de 0,001 para o classificador e 0,0001 para o regressor. O treinamento foi realizado por 40 épocas com registro de *checkpoints* após 64 lotes, que produziu aproximadamente 300 *checkpoints* por *fold*, por rede.

Para combater o sobre-ajuste (do inglês *overfitting*, situação em que o modelo se especializa no conjunto de treinamento às custas da sua generalizabilidade para dados inéditos, geralmente causado pela extensão do treinamento ou excesso de parâmetros), empregamos uma estratégia de interrupção adiantada (do inglês, *early stopping*) retroativa. Permitimos que o modelo treinasse por mais tempo que o necessário, efetivamente amostrando diversas épocas em que ele já está saturado da performance máxima que pode alcançar no conjunto de validação. Dessa forma, pudemos determinar a distribuição de métricas do platô de saturação e, assim, determinar o momento exato em que a média móvel da função objetivo do conjunto de validação deixou de ser significativamente diferente dessa ( $\alpha = 95\%$ ), por meio de um teste  $t$  entre os pontos da média móvel e a distribuição da parte plana da curva, sendo mantido por 10 épocas.

## 4.3 Métricas de análise dos resultados

### 4.3.1 Conjuntos de dados para avaliação

Elaboramos três estratégias diferentes para avaliar o modelo TintiNet.jl juntamente com os outros modelos considerados.

A primeira estratégia, chamada VF3162 (“Validation Fold - 3162 amostras”), é a estratégia-padrão de validação cruzada empregada em problemas de *Machine Learning*, representada por uma média das métricas em todos os *folds* de validação. É importante relatar os resultados para um conjunto de validação no caso de um algoritmo de ML, porque isso permite verificar a generalização do modelo, uma vez que nenhuma das proteínas dos conjuntos de validação são vistas pelos *checkpoints* do modelo durante o treinamento.

Numa situação de *benchmark* comparativo, no entanto, exibir métricas exclusivamente

para o conjunto VF pode ser uma situação injusta para a TintiNet.jl frente aos demais modelos, pois eles podem ter sido treinados em parte das sequências em que nosso modelo está sendo avaliado. A forma provavelmente mais rigorosa de resolver essa questão seria obter os conjuntos de treinamento e teste de cada um dos modelos comparados e realizar uma matriz comparativa com treinamentos separados da TintiNet.jl em cada um deles. Essa estratégia, no entanto, tem alguns problemas:

- não temos acesso aos conjuntos de treinamento e teste de **todos** os outros modelos;
- algumas das proteínas contidas nos conjuntos que nos conseguimos obter já não estão mais disponíveis nas bases de dados originais (são de versões anteriores do CATH com representantes ligeiramente diferentes que foram atualizados ou removidos por motivos de controle de qualidade, ou mudança do *cluster*);
- essa matriz comparativa ainda estaria incompleta porque não seria possível comparar os outros modelos entre si com os mesmos conjuntos

Portanto, chegamos à conclusão que considerando que:

- essa análise comparativa só será feita para fins exploratórios,
- todo cuidado será tomado na estratégia de *early stopping* para evitar *overfitting*,
- toda conclusão sobre os méritos do modelo deve ser derivada exclusivamente dos conjuntos de validação,

uma avaliação comparativa mais justa e ao mesmo tempo muito mais simples, e que também tem o benefício de imitar um pouco mais o caso de uso real de qualquer um desses algoritmos, seria construir um grande conjunto de avaliação aleatório, permitindo que cada modelo (inclusive o TintiNet.jl) veja uma mistura potencial de proteínas de treinamento e teste. Com isso em mente, selecionamos aleatoriamente um terço de todas as sequências CATH-S40, criando um conjunto FS10558 (“Fair Set - 10558 amostras”) com as mesmas 10.000 sequências para cada algoritmo, independentemente da composição do conjunto de treinamento durante a validação, e avaliamos todos os modelos nesse conjunto também.

Um conjunto final foi construído para determinar a capacidade dos modelos de realizar inferência em sequências para as quais as informações de homologia são superficiais ou inexistentes, consideradas alvos difíceis (“*hard targets*”). Ao executar o software de busca de homologia HHBlits

[84] na base de dados Uniref30 [85] por três iterações em cada sequência, obtivemos o comprimento efetivo do MSA da família de proteínas para cada uma delas, e separamos aquelas proteínas com um valor  $N_{eff}$  menor ou igual a 4.0. Este conjunto de alvos difíceis foi rotulado como HS3949 (“Hard Set - 3949 amostras”).

### 4.3.2 Métricas de avaliação de classificação

#### Acurácia

A figura de mérito mais tradicional para a previsão da estrutura secundária de 3 estados é a acurácia da classificação por resíduo ( $Q_3$ ) [53]. Ele mede a fração (ou proporção) de previsões corretas, que podem ser representadas por

$$Q_3 = \frac{\sum_{i=1}^L M_i}{L}$$

onde  $M_i$  é 1 quando a previsão corresponde à estrutura secundária de DSSP de 3 estados para o resíduo  $i$ , e 0 caso contrário; e  $L$  é o número de resíduos (comprimento da sequência).

#### Segment Overlap Score (SOV)

Outra métrica importante para a classificação da estrutura secundária SS3 é o Segment Overlap Score - SOV [86], que evolui o conceito de acurácia Q3 introduzindo uma maneira de contabilizar as extremidades correspondentes dos segmentos de estrutura secundária, ou a sobreposição efetiva da estrutura secundária prevista e a de referência, derivada do DSSP. Dessa forma, se baseia na sobreposição média entre os segmentos observados e os previstos, em vez da acurácia média por resíduo. Por conta disso, as previsões com alta precisão por resíduo, mas que se desviam das distribuições de comprimento (início e fim do fragmento experimental) recebem pontuações mais baixas. Neste trabalho, empregamos uma versão modificada do SOV descrita por Zemla et al [87]. A definição da medida SOV para um determinado estado **X** (**H**, **E** ou **C**) é a seguinte:

$$\text{SOV}_X = \frac{1}{N_X} \sum_{S_X} \frac{\min(\text{OV}(s_1, s_2)) + \delta(s_1, s_2)}{\max(\text{OV}(s_1, s_2))} \quad (4.3.1)$$

onde  $s_1$  e  $s_2$  são os segmentos de estrutura secundária observados e previstos no estado  $\mathbf{X}$ ;  $S_{\mathbf{X}}$  é o número de todos os pares de segmentos  $(s_1, s_2)$ , onde  $s_1$  e  $s_2$  têm pelo menos um resíduo em comum no estado  $\mathbf{X}$ ,  $\min(\text{OV}(s_1, s_2))$  é o comprimento da sobreposição real de  $s_1$  e  $s_2$  e  $\max(\text{OV}(s_1, s_2))$  é o comprimento do total pelo qual qualquer um dos segmentos  $s_1$  ou  $s_2$  tem um resíduo no estado  $\mathbf{X}$ .  $N_{\mathbf{X}}$  é o número total de resíduos de aminoácidos observados na conformação  $\mathbf{X}$ . A definição de  $\delta(s_1, s_2)$  é a seguinte [87]:

$$\delta(s_1, s_2) = \min \begin{cases} \max(\text{OV}(s_1, s_2)) - \min(\text{OV}(s_1, s_2)) \\ \min(\text{OV}(s_1, s_2)) \\ \text{int}(0.5 \times \text{len}(s_1)) \\ \text{int}(0.5 \times \text{len}(s_2)) \end{cases} \quad (4.3.2)$$

onde  $\text{len}(s_1)$  é o número de resíduos de aminoácidos no segmento  $s_1$ . A medida de sobreposição de segmento para todos os três estados (**H**, **E** ou **C**), chamada SOV3(%), é então calculada sobre todos os resíduos de maneira semelhante à medida de acurácia Q3(%):

$$\text{SOV3}(\%) = \frac{1}{N} \left( \sum_{i \in \mathbf{H}, \mathbf{E}, \mathbf{C}} \sum_{S(i)} \left[ \frac{\min(\text{OV}(s_1, s_2)) + \delta(s_1, s_2)}{\max(\text{OV}(s_1, s_2))} \times \text{len}(s_1) \right] \right) \times 100 \quad (4.3.3)$$

### Métricas da Matriz de Confusão para classificações binárias um-contra-todos

Como estamos tratando de um problema de classificação multi-classes (três classes), a Matriz de Confusão real do modelo é uma matriz  $3 \times 3$  contendo as classes **H**, **E** e **C** em cada eixo. No entanto, uma maneira alternativa de interpretar o problema é transformar essa única matriz  $3 \times 3$  em três matrizes  $2 \times 2$  individuais, cada uma se referindo a um problema de classificação binária “um-contra-todos”. Nessa situação, para uma dessas novas matrizes, por exemplo, a classe positiva seria **H**, com **E** e **C** conjuntamente perfazendo a classe negativa. Em outra, a classe positiva seria **E**, com **H** e **C** conjuntamente perfazendo a classe negativa; na última, a classe positiva seria **C**, com **H** e **E** conjuntamente perfazendo a classe negativa.

Nesses casos, será possível calcular figuras de mérito derivadas de matrizes de confusão

de classificações binárias, e associá-las com cada uma das classes. Utilizaremos neste trabalho a sensibilidade ( $TVP$ ), especificidade ( $TVN$ ) e acurácia balanceada ( $AB$ ) das classificações, definidas como:

$$TVP = \frac{VP}{VP + FN} \quad (4.3.4)$$

$$TVN = \frac{VN}{VN + FP} \quad (4.3.5)$$

$$AB = \frac{TVP + TVN}{2} \quad (4.3.6)$$

onde  $VP$  é o número de Verdadeiros-Positivos,  $VN$  é o número de Verdadeiros-Negativos,  $FP$  é o número de Falsos-Positivos e  $FN$  é o número de Falsos-Negativos.

### 4.3.3 Métricas de avaliação de regressão

#### Erro Absoluto Médio

Inspirados nas publicações do SPIDER3-single, ProteinUnet e SPOT1D, empregamos o Erro Absoluto Médio (MAE) como o principal indicador de desempenho para regressão:

$$MAE = \frac{\sum_{i=1}^n |y_i - \bar{y}_i|}{n} = \frac{\sum_{i=1}^n |\varepsilon_i|}{n} \quad (4.3.7)$$

onde  $y_i$  é o valor numérico real da propriedade estimada, e  $\bar{y}_i$  é o valor estimado pelo modelo.

#### Erro Absoluto Médio Circular

Para acessibilidade ao solvente, o MAE convencional foi a métrica de desempenho; no entanto, como a rede produz previsões de seno e cosseno para os ângulos  $\varphi$ ,  $\psi$ , empregamos nesses casos o MAE circular (cMAE), de acordo com as equações:

$$\alpha_i = \left| \tan^{-1} \frac{\sin(\theta_{i,pred})}{\cos(\theta_{i,pred})} - \theta_{i,DSSP} \right|$$

$$cAE = \text{Min}(\alpha_i, (360^\circ - \alpha_i))$$

$$cMAE = \frac{\sum_{i=1}^n cAE}{n}$$

onde  $\alpha_i$  é o ângulo calculado a partir dos valores de seno e cosseno que saem da rede neural,  $cAE$  é o erro absoluto circular para um único aminoácido, e  $cMAE$  é o MAE circular considerando todos os aminoácidos.

#### 4.4 *Benchmark computacional*

Experimentos de *benchmark* (comparações de desempenho entre os modelos) foram realizados para avaliar o tempo médio necessário para gerar uma previsão para uma sequência de entrada. Para excluir o tempo de “aquecimento” (compilação, inicialização) para cada algoritmo, montamos um protocolo em que cada previsor receberia quantidades definidas de sequências, em etapas incrementais lineares de 1.000 a 10.000 em cinco réplicas, gerando 50 experimentos para cada um dos 4 preditores, ou um total de 200 experimentos.

A métrica de avaliação do resultado é o próprio tempo real (em segundos) necessário apenas para a inferência. Todos os *benchmarks* rodaram no mesmo hardware, que era um notebook AVELL A62-5 MUV, com um Intel CORE I5-9300H, 16GB de RAM e uma GPU Nvidia GeForce GTX 1660Ti (6GB GDDR5, capacidade de computação 7.5). De todos os 4 modelos, TintiNet.jl, SPOT1D-single e ProteinUnet foram compilados e executados na GPU, mas o SPIDER3-single não está pronto para GPU e foi executado em CPU pura usando todos os 8 núcleos.

#### 4.5 Interpretação do modelo

Se recordarmos a arquitetura da TintiNet.jl na Figura 4.1, hipotetizamos que a extração de informação estrutural acontece em nível crescente de distância e complexidade conforme a sequência percorre a rede neural. Ou seja, esperamos que a porção convolucional do codificador, por quanto se responsabiliza de aprender relações de curto alcance, **desonere** a porção BERT da rede de ter que aprender essas mesmas informações de forma redundante, de maneira que as camadas do tipo Transformer podem focar apenas em capturar relações de longo alcance e refinar as saídas das camadas anteriores com a captura de padrões complexos que as convoluções não são capazes de abstrair. Dessa forma, a interpretação dos modelos será focada no que

acontece antes e durante o percorrimento das camadas BERT pelos dados.

#### 4.5.1 Interpretando operações de atenção

Nas arquiteturas tipo Transformer, chama-se mecanismo de **atenção** o componente responsável por *pesar dinamicamente* a importância das diferentes partes da sequência de entrada para cada *token* de saída. A interpretação do processo de tomada de decisão de um modelo Transformer depende, portanto, dessas pontuações de atenção. No caso das camadas BERT utilizadas na TintiNet.jl, o mecanismo de atenção se denomina auto-atenção (do inglês, *self-attention*), processo que é protagonizado por três componentes principais: Consultas (do inglês, *Queries*), Chaves (do inglês, *Keys*) e Valores (do inglês, *Values*), calculados a partir dos dados que entram na camada. Nesse mecanismo, as pontuações de atenção são primeiro calculadas realizando um produto escalar entre Consultas e Chaves, seguido por uma operação softmax para normalizar as pontuações, garantindo que somam para a unidade.

Essas pontuações se tornam os coeficientes de uma soma ponderada de cada Valor, por meio da qual cada termo que sai da camada em uma determinada posição teve a oportunidade de interagir com todos os outros termos da sequência, em maior ou menor grau (ou, “prestando mais ou menos atenção”) dependendo dos escores de atenção. Dessa maneira, a rede tem a oportunidade de aprender relações entre partes diferentes da sequência independente do seu grau de separação. Em termos algébricos, podemos dizer que

$$\begin{aligned} \text{SelfAttention}(X) &= \text{Attention}(X)V_X \\ \text{Attention}(X) &= \text{softmax}\left(\frac{Q_X K_X^T}{\sqrt{d}}\right) \\ \text{SelfAttention}(X) &= \text{softmax}\left(\frac{Q_X K_X^T}{\sqrt{d}}\right) V_X \end{aligned}$$

onde  $Q_X = XW^Q$ ,  $K_X = XW^K$  e  $V_X = XW^V$ , sendo  $W^Q$ ,  $W^K$ , e  $W^V$  os parâmetros treináveis da operação de auto-atenção.

#### 4.5.2 Protocolo geral de interpretação

Conforme o exposto, para interrogar o modelo, primeiro investigaremos o formato do tensor ( $X$ ) que simboliza a sequência primária de uma proteína logo depois de sair das camadas convolucionais. Depois, acompanhando esse tensor passo a passo do modelo Transformer, extrairemos as pontuações brutas de atenção ( $\text{Attention}(X)$ ) do modelo. Isso envolve acessar as camadas de atenção dentro do transformador e recuperar as pontuações geradas durante a passagem direta. Com as pontuações em mãos, analisaremos as matrizes de atenção para entender o peso dado por cada resíduo a cada outro resíduo para geração de seu próprio sinal de saída específico. Pontuações de atenção mais altas indicam uma relação ou relevância mais forte entre *tokens* de entrada e saída. Dada a natureza multcamadas e multi-cabeças das arquiteturas de transformadores, as pontuações de atenção estão dispersas por várias matrizes. Agregar essas pontuações de forma significativa é crucial para a interpretação. Usaremos mapas de calor para as múltiplas cabeças de previsão para visualização e compreensão de padrões, e estruturas moleculares para interpretação.

## Capítulo 5

# Resultados e Discussões

### 5.1 Treinamento do modelo

#### 5.1.1 Curvas de Aprendizado

Curvas de aprendizado são gráficos que apresentam o valor da função objetivo ou outras métricas de avaliação ao longo dos passos da otimização realizada no treinamento do modelo. Geralmente, as curvas são exibidas em pares: uma para o conjunto de treinamento e outra para o conjunto de validação. Quando a função objetivo é um erro (como, por exemplo, a entropia cruzada utilizada pelo classificador da TintiNet.jl), a curva de aprendizado deve exibir um comportamento decrescente até chegar à estabilidade quase simultânea dos erros de treinamento e de validação. Durante o período em que o modelo adquire viés generativo sobre os dados, observa-se o progresso de ambos os conjuntos. Fatalmente, se o modelo tiver tempo suficiente para ser treinado, alcançará uma situação em que a performance no conjunto de validação fica estagnada, mas continua melhorando no conjunto de treinamento. Trata-se da transição para o estado de sobre-ajuste (do inglês, *overfitting*), que pode ser combatido por meio de estratégias como o monitoramento do conjunto de validação e a interrupção adiantada do treinamento (do inglês, *early-stopping*).

A Figura 5.1 mostra o acompanhamento do treino do modelo de classificação. A primeira observação é que, tanto no painel A quanto no B, existe uma grande concordância visual entre as 10 curvas plotadas para cada conjunto (treino e validação), o que não dá indícios de problemas na amostragem na construção dos 10 *folds* de validação. A inspeção visual das curvas no painel A mostra que uma situação de estabilidade do erro de validação foi alcançada entre 10 e 20 épocas, reforçada pela estabilidade da acurácia no mesmo período no painel B. Para confirmar o número

com precisão, calculamos a média e o desvio padrão do erro de validação na porção plana da curva (entre 30 e 40 épocas), depois suavizamos as curvas de aprendizado com um filtro de média móvel e computamos o número de épocas decorridas, desde o início do treinamento, até que esse erro não fosse significativamente diferente da região plana. Em média, os 10 folds indicaram que o treinamento deveria terminar após 17 épocas.

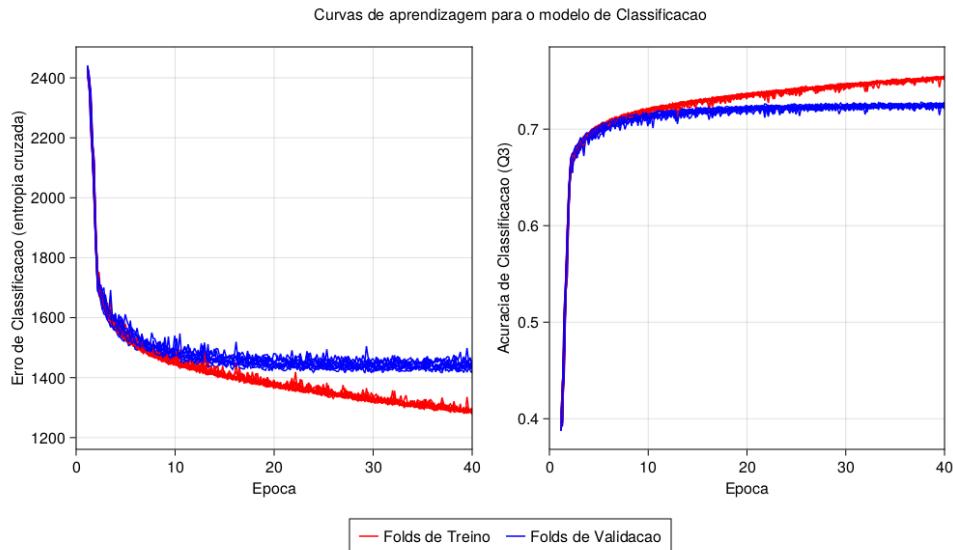


Figura 5.1: Curvas de aprendizado para o modelo de Classificação da TintiNet.jl. O monitoramento ocorreu durante 50 épocas. A esquerda, acompanha-se a função objetivo medida como Entropia Cruzada mascarada, nos conjuntos de treino e validação, para cada um dos 10 *folds*, onde o eixo vertical representa a soma da entropia cruzada, normalizada para o tamanho do *fold* de validação. A direita, a acurácia do modelo nos conjuntos de treino e validação, para cada um dos 10 *folds*.

De forma análoga, a Figura 5.2 mostra o acompanhamento do treino do modelo de regressão. Cabe aqui o mesmo comentário, em que a concordância visual entre as 10 curvas plotadas para cada conjunto não dá indícios de problemas na amostragem na construção dos 10 *folds* de validação. Utilizando o mesmo método descrito para as curvas de aprendizado no modelo de classificação, alcançamos um número médio de 67 épocas até que o erro do conjunto de treinamento deixasse de ser significativamente diferente do erro na região plana da curva.

## 5.2 Resultados da validação cruzada

Após treinamento dos módulos de classificação e regressão, a TintiNet.jl foi avaliada nos três conjuntos de avaliação juntamente com os preditores de referência

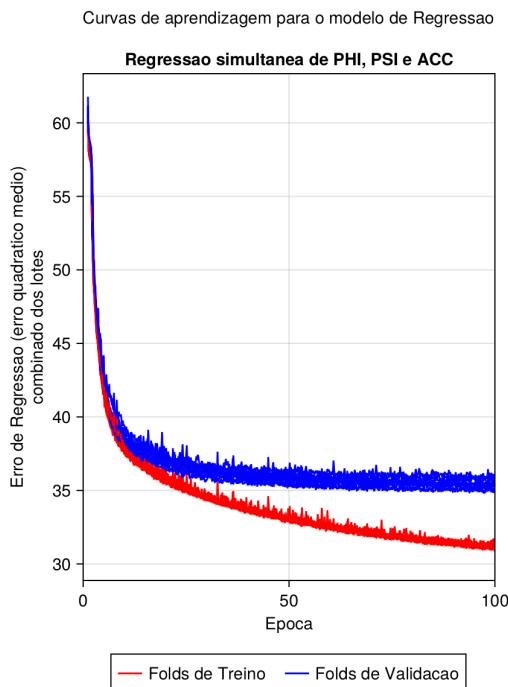


Figura 5.2: Curvas de aprendizado para o modelo de Regressão da TintiNet.jl. O monitoramento ocorreu durante 100 épocas. A função objetivo foi o Erro quadrático médio mascarado, nos conjuntos de treino e validação, para cada um dos 10 *folds*.

SPIDER3-single, ProteinUnet e SPO1D-single, de acordo com as métricas mostradas na seção 4.3. A Tabela 5.1 resume os resultados alcançados, que serão discutidos nas seções a seguir.

### 5.3 Análise dos méritos do modelo Classificador

A Figura 5.3 mostra a distribuição dos méritos de classificação entre os 4 métodos testados. Mesmo em um conjunto totalmente composto por sequências inéditas em tempo de treinamento (VF3162), o TintiNet.jl supera os outros três modelos em termos de métricas de classificação, atingindo 72,69% de acurácia e um SOV de 69,32 no conjunto de validação escolhido para representar o modelo neste trabalho. Quando avaliado nos conjuntos FS e HS, devido à inclusão de um percentual de sequências de treinamento, seu desempenho apresenta um aumento significativo.

Esses resultados demonstram que, pelo menos do ponto de vista das métricas mais comuns da ciência de dados (aqueles relacionadas à qualidade da estimativa em si), nós cumprimos nosso objetivo de desenvolver um classificador de SS3 baseado em sequências únicas

(ainda que não maiores que 192 resíduos). Mesmo com um número muito inferior de parâmetros treináveis - chegando a ter menos de 25% dos parâmetros do menor modelo concorrente, e apenas **0,44% dos parâmetros do maior concorrente**, TintiNet.jl alcançou os maiores valores para figuras de mérito de classificação, equilibrando uma contagem menor de parâmetros com paradigmas arquitetônicos mais modernos, oferecendo acurácia de classificação competitiva. Cumprimos aqui nosso princípio de design que se compromete a ser “menor, e tão bom quanto” ao invés de ir na direção de “maior e melhor”. Os números detalhados de parâmetros de cada modelo podem ser observados na Tabela 5.2.

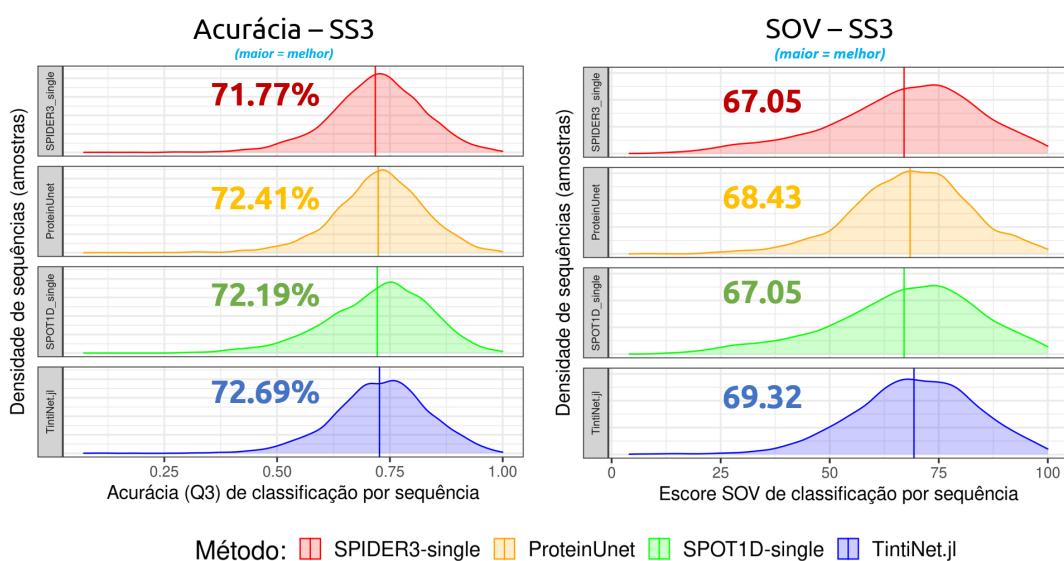


Figura 5.3: Distribuição das figuras de mérito de Classificação (Q3 e SOV) para os quatro modelos avaliados no conjunto VF3162. As linhas horizontais representam os valores médios.

Uma das formas de entender melhor a generalização e a perplexidade do modelo classificador da TintiNet.jl em comparação com os demais é analisar as matrizes de confusão para classificação no conjunto VF para todos os quatro modelos avaliados. Essas matrizes mostram como se distribuem os mais de 300.000 aminoácidos das 3162 sequências do *fold* 10 deste conjunto em relação a classe real e a classe sugerida pela rede neural, e são apresentadas na Figura 5.4 a seguir.

Pode-se perceber que, dentre os quatro modelos, a TintiNet.jl não é nem o melhor, nem o pior classificador para nenhuma das três classes de estrutura secundária alvo. Assim como os demais modelos, ela apresenta o pior desempenho na classificação de estruturas secundárias do tipo fita- $\beta$  (E), e erra a classificação de mais de 35% desses aminoácidos, inferindo que a esmagadora

Tabela 5.1: Resumo das Figuras de Mérito para Classificação e Regressão para o modelo TintiNet.jl comparado com SPIDER3-single, ProteinUnet e SPOT1D-single

Resultados no conjunto VF3162 (Validation Fold)				
Critério	TintiNet.jl	SPIDER3-single	ProteinUnet	SPOT1D-single
Acurácia SS3 (Q3)	72.69%	71.77%	72.41%	72.19%
SOV SS3	69.32	67.05	68.43	67.05
MAE do ângulo PHI / graus	24.28	25.94	25.72	23.89
MAE do ângulo PSI / graus	42.69	43.79	43.79	39.49
MAE da acess. ao solvente / Å <sup>2</sup>	28.83	30.72	29.93	27.27
Resultados no conjunto FS10551 (Fair Set)				
Critério	TintiNet.jl	SPIDER3-single	ProteinUnet	SPOT1D-single
Acurácia SS3 (Q3)	75.31%	71.47%	72.06%	71.94%
SOV SS3	71.76	66.76	67.90	66.76
MAE do ângulo PHI / graus	22.88	26.25	26.02	24.20
MAE do ângulo PSI / graus	36.37	44.17	44.19	39.99
MAE da acess. ao solvente / Å <sup>2</sup>	28.33	30.71	30.11	27.44
Resultados no conjunto HS3949 (Hard Set)				
Critério	TintiNet.jl	SPIDER3-single	ProteinUnet	SPOT1D-single
Acurácia SS3 (Q3)	74.62%	71.58%	71.78%	71.70%
SOV SS3	69.95	65.89	66.55	65.89
MAE do ângulo PHI / graus	23.82	26.46	26.31	24.88
MAE do ângulo PSI / graus	38.40	44.12	44.24	40.77
MAE da acess. ao solvente / Å <sup>2</sup>	30.52	31.05	32.00	28.56

Tabela 5.2: Número aproximado de parâmetros totais dos modelos avaliados

Modelo	Número de Parâmetros
TintiNet.jl	360.000
SPIDER3-single	3.200.000
ProteinUnet	1.600.000
SPOT1D-single	82.000.000

(A) TintiNet.jl				
		Classe Inferida		
		C	E	H
Classe Real	<b>C</b>	<b>100016</b> <b>74,22%</b>	14705 10,91%	20044 14,87%
	<b>E</b>	20807 27,45%	<b>48466</b> <b>63,93%</b>	6534 8,62%
	<b>H</b>	24204 18,72%	6523 5,04%	<b>98581</b> <b>76,24%</b>

(B) SPIDER3-single				
		Classe Inferida		
		C	E	H
Classe Real	<b>C</b>	<b>92117</b> <b>68,35%</b>	18131 13,45%	24517 18,19%
	<b>E</b>	18454 24,34%	<b>50743</b> <b>66,94%</b>	6610 8,72%
	<b>H</b>	21377 16,53%	6859 5,30%	<b>101072</b> <b>78,16%</b>

(C) ProteinUnet				
		Classe Inferida		
		C	E	H
Classe Real	<b>C</b>	<b>97559</b> <b>72,39%</b>	16191 12,01%	21015 15,59%
	<b>E</b>	19971 26,34%	<b>49441</b> <b>65,22%</b>	6395 8,44%
	<b>H</b>	23370 18,07%	6833 5,28%	<b>99105</b> <b>76,64%</b>

(D) SPOT1D-single				
		Classe Inferida		
		C	E	H
Classe Real	<b>C</b>	<b>105813</b> <b>78,52%</b>	11697 8,68%	17255 12,80%
	<b>E</b>	25487 33,62%	<b>45664</b> <b>60,24%</b>	4656 6,14%
	<b>H</b>	31260 24,17%	4164 3,22%	<b>93884</b> <b>72,60%</b>

Figura 5.4: Matrizes de confusão multiclasse para o conjunto VF3162 em todos os 4 modelos avaliados. A diagonal principal (cor azul) indica as classificações corretas. Na configuração escolhida, as linhas da matriz devem somar para 100% em todos os casos.

maioria desses seriam resíduos sem estrutura secundária definida.

A estimativa de estruturas secundárias do tipo fita-beta em modelos preditivos apresenta desafios históricos significativos devido à complexidade intrínseca dessas estruturas em proteínas. Desde o início desse esforço com o método de Chou-Fasman e o próprio método GOR, passando pela era pré-*Machine Learning* com métodos importantes como PSIPRED [28] e JPRED [29], existe uma lacuna significativa entre a acurácia atingida em hélices- $\alpha$  e fitas- $\beta$ .

Um dos principais desafios na predição de fitas-beta está relacionado à sua diversidade e complexidade estrutural, diferentemente das hélices- $\alpha$ , que possuem uma geometria mais regular e previsível. Fitas- $\beta$  podem fazer parte de extensos arranjos estendidos na forma de folhas através de ligações de hidrogênio entre segmentos da cadeia, o que por si só já introduz uma complexidade relacionada ao pareamento que não existe nas hélices- $\alpha$ . Essas folhas podem ser paralelas, antiparalelas ou mistas, dependendo da direção das cadeias adjacentes, e podem formar arranjos extensos e intrincados, como barris- $\beta$  e sanduíches- $\beta$ , que são difíceis de prever a partir da sequência de aminoácidos sozinha. Elas também podem existir fora das configurações de folha, geralmente em segmentos menores, entremeados a segmentos de hélices ou na transição entre

motivos estruturais.

Um segundo problema vem do assinalamento dessa estrutura. No início da metodologia, criamos um mapeamento de *outputs* do padrão DSSP SS8/SS9 para SS3, seguindo regras já estabelecidas dessa conversão. As classes SS8 **E** e **B** são, portanto, convertidas para a classe SS3 **E**. A classe **B** representa segmentos de estrutura secundária onde existe um único resíduo cujos ângulos e ligações hidrogênio satisfazem as condições para classificação como fita- $\beta$ . Trata-se de uma situação rara, correspondendo a aproximadamente 5% dos resíduos contabilizados como **E** ao final, mas ainda assim significativo para esse fenômeno, por ter uma característica estrutural muito diferente do padrão de fitas-beta contínuas e estendidas (estejam elas em folhas ou não).

Inicialmente, poder-se-ia sugerir que o modelo TintiNet.jl adota uma postura conservadora nesse caso: sem evidência de um sinal suficientemente alto, prefere não classificar a estrutura secundária como **H** ou **E**. No entanto, um estudo minucioso das matrizes de confusão revela o contrário. Se recordarmos que a série temporal de desenvolvimento dos modelos concorrentes foi SPIDER3-single, depois ProteinUnet e, por fim, SPOT1D-single, percebemos que, ao longo do tempo, havia um movimento constante de aumento do conservadorismo do modelo: o SPOT1D-single acerta muito mais na classe **C** às custas da acurácia nas classes **E** e **H** comparado ao ProteinUnet. Da mesma forma, este último acerta muito mais na classe **C** também às custas da acurácia nas classes **E** e **H** comparado ao SPIDER3-single. Portanto, percebemos aqui que, na realidade, enquanto o TintiNet.jl mantém a capacidade preditiva crua do SPOT1D-single (em porcentagem média), ele tem a vantagem de gerar estimativas mais equilibradas entre as classes. Isso pode ser melhor apreciado por meio dos dados da Tabela 5.3: do Spider3-single até o SPOT1D-single, o incremento de qualidade veio com o aumento da sensibilidade a **C**, mas às custas da sensibilidade a **E**. TintiNet.jl quebra essa tendência, aumentando a acurácia em relação ao seu predecessor *ao mesmo tempo* em que recupera parte da sensibilidade a **E** que vinha sendo, historicamente, perdida pelo seu antecessor direto (SPOT1D-single).

Tabela 5.3: Figuras de mérito para a classificação binária “um-contra-todos” por categoria de estrutura terciária

Figura de Mérito	Spider3-Single	ProteinUnet	Spot1D-Single	TintiNet.jl
Sensibilidade C	0.684	0.724	0.785	0.742
Especificidade C	0.806	0.789	0.723	0.781
Acurácia Balanceada C	74.47%	75.63%	75.43%	76.14%
Sensibilidade E	0.669	0.652	0.602	0.639
Especificidade E	0.905	0.913	0.940	0.920
Acurácia Balanceada E	78.71%	78.29%	77.13%	77.94%
Sensibilidade H	0.660	0.616	0.587	0.608
Especificidade H	0.897	0.919	0.929	0.919
Acurácia Balanceada H	77.88%	76.79%	75.82%	76.36%

### 5.3.1 Dependência da classificação com propriedades genéricas da sequência

Como nosso modelo foi baseado em modelos de linguagem em sequências únicas e independentes, algumas questões derivadas do domínio de NLP podem ser adaptadas para o domínio da proteômica estrutural e exploradas para nos ajudar a entender melhor o funcionamento do modelo.

Primeiramente, em NLP, é comum o questionamento sobre a dependência da qualidade de uma tradução e o tamanho da frase, do período, ou texto que é fornecido como entrada. Adaptada, essa questão se torna *“a performance do modelo depende do comprimento da sequência primária de cada proteína?”* De forma similar, em NLP, é comum que se questione sobre a capacidade do modelo de trabalhar com palavras, construções e conceitos mais ou menos frequentes, tanto na entrada quanto na saída. Podemos adaptar esse questionamento na forma de duas outras questões: *“a performance do modelo depende da quantidade de homólogos que uma determinada proteína tem?”* e *“a qualidade do modelo depende da raridade do aminoácido que lhe é apresentado ou do desequilíbrio de classes a ser previsto?”*

Em relação à primeira questão, sabe-se que, para modelos modernos e robustos de linguagem, o desempenho não deve depender do tamanho da sentença. Para descobrir se esse também era o caso em nosso modelo de linguagem, investigamos a relação entre o comprimento da sequência e a acurácia de classificação (Q3) por sequência em todo o conjunto VF (Figura 5.5). Utilizamos duas métricas para avaliar a covariância dessas medidas. A primeira delas foi o clássico coeficiente de correlação de Pearson ( $r$ ) para comparação entre vetores de valores numéricos reais.

Com  $r^2 = 0.008$  ( $p > 0.05$ ), não podemos dizer que a qualidade da previsão está significativamente correlacionada com o comprimento da sequência.

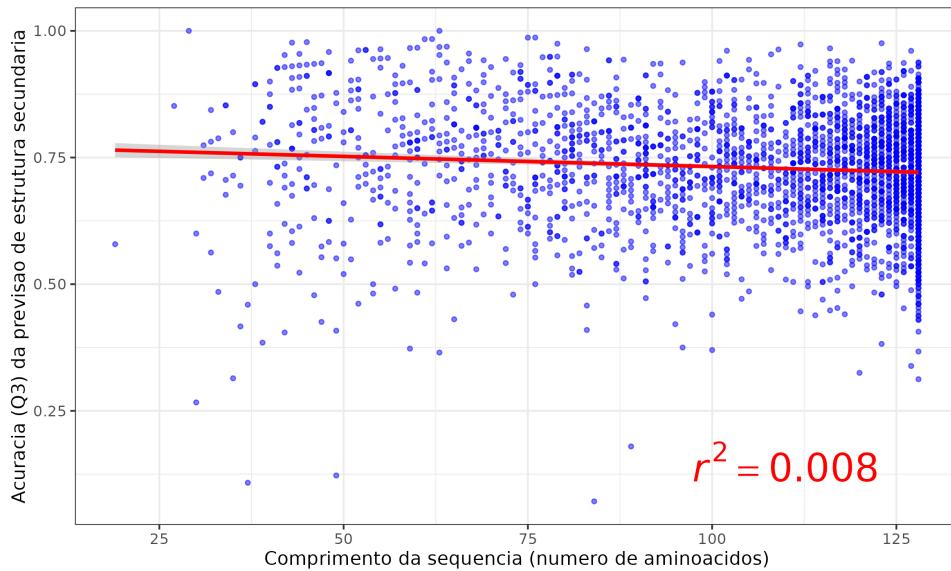


Figura 5.5: Avaliação dos méritos derivados do domínio de NLP na arquitetura TintiNet.jl. Acurácia de classificação em função do comprimento da sequência para o conjunto VF. A linha vermelha mostra o resultado de uma regressão linear.

Contudo, pode-se afirmar que o comprimento da sequência não é uma medida contínua, sendo composta apenas por números inteiros, e que, por consequência, para cada caso de comprimento de sequência, os possíveis valores de acurácia são também discretizados (é esse fenômeno, inclusive, que confere a Figura 5.5) o seu “desenho” interessante. Dessa forma, rigorosamente, poderíamos estar violando fracamente as condições de contorno da correlação de Pearson. Portanto, utilizamos também uma medida de correlação não-paramétrica para confirmar essa observação, escolhendo a correlação de Spearman ( $\rho$ ). Com  $\rho^2 = 0.012$  ( $p > 0.05$ ), confirmamos novamente que não há rejeição da hipótese nula; a qualidade da previsão não está significativamente correlacionada com o comprimento da sequência.

Verificar a independência entre desempenho e grau de homologia (segunda questão) foi importante, pois uma das justificativas para o treinamento de modelos de sequência única é que, considerando a ausência de informações evolutivas para apoiar a previsão, o modelo é independente do tamanho e diversidade da família, proporcionando desempenho semelhante em alvos fáceis e difíceis. Calculamos a correlação entre o MSA  $N_{eff}$  derivado do HHBlits e a acurácia da classificação para o conjunto VF para avaliar essa questão (Figura 5.6). Com  $r^2 = 0.005$ , observamos que a

qualidade da previsão não está correlacionada com MSA  $N_{eff}$  e, portanto, não depende do grau de homologia da sequência de entrada.

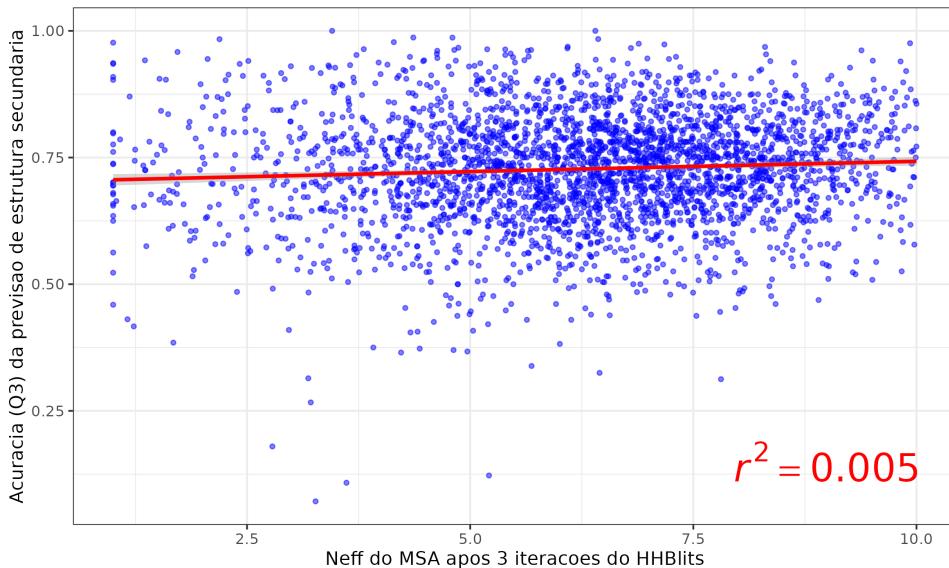


Figura 5.6: Avaliação dos méritos derivados do domínio de NLP na arquitetura TintiNet.jl. Acurácia de classificação em função de MSA Neff de HHBlits para o conjunto VF. A linha vermelha mostra o resultado de uma regressão linear.

Essa conclusão pode ser reforçada por testes de hipótese adicionais. Uma das formas seria, por exemplo, comparar as médias e desvios-padrão dos valores de Q3 entre amostras com  $N_{eff}$  baixo e  $N_{eff}$  alto, sob a hipótese nula de que pertencem a mesma distribuição. A realização deste teste a seguir:

$$N_{eff} \leq 2 : \mathcal{N} \sim (0.712, 0.127)$$

$$N_{eff} \geq 9 : \mathcal{N} \sim (0.755, 0.096)$$

$$Z = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}} = -0.273 \quad (p = 0.392)$$

mostra que o p-valor não permite o descarte da hipótese nula, ou seja, não podemos descartar a hipótese que sequências com alto e baixo grau de homologia tem a mesma distribuição de acurácia na classificação.

Para a última pergunta, avaliamos como a acurácia da previsão varia entre *tokens* de entrada raros e frequentes (que, no nosso caso, são justamente os aminoácidos). Palavras raras

podem ser problemáticas em modelos de linguagem e são um ponto de interesse importante em tarefas como tradução automática. A Figura 5.7 foi produzida para determinar a correlação entre a acurácia da previsão e a abundância de resíduos no conjunto VF. Para os três estados, os valores alcançados para  $r^2$  - respectivamente, 0.002 ( $p = 0.849$ ), 0\* ( $p = 0.999$ ) e 0.027 ( $p = 0.476$ ) para H, E e C - mostram que não há correlação significativa ( $\alpha = 95\%$ ) entre a abundância de aminoácidos e a acurácia da previsão de cada estado para esses aminoácidos.

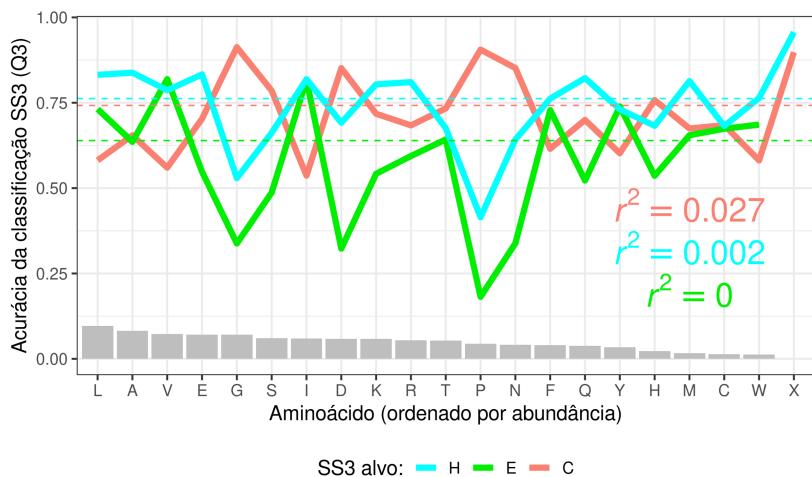


Figura 5.7: Avaliação dos méritos derivados do domínio de NLP na arquitetura TintiNet.jl. Acurácia de classificação para cada estado SS alvo por identidade de aminoácidos, ordenados por abundância (barras cinzas). As linhas horizontais tracejadas mostram a acurácia média de classificação para cada estado, codificado por cores.

Nessa figura, chama atenção o desempenho excepcionalmente baixo do modelo em prever estruturas do tipo fita- $\beta$  especialmente para quatro aminoácidos específicos: Prolina (P), Glicina (G), Asparagina (N) e Ácido Aspártico (D). Quando notamos isso, já sabíamos que a Prolina e a Glicina tinham características estruturais bem específicas - que serão mais profundadas na seção 5.4.1, mas inicialmente não entendemos o que podia estar acontecendo com a dupla N/D. Para investigar esse fenômeno, o primeiro passo foi entender se algum viés populacional poderia estar prejudicando a geração de previsões de estrutura do tipo fita- $\beta$ . Para isso, construímos a Figura 5.8, que mostra a proporção das estruturas secundárias de referência por aminoácido, no *fold* 10 de validação, que verifica uma baixa abundância desses resíduos em fita- $\beta$ .

Posteriormente, encontramos trabalhos que elucidavam as razões moleculares para essa baixa propensão de resíduos N/D a aparecer em fitas- $\beta$  [88, 89]. De acordo com os autores, N/D atuam comumente como interruptores de segmentos de fita- $\beta$ , seja pela indução da rotação do

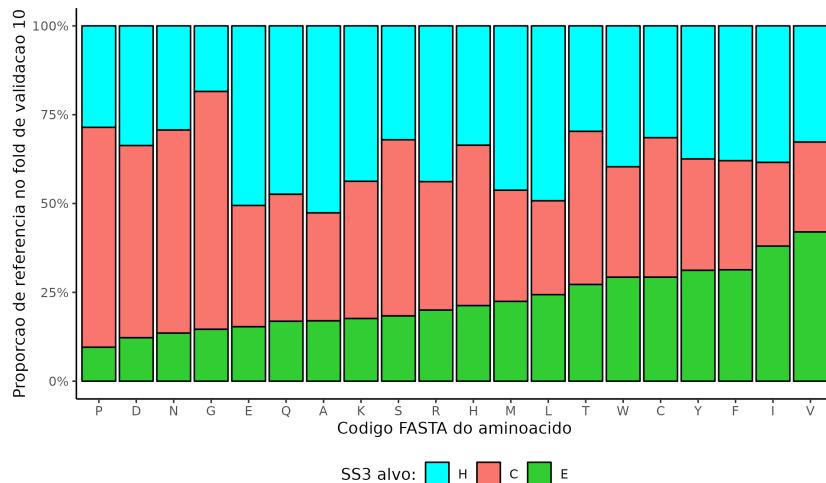


Figura 5.8: Distribuição proporcional entre os estados SS3, para os 20 aminoácidos naturais, para todas as proteínas da base de dados CATHS40.

*backbone* (o que atrapalha o estendimento regular da fita), ou pela formação de uma ligação de hidrogênio com o grupo  $-NH$  do resíduo seguinte pelo seu átomo  $O\delta$ , terminando a formação de folhas.

Acompanhando manualmente algumas das amostras com resíduos N/D erroneamente classificados ao longo da rede até a camada de saída e examinando os *outputs* do modelo antes da etapa de decisão, percebemos em quase todos os casos que uma distribuição de probabilidades entre os estados onde os valores para as classes E e C são muito próximos entre si. É uma situação em que a decisão é tomada com uma resolução muito mais baixa do que o comum para a maioria das classificações, especialmente no caso em que se usa uma função de ativação do tipo softmax, que tende a intensificar muito as diferenças entre os sinais. Uma analogia didática seria dizer que é um caso em que o modelo ficou mais “indeciso” do que normalmente ficaria, e por diversos fatores que não podemos necessariamente explicar - mas que, dentre eles, certamente está uma contribuição massiva do viés de população da classe “C” - a probabilidade da classe “C” acabou ficando muito maior do que deveria ser.

## 5.4 Análise dos méritos do modelo Regressor

A Figura 5.9 mostra a distribuição dos méritos de regressão entre os 4 métodos testados. Para a tarefa de regressão, onde o modelo prevê simultaneamente ângulos de torção (através de seus senos e cossenos) e acessibilidade ao solvente, TintiNet.jl alcançou MAE de 24,28, 42,69 e

28,83, respectivamente para PHI, PSI e ACC no conjunto VF3162. Quando avaliados nas séries TS e HS, houve novamente um aumento significativo no desempenho, pelo mesmo motivo relatado na tarefa de classificação.

Em termos comparativos, para a tarefa de regressão, o TintiNet.jl supera consistentemente tanto o ProteinUnet quanto o SPIDER3-single, mas é marginalmente menos exato do que o SPOT1D-single no conjunto VF. A quantidade de parâmetros no TintiNet.jl é, no entanto, aproximadamente duas ordens de magnitude menor que a do SPOT1D-single. Observamos, portanto, que nossa abordagem atinge um nível competitivo de desempenho de regressão com a menor quantidade de parâmetros, conforme delimitado inicialmente em nossos objetivos. No que concerne os conjuntos TS e HS, o saldo do mérito superior é inclinado para a TintiNet.jl, que se torna o melhor preditor para PHI e PSI e o segundo melhor para ACC (perdendo novamente para SPOT1D-single).

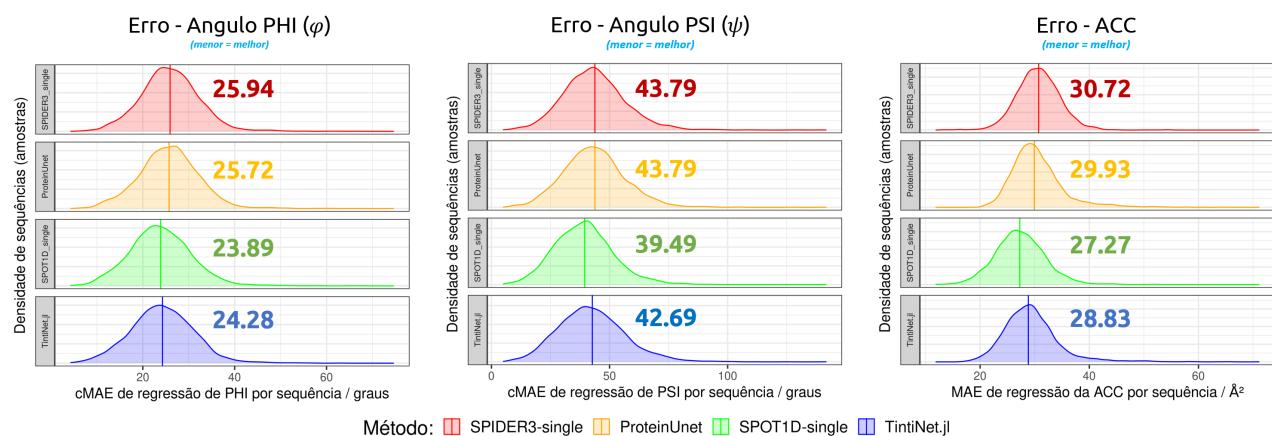


Figura 5.9: Distribuição das figuras de mérito de Regressão (MAE e cMAE) para os quatro modelos avaliados no conjunto VF3162. As linhas horizontais representam os valores médios.

#### 5.4.1 Dependência de desempenho de regressão com propriedades de sequência

A maneira mais imediata de interrogar a relação entre os ângulos de diedro produzidos pela rede neural e a sequência de entrada e entender como - e com qual extensão - esses ângulos estão refletindo o que conhecemos sobre a termodinâmica dos ângulos de diedro dos aminoácidos naturais. Desde a década de 1960 já se sabe que nem todas as conformações no espaço esférico são equiprováveis, nem sequer permitidas. Ainda que o modelo TintiNet.jl esteja produzindo erros de

regressão comparáveis aqueles dos demais concorrentes, questionar se os próprios valores dos ângulos produzidos fazem sentido na paisagem energética das proteínas naturais torna-se imperativo.

O gráfico Ramachandran é uma ferramenta fundamental em biologia estrutural que fornece uma representação visual do espaço conformacional disponível para os ângulos diédricos do *backbone*, Phi ( $\varphi$ ) e Psi ( $\psi$ ), de aminoácidos em uma estrutura proteica. Recebe esse nome em homenagem ao físico indiano G.N. Ramachandran, que o introduziu pela primeira vez em 1963. Este gráfico identifica regiões onde combinações de ângulos  $\varphi$  e  $\psi$  são estericamente viáveis, e pode inclusive ser interpretado como um mapa de densidade de probabilidade de estados conformacionais. Devido ao impedimento estérico e às conformações energeticamente favoráveis, apenas certas regiões do eixo ( $\varphi, \psi$ ) são povoadas por ângulos diédricos de aminoácidos de proteínas naturais no estado nativo. Eles são normalmente divididos em áreas que favorecem hélices- $\alpha$ , folhas- $\beta$  e, de maneira secundária, as hélices canhotas. A Figura 5.10 mostra o gráfico de Ramachandran obtido a partir do processamento de todas as mais de 3000 estruturas de um dos *folds* de validação do projeto (fold 10).

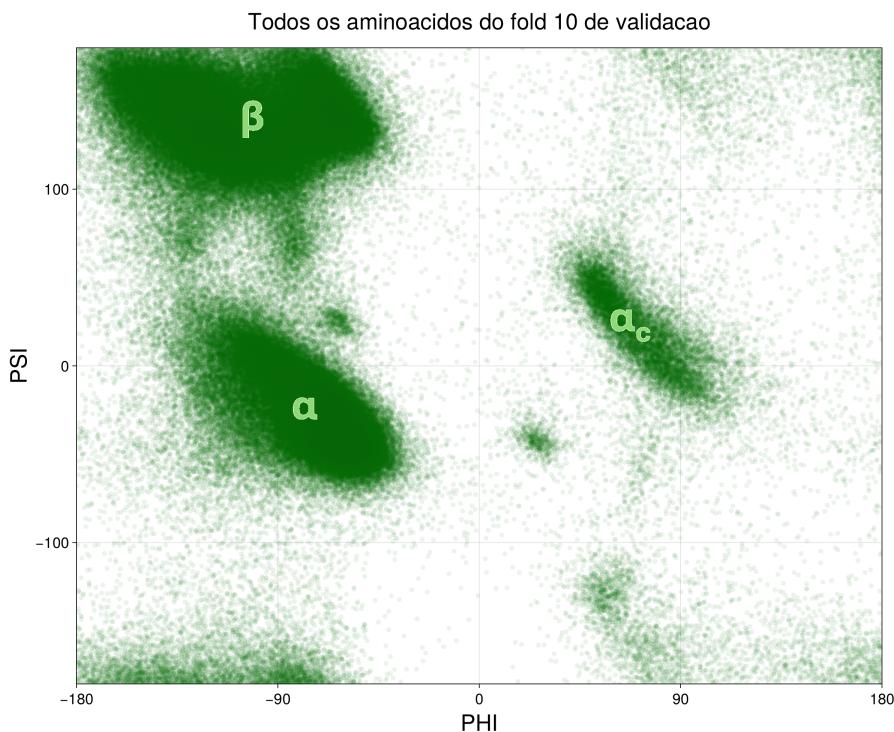


Figura 5.10: Gráfico de Ramachandran computado para o *fold* 10 de validação. Cores mais escuras representam regiões mais permitidas. Destaque para as regiões mais densas do gráfico: no topo e a esquerda, a região de fitas e folhas-beta; no centro, a esquerda, a alfa-hélice, e no centro a direita, a hélice canhota

Nesse gráfico, áreas com poucos ou nenhum ponto representam conformações estericamente impossíveis ou energeticamente desfavoráveis para a estrutura da proteína, principalmente devido a choques entre átomos na cadeia polipeptídica. Embora o gráfico Ramachandran seja geralmente aplicável a todos os aminoácidos, a presença de uma cadeia lateral pode influenciar as regiões permitidas e proibidas. Por exemplo, a glicina, devido à sua pequena cadeia lateral (um único átomo de hidrogênio), tem mais flexibilidade conformacional e, portanto, ocupa uma área mais ampla na parcela. Em contraste, a prolina é mais restrita devido à sua estrutura em anel, o que limita a gama de ângulos viáveis.

Neste trabalho, usaremos os gráficos de Ramachandran de todos os aminoácidos das proteínas de referência no conjunto de validação para explorar a hipótese de que o modelo está aprendendo a prever ângulos  $\varphi$  e  $\psi$  que fazem sentido para cada aminoácido da sequência de entrada. Eles estão representados na Figura 5.11

Uma análise visual ampla e preliminar dos gráficos nos permite observar que, de forma geral, os valores gerados pela TintiNet.jl ocupam as regiões mais permitidas dos gráficos de Ramachandran, enquanto geram muito poucas previsões nas regiões mais proibidas, o que reforça o mérito do modelo implementado. Contudo, também é possível perceber que, em quase todos os gráficos, existe uma tendência acentuada de popular a região do vale entre os máximos de  $\alpha$  e  $\beta$ , no quadrante de  $\varphi$  negativo e  $\psi$  positivo, o que desvia do comportamento de referência.

Embora seja difícil construir experimentos para interpretar esse fenômeno, não podemos ignorar a possibilidade de que o modelo não tenha parâmetros suficientes para adquirir a resolução necessária nessa região do espaço  $(\varphi, \psi)$ . Mesmo que esse seja o caso, os gráficos ainda revelam que, nos quadrantes de  $\varphi < 0$ , os pontos se agrupam em dois *clusters* cujos centrômeros concordam com as regiões mais povoadas do gráfico de referência, reforçando a tese de que, ao invés de produzir valores uniformemente distantes das médias ou medianas dos ângulos de referência, a rede neural está de fato produzindo previsões que refletem - ainda que parcialmente - a termodinâmica factual desses aminoácidos no estado nativo.

Aproveitando a Figura 5.11, podemos selecionar alguns gráficos para uma discussão mais aprofundada. Especialmente, a glicina e a prolina são dois aminoácidos que representam casos excepcionais no contexto da análise de Ramachandran devido às suas características estruturais únicas, que influenciam significativamente a sua flexibilidade conformacional e

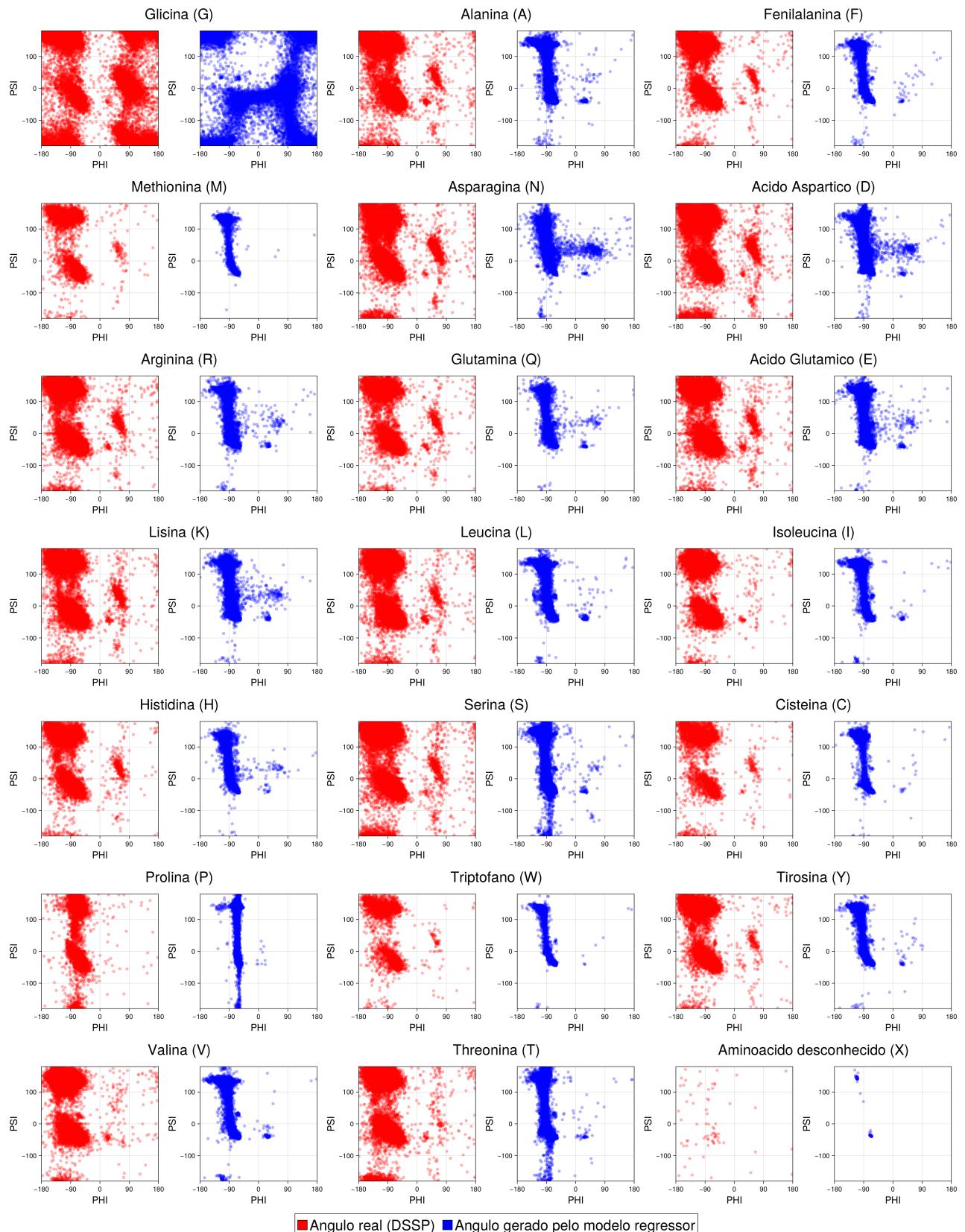


Figura 5.11: Gráfico de Ramachandran para cada um dos 20 aminoácidos naturais e para o aminoácido desconhecido (X), para todas as proteínas do fold 10 do conjunto VF3162.

preferências de ângulos torcionais.

A glicina é o mais simples de todos os aminoácidos, com um átomo de hidrogênio como cadeia lateral. Esta cadeia lateral mínima confere à glicina um grau excepcional de flexibilidade conformacional, tornando-a muito menos impedida estericamente do que outros aminoácidos. Como resultado, no gráfico de Ramachandran, a glicina é caracterizada por uma faixa muito mais ampla de ângulos  $\varphi$  e  $\psi$  permitidos, pois pode adotar conformações que seriam estericamente proibidas para outros aminoácidos, levando a uma ocupação mais extensa tanto nas regiões positivas quanto negativas. A flexibilidade da glicina é crítica em proteínas onde curvas ou curvas fechadas são necessárias, como nas alças que conectam elementos da estrutura secundária ou em dobras compactas de proteínas, onde outros aminoácidos criariam conflitos estéricos. Essas regiões são comumente caracterizadas pelos softwares de referência como regiões de *loop* ou sem estrutura secundária definida, o que reflete no panorama observado anteriormente na figura 5.8.

Na Figura 5.11, o painel da Glicina revela, para o estado de referência, o formato característico e reconhecível do gráfico de Ramachandran para esse resíduo, no qual existe um padrão expansivo de ocupação do espaço  $(\varphi, \psi)$ . Ao examinar o mesmo gráfico, produzido na previsão, notamos uma reprodução de características marcantes e definidoras do gráfico de referência, como uma população específica de pontos nas vizinhanças de  $(\varphi, \psi) = (-180, -180)$ ,  $(+180, +180)$  e  $(+180, -180)$  (a partir do centro), o que indica que o modelo está sendo capaz de explorar esse espaço permitido apenas nesse caso especial. Também notamos positivamente que, mesmo com mais flexibilidade, as previsões continuam evitando as regiões proibidas na proximidade dos pontos  $(0, +180)$  e  $(0, -180)$ , indicando que o modelo aprendeu a explorar o espaço de forma determinística, e não apenas relaxou as distribuições de valores possíveis.

A prolina, por outro lado, é única devido à sua estrutura cíclica, onde a cadeia lateral está ligada tanto ao carbono  $\alpha$  quanto ao nitrogênio do grupo amino, formando um anel rígido. Esta estrutura distinta tem várias implicações, inclusive que a prolina frequentemente “quebra”, ou interrompe, uma sequência estrutural em hélices- $\alpha$  e folhas- $\beta$ , justamente porque sua estrutura rígida é incompatível com os padrões regulares de ligações de hidrogênio dessas estruturas secundárias, sendo mais comum nas extremidades das hélices e nas espiras (do inglês, *coils*) entre os filamentos das folhas- $\beta$ . No entanto, a principal consequência e a severa restrição dos ângulos diédricos do seu *backbone*, levando a um conjunto significativamente limitado de ângulos  $\varphi$  permitidos. No gráfico

de Ramachandran, a prolina apresenta uma distribuição muito restrita, com preferência pela região associada à conformação da hélice de poliprolina (hélice PPII).

Examinando o painel da Prolina na Figura 5.11, notamos que de fato as previsões da TintiNet.jl se concentram conservadoramente em uma faixa bem estreita de ângulos  $\varphi$ , conforme o esperado para esse aminoácido e acompanhando o comportamento do gráfico de referência ao seu lado. Conseguimos também observar um *cluster* bem pronunciado de pontos na vizinhança da região  $(\varphi, \psi) = (-75, 150)$ , que corresponde a conformação PPII, exclusiva e característica desse aminoácido.

Cabe aqui também um comentário retomando as exposições feitas anteriormente sobre as asparaginas e ácidos aspárticos, na Seção 5.3.1. Se recordarmos, acompanhando a convolução do sinal que gerava as probabilidades de cada classe, chegávamos frequentemente aquele estado de “indecisão” da rede entre as estruturas “C” e “E”, onde valores excepcionalmente próximos de probabilidade eram gerados mesmo depois da função de ativação softmax. Na rede de regressão, a função de ativação da camada final é simplesmente uma ativação do tipo identity ( $f(x) = x$ ). O mesmo mecanismo de confluência de sinais conflitantes na cabeça de detecção pode explicar o fato de que *especialmente* os gráficos de Ramachandran das estimativas, para esses dois tipos de aminoácidos, apresentam tantos pontos na região costumeiramente proibida próxima de  $(\varphi, \psi) = (0, 0)$ .

Finalmente, em relação a previsão de área superficial acessível ao solvente (SASA), cabe primeiro aqui um adendo. Evidentemente, e conforme dito na introdução, todas as propriedades unidimensionais figuradas neste trabalho, mesmo sendo computadas aminoácido por aminoácido, refletem relações e imposições físico-químicas que, em última instância, vão refletir o estado tridimensional enovelado da proteína. No entanto, é evidente que, para cada uma dessas propriedades, deve existir um certo “grau de localidade” da informação mínima necessária e suficiente para estimar a propriedade com sucesso, que provavelmente não é igual para cada uma delas. Sendo assim, pelo menos *a priori*, é razoável pensar que a estimativa de SASA tem uma sofisticação talvez superior àquela da própria estrutura secundária; torna-se obrigatório aqui, **necessariamente**, abstrair padrões que *pelo menos indiretamente* reflitam algum tipo de posicionamento relativo entre os aminoácidos, uma ideia que novamente evoca a possibilidade desses padrões de propriedades unidimensionais estarem - *ainda que parcialmente* - contaminados

de contexto tridimensional.

Voltando para a análise estatística propriamente dita, a Figura 5.12 nos permite analisar como o MAE reportado na Figura 5.9 se distribui ao longo dos diferentes tipos de aminoácido. Em relação ao formato das distribuições, podemos perceber que, na realidade, não há muita diferença entre a simetria, o formato e - em diversos casos - sequer na mediana das distribuições. No entanto, há uma diferença notável na **dispersão** dos valores de SASA reais, que abrangem uma faixa muito maior e mais larga de valores do que os valores estimados pelo modelo. ( $\bar{\sigma}_{DSSP} = 50.57 \text{ \AA}^2$  contra  $\bar{\sigma}_{TintiNet.jl} = 34.57 \text{ \AA}^2$ ). Nas situações em que as medianas são significativamente diferentes, a mediana das estimativas para a TintiNet.jl demonstra ser sempre mais conservadora no sentido da mediana geral da população de validação, da mesma forma como relatado nas estimativas de ângulos de diedro. Essa observação não surpreende, dado que ambas as previsões  $((\varphi, \psi)$  e SASA) compartilham um tronco inteiro de processamento de dados que só diverge na última operação da cabeça de detecção. As medianas em si seguem uma ordem de tamanho previsível, com os maiores valores reservados aos maiores aminoácidos que também possuem na sua cadeia lateral átomos capazes de realizar ligações de hidrogênio com a água (ARG, GLU, LYS), ao passo que os menores valores correspondem a cisteína (CIS) e aos pequenos aminoácidos apolares e ramificados (VAL, LEU, ILE)

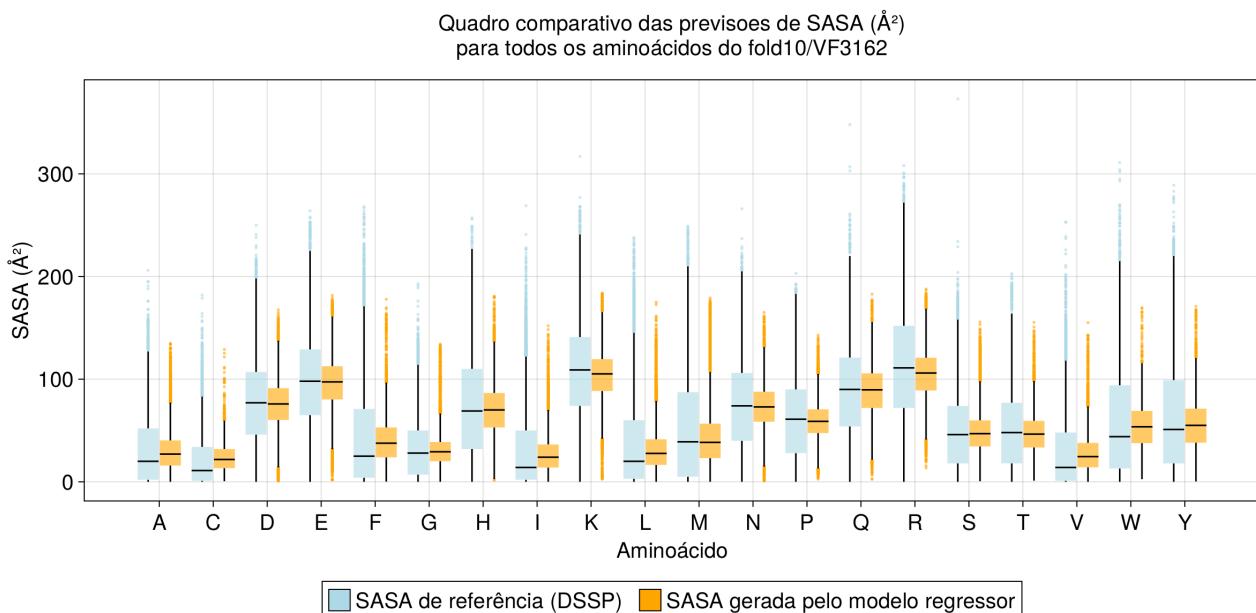


Figura 5.12: *Boxplots* comparando a distribuição de referência e estimada das áreas acessíveis ao solvente, para cada um dos 20 aminoácidos naturais, no fold10/VF3162.

### 5.4.2 *Benchmark* de tempo de execução

*Benchmarks* computacionais foram realizados para avaliar o tempo para gerar previsões de todos os 4 modelos. Nossa hipótese era que a combinação de um *pool* de parâmetros menor, o uso exclusivo de camadas paralelizáveis e a implementação em Julia pura permitiriam ao TintiNet.jl obter uma diminuição no tempo de previsão comparável à diminuição no número de parâmetros. Os resultados do *Benchmark* são mostrados na Figura 5.13.

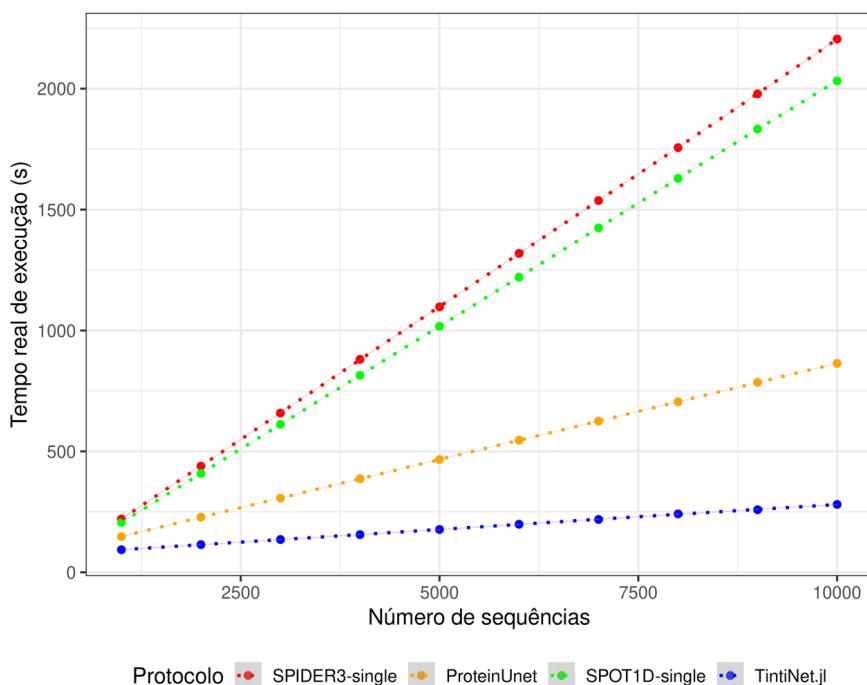


Figura 5.13: *Benchmark* computacional da arquitetura TintiNet.jl comparada aos outros preditores. Tempo real de execução para previsão em função do tamanho da amostra (em número de sequências) para todas as cinco replicatas. As linhas tracejadas são resultados de regressão linear.

Como esperado, em todos os casos, o tempo real decorrido para previsão escala linearmente com a quantidade de sequências apresentadas aos modelos. Realizar a regressão linear nesses tempos de execução em função do número de sequências revelou os tempos de previsão como a inclinação (coeficiente angular), deixando os tempos de aquecimento do modelo como intercepto (coeficiente linear). Esta é uma avaliação mais adequada do desempenho computacional do que o tempo médio de previsão em um determinado conjunto, pois em um cenário de previsão, o modelo é compilado e inicializado apenas uma vez e consumido como serviço para qualquer quantidade arbitrariamente alta de sequências.

A Tabela 5.4 mostra as inclinações dessas regressões juntamente com a quantidade de

Tabela 5.4: Tempo real de execução da previsão de estrutura secundária nos experimentos de *benchmark*

Modelo	Tempo real de previsão (segundos)
TintiNet.jl	0.0208 ± 0.0001
SPIDER3-single	0.2201 ± 0.0003
ProteinUnet	0.0796 ± 0.0001
SPOT1D-single	0.2032 ± 0.0002

parâmetros para cada modelo. Como esperado, os modelos contendo Redes Neurais Recorrentes, SPIDER3-single e SPOT1D-single são responsáveis pelos tempos mais longos, por consequência da natureza intra-serializada das RNNs em cada amostra. O ProteinUnet, que foi construído com convoluções paralelizáveis, está na camada intermediária dos tempos de previsão. Por fim, TintiNet.jl tem o menor tempo decorrido para previsão, com apenas 0,02 segundos por sequência, em média, endossando as observações feitas sobre sua arquitetura leve e o uso, por princípio, de camadas de processamento paralelizáveis.

## 5.5 Interpretação do modelo

A interpretação de modelos seq2seq baseados em Transformer por meio de pontuações de atenção fornece uma janela para o funcionamento interno do modelo, vinculando saídas a segmentos de entrada relevantes. Ao analisar e visualizar meticulosamente essas pontuações, obtemos *insights* sobre o processo de tomada de decisão do modelo, aumentando nossa compreensão de como esse processo pode estar acontecendo. Tendo atingido a primeira parte do objetivo deste trabalho, relacionado com desenvolver, implementar, treinar e avaliar os méritos preditivos da arquitetura enquanto modelo de *Machine Learning*, voltamo-nos portanto para a segunda parte deste trabalho, que é justamente o empreendimento de interrogar e entender o processo de aprendizagem do modelo. Essa seção relata nossas tentativas de tirar proveito da simplicidade da rede neural construída e de sua estrutura arquitetônica propositalmente acessível para identificar maneiras como os parâmetros treinados informam relações entre a sequência e a estrutura das proteínas.

Antes de reportar alguns estudos de caso, vale discutir que a natureza distribuída da informação através de múltiplas cabeças de atenção e camadas Transformer, e a natureza

puramente matemática do modelo significa que pontuações elevadas de atenção **nem sempre** se traduzem diretamente em importância e, *principalmente*, causalidade. Além disso, a complexidade do modelo e o potencial para correlações espúrias necessitam de uma abordagem cautelosa para tirar conclusões dos padrões de atenção. Isso não quer dizer que não existe validade entre os padrões interessantes que serão relatados, apenas que as alegações que faremos aqui devem ser percebidas conforme os autores querem que elas sejam percebidas: observações interessantes, curiosas e que podem inspirar outros experimentos sobre a relação entre a álgebra linear dos computadores e a biologia molecular das sequências de proteínas, sem nenhuma aspiração de afirmar uma relação de causalidade ou dependência.

### 5.5.1 Estudo de caso 1: CATH-1C75A01

O primeiro sistema para estudo de caso é o citocromo C-553 do *Bacillus pasteurii*. A estrutura cristalina desta proteína foi determinada e divulgada com excepcional resolução (0.97 Å) no início dos anos 2000. Trata-se de uma proteína pequena, contendo apenas 71 aminoácidos, pertencente a classe estrutural  $\alpha$ , por conter apenas  $\alpha$ -hélices na forma cristalizada. Além de ser uma proteína simples e fácil de estudar, já foi estudada anteriormente em nosso grupo, em um artigo publicado pelo autor desta Tese, e, portanto, apresenta uma estrutura familiar e fácil de ser analisada. Essa proteína possui uma particularidade muito interessante: por conta de uma região helicoidal de alta flexibilidade, durante o experimento de difração de raios-x, mais de uma conformação da proteína foi observada no cristal, e essa informação está registrada no arquivo mmCIF depositado no PDB. Tínhamos interesse em descobrir de que forma isso se refletiria nos coeficientes intermediários do modelo.

O primeiro passo do estudo de caso foi analisar o estado do tensor de saída depois das quatro camadas InceptiGOR8. A Figura 5.14A mostra um mapa de calor dos sinais de saída, concatenados no eixo vertical para serem representados na forma de matriz. A primeira coisa que estamos procurando observar é a confirmação de que as convoluções com *kernels* sucessivamente maiores estão funcionando da maneira esperada, ou seja, que de baixo para cima, os sinais que começam relativamente fracos e locais (quadrados pequenos e isolados) se tornam cada vez mais intensos, com um campo receptivo maior (linhas longas) e com uma distribuição apreciável (determinada por um sinal médio mais intenso e uma trilha de dispersão a partir desse sinal). Esse

padrão visual pode ser confirmado na Figura 5.14A. Também podemos perceber que, conforme esperado, os sinais que correspondem ao elemento neutro de complemento da sequência (o *gap token*, “-”) são, de forma geral, bastante uniformes, esparsos, bastante similares entre si e diferentes daqueles dos aminoácidos que fazem parte da estrutura primária da proteína. Isto é um resultado de certa forma positivo: indica que a rede está sendo capaz de diferenciar muito bem, entremelio toda a sequência de entrada, o que de fato faz parte da sequência primária da proteína, e o que não faz parte, sem a necessidade de um *token* específico que sinaliza o N-terminal ou C-terminal.

Conforme os sinais numéricos dos aminoácidos percorrem as camadas convolucionais e são contaminados pelos seus vizinhos de curto e médio alcance no campo receptivo, eles começam a se diferenciar dos sinais originais, degenerados para o mesmo aminoácido. Uma vez que as estruturas secundárias se manifestam em fragmentos de sequência contendo algumas unidades de aminoácidos, é razoável supor que existe uma fração significativa do esforço de classificação que já ocorre nas próprias camadas convolucionais, principalmente por conta da sua capacidade de processar conjuntos de sinais espacialmente vizinhos. Para visualizar o efeito parcial da rede convolucional na classificação das estruturas secundárias, realizamos uma Análise Hierárquica de Agrupamentos (HCA) nos tensores de saída exibidos na Figura 5.14A, cuja ordenação deu origem a figura 5.14B.

Uma vez que o alvo CATH-1C75A01 é uma proteína bem simples que praticamente contém apenas  $\alpha$ -hélices e segmentos sem estrutura secundária definida, a Figura 5.14B consegue ilustrar de forma muito visualmente evidente a maneira como a classificação já ocorre nas próprias camadas convolucionais. Conforme pode-se observar, a HCA resulta em três *clusters* muito bem definidos. O *cluster* central claramente corresponde aos elementos neutros de sequência (*gap tokens*), e, portanto, são agrupados conforme esperado devido às particularidades já discutidas dos seus sinais. Os outros dois grupos são bem distintos entre si, e, curiosamente, têm distribuições radicalmente opostas de estrutura secundária prevista. O *cluster* da esquerda contém 40 resíduos, dos quais a maioria representativa (27) não tem estrutura secundária de acordo com o DSSP. Já o *cluster* da direita, muito mais coeso, possui 31 resíduos, dos quais 27 são  $\alpha$ -hélices de acordo com o DSSP.

É surpreendente perceber que, se a classificação ocorresse apenas com a rede convolucional, provavelmente baseada apenas nesses dois *clusters*, assinalando a cada um deles a

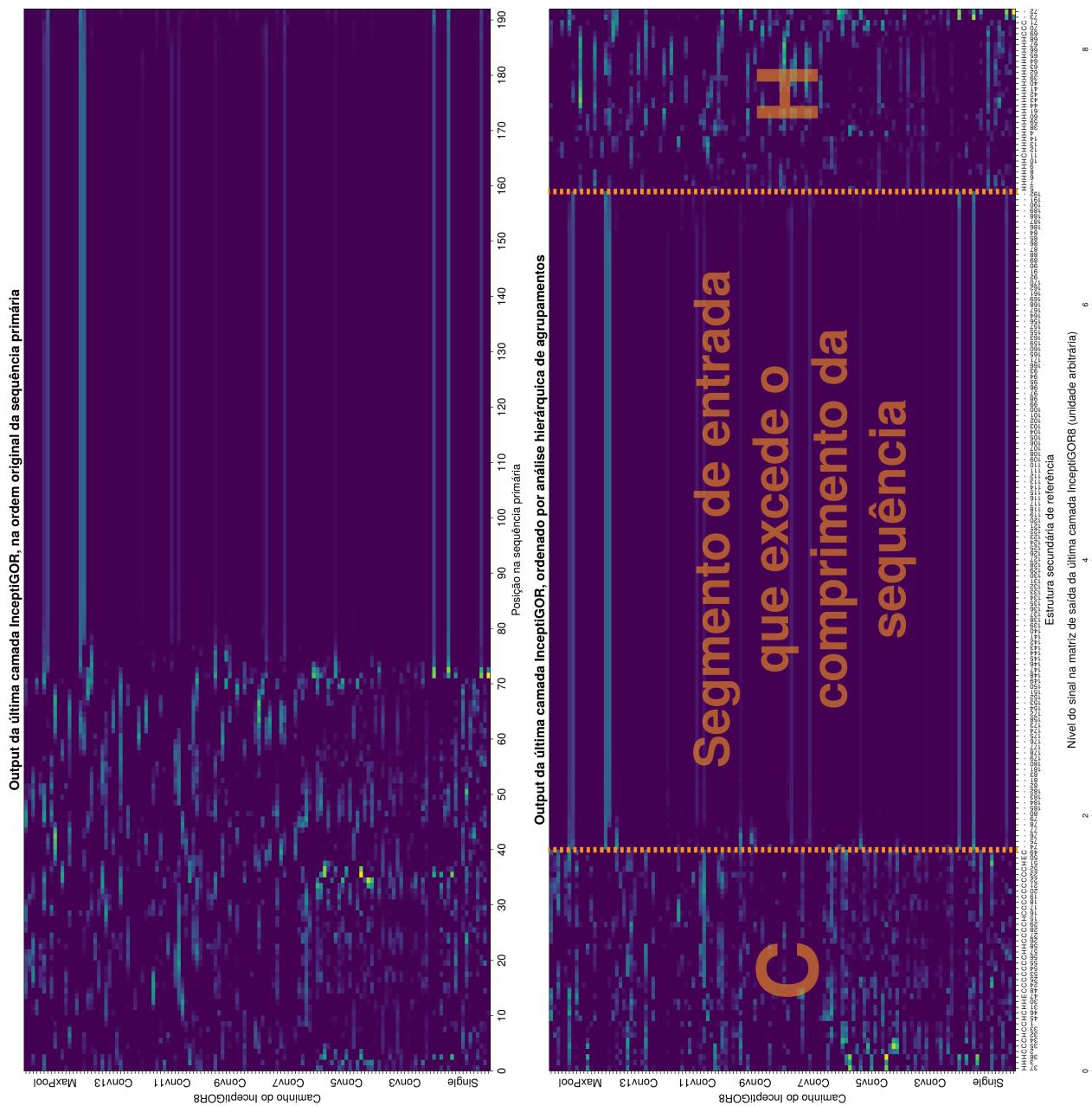


Figura 5.14: Sinal de saída da última camada InceptiGOR8 da parte convolucional do modelo TintiNet.jl para o alvo 5CXOB01. As anotações na cor laranja ilustram as classes que seriam estimadas para cada bloco resultante da análise hierárquica de agrupamentos se a classificação fosse realizada apenas com a saída da camada convolucional.

classe SS3 representativa de sua pertinência, alcançaríamos uma acurácia de classificação de 76% para essa proteína. Esse é apenas um exemplo da quantidade de informação estrutural que os módulos InceptiGOR desenvolvidos neste trabalho estão conseguindo capturar e imprimir ao sinal do *token* de cada aminoácido. Adiante, notaremos que, com a ajuda das camadas BERT e após processamento pela cabeça de detecção, para esse alvo, a acurácia dessa classificação na realidade

superá 80%.

Em relação às informações que podem ser obtidas por meio da interpretação dos pesos de atenção, a Figura 5.15 mostra, primeiramente, uma análise de grande escala com todas as cabeças de previsão em ambas as camadas. Naturalmente, espera-se que, ao mesmo tempo em que cabeças diferentes estejam atentando para pontos diferentes da sequência, posições-chave estejam polarizando a atenção global do modelo, potencialmente em mais de uma cabeça, e possivelmente em mais de uma camada Transformer.

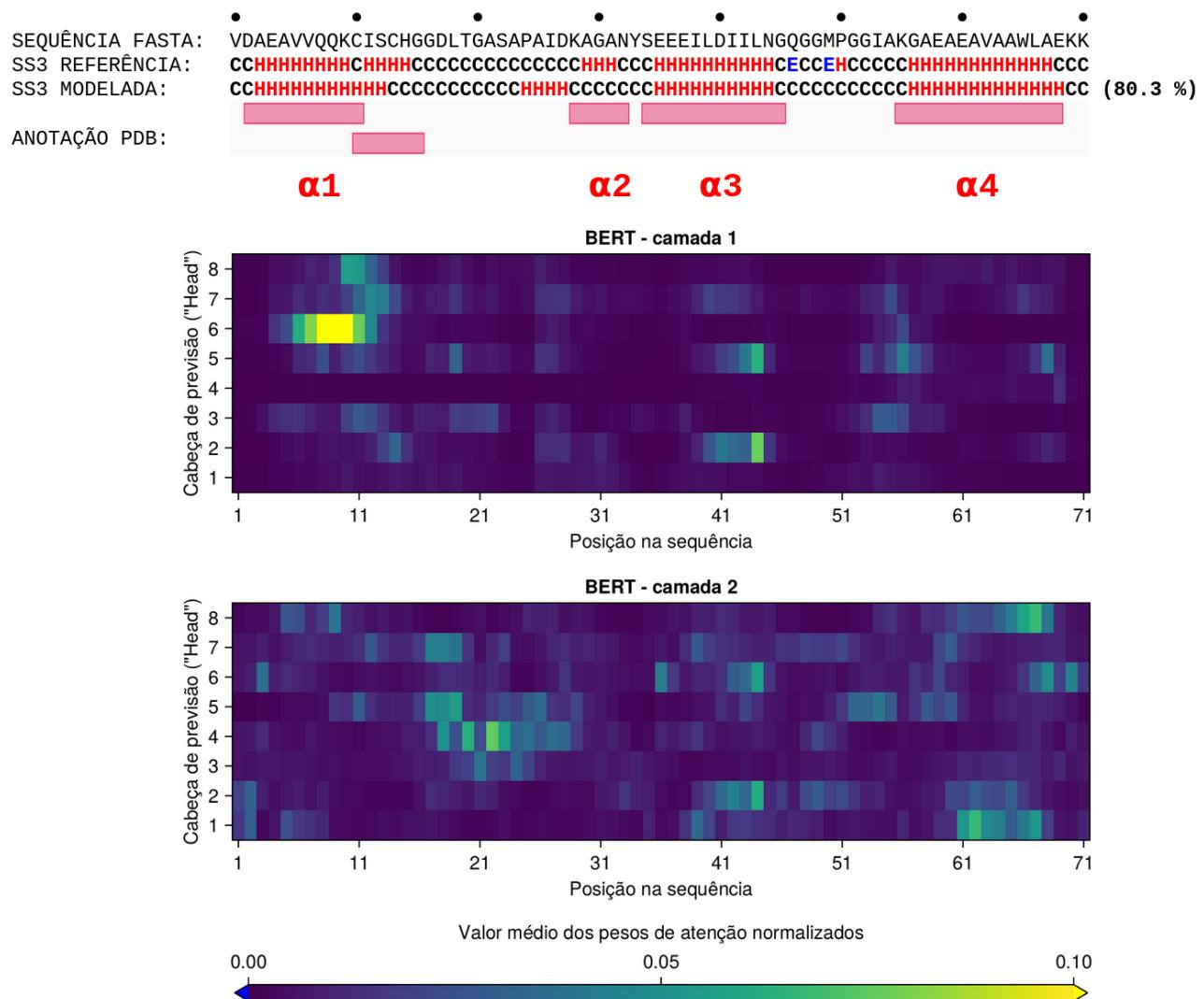


Figura 5.15: Pesos de atenção médios recebidos por cada aminoácido da sequência do alvo CATH-1C75A01 em cada cabeça de detecção das camadas BERT em tempo de inferência de SS3

Uma análise visual da Figura 5.15 revela justamente esse padrão de distribuição dos pesos de atenção, e é especificamente notável uma concentração bastante pronunciada de pesos de atenção

entre os resíduos 7 a 11, sendo o resíduo 10 responsável pelo máximo escore de atenção recebida. Um efeito notável que parece se alinhar com a hipótese de concepção da arquitetura TintiNet.jl, onde a rede Transformer atua como refinadora da rede convolucional, é que muitos dos picos de atenção se alinham com resíduos que se encontram nas extremidades ou ante-extremidades de fragmentos de estrutura secundária. A lógica aqui seria que esses terminais são críticos para a classificação de segmentos inteiros, e têm peso substancial na propensão de sucessivos aminoácidos em adotar determinada estrutura.

Selecionamos a cabeça de atenção 6 da camada Transformer 1 para estudar de forma mais aprofundada, recuperando a matriz de atenção original que deu origem às médias reportadas. Um mapa de calor representando essa matriz é apresentado na Figura 5.16A. O painel revela que, a despeito do resíduo 10 e sua vizinhança serem bastante importantes para quase toda a sequência, são especialmente relevantes no contexto da classificação de um segmento de estrutura secundária entre os resíduos 63 a 71, recebendo atenção máxima do resíduo 67. A priori, além de ser o penúltimo componente da hélice  $\alpha$ 4, não existe um motivo evidente nas estruturas primária ou secundária para justificar que esse especificamente esteja interagindo com o resíduo 10 de forma tão específica.

Traçamos então a hipótese de que os escores de atenção na interação entre os resíduos 10 e 67 (e suas vizinhanças) poderiam estar sinalizando alguma relação numérica entre aqueles resíduos que, em última instância, refletiriam tanto na estrutura secundária quanto na própria estrutura terciária da molécula. Explorando essa hipótese, primeiro verificamos que, de fato, o resíduo 10 estabelece uma bifurcação na anotação de estrutura secundária no próprio PDB. Posteriormente, examinamos manualmente as coordenadas tridimensionais no arquivo mmCIF para essa proteína. Nele, os resíduos 10 e 67 registram a ocorrência de duas conformações (Conforme mostrado no Código 2), dentre as quais especialmente para a LYS10, ocorre uma modificação da interação não-covalente realizada pelo NZ, que em um dos casos realiza uma ligação hidrogênio com o O da ILE54 (sem estrutura secundária definida), e na outra realiza uma ligação hidrogênio com o OE do GLU59 (na hélice  $\alpha$ 4). Uma visualização desses casos está exposta na Figura 5.16B.

Essa observação corrobora a tese de que os escores de atenção nas camadas Transformer, atuando como coeficientes de uma combinação linear no centro de operações de autoatenção, estão de fato mediando a interação entre os sinais numéricos dos aminoácidos em

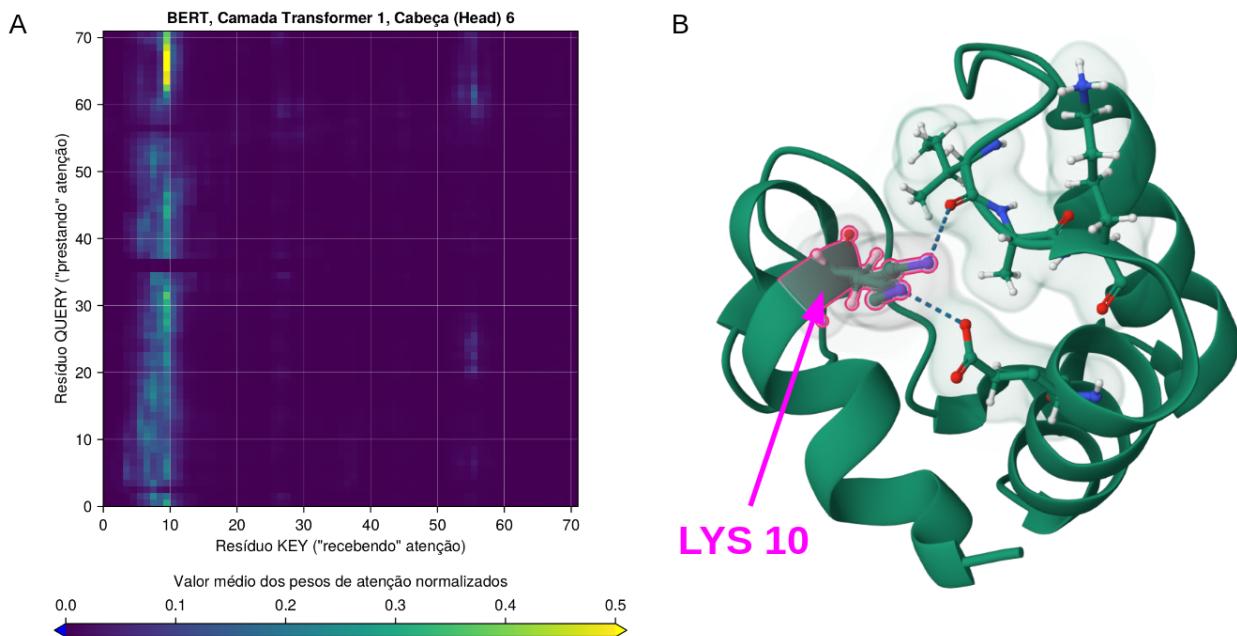


Figura 5.16: (A) Matriz de atenção da cabeça 6 da primeira camada Transformer do componente BERT da TintiNet.jl durante a inferência de estrutura secundária categórica (SS3) para o alvo CATH-1C75A01. (B) Detalhes estruturais do alvo CATH-1C75A01 evidenciando as duas conformações da LYS10. Imagem gerada com Mol\* Viewer [90]

algumas situações em que existe uma explicação estrutural latente por trás dessa interação. Trata-se de uma evidência na direção de demonstrar que o modelo de linguagem está de fato sendo capaz de abstrair a relação sequência-estrutura latente representada no conjunto de treinamento apresentado à rede neural. Mais interessante do que isso, embora *completamente especulativo dentro das limitações e evidências geradas neste trabalho*, foi perceber que - a exemplo dos escores de atenção que conduziram ao exame das diferentes conformações no alvo CATH-1C75A01, existe a *possibilidade* de alguns desses padrões estarem extrapolando o contexto apenas unidimensional, e codificando também, em certo grau, informação sobre a estrutura tridimensional das proteínas.

### 5.5.2 Estudo de caso 2: CATH-5CXOB01

O segundo sistema para estudo de caso é a proteína conhecida como SALBIII, uma epóxido hidrolase-ciclagem que participa da biossíntese da Salinomicina. A estrutura cristalina desta enzima foi determinada e divulgada (com uma resolução menor que 2 Å) no final de 2015 por uma equipe que incluía cientistas do Instituto de Química da UNICAMP. A SALBIII tem uma estrutura de homodímero, e apenas uma das suas cadeias, que possui uma sequência de 134 aminoácidos,

foi objeto de modelagem. Na realidade, a sequência FASTA desse aminoácido, determinada por espectroscopia de massas, possui 148 aminoácidos, mas a estrutura PDB depositada só possui coordenadas para os primeiros 131 deles da cadeia A, e 134 da cadeia B. Portanto, a base CATHS40 tem a estrutura da cadeia B, e foi essa que foi utilizada neste trabalho.

**Código 2.** Algumas linhas extraídas do arquivo mmCIF para o alvo CATH-1C75A01, para os resíduos 10 e 67. Destaque para a coluna 14, que registra a *occupancy* do átomo. Átomos repetidos com números menores que 1.00 indicam mais de uma conformação capturada no cristal.

```

ATOM 139 N N . LYS A 1 10 ? 6.293 18.292 9.227 1.00 9.23 ? 31 LYS A N 1
ATOM 140 C CA . LYS A 1 10 ? 6.095 18.477 7.806 1.00 9.39 ? 31 LYS A CA 1
ATOM 141 C C . LYS A 1 10 ? 7.309 19.011 7.040 1.00 8.72 ? 31 LYS A C 1
ATOM 142 O O . LYS A 1 10 ? 7.298 18.998 5.810 1.00 10.64 ? 31 LYS A O 1
ATOM 143 C CB . LYS A 1 10 ? 4.915 19.453 7.546 1.00 9.87 ? 31 LYS A CB 1
ATOM 144 C CG . LYS A 1 10 ? 3.620 18.901 8.043 1.00 13.47 ? 31 LYS A CG 1
ATOM 145 C CD A LYS A 1 10 ? 2.465 19.738 7.413 0.55 15.53 ? 31 LYS A CD 1
ATOM 146 C CD B LYS A 1 10 ? 2.340 19.745 8.016 0.45 15.79 ? 31 LYS A CD 1
ATOM 147 C CE A LYS A 1 10 ? 1.083 19.108 7.618 0.55 19.18 ? 31 LYS A CE 1
ATOM 148 C CE B LYS A 1 10 ? 1.289 19.010 8.859 0.45 16.59 ? 31 LYS A CE 1
ATOM 149 N NZ A LYS A 1 10 ? 0.046 20.000 7.048 0.55 27.32 ? 31 LYS A NZ 1
ATOM 150 N NZ B LYS A 1 10 ? -0.088 19.307 8.388 0.45 15.09 ? 31 LYS A NZ 1
ATOM 151 H H . LYS A 1 10 ? 6.449 18.964 9.740 1.00 11.07 ? 31 LYS A H 1
ATOM 152 H HA . LYS A 1 10 ? 5.856 17.608 7.421 1.00 11.27 ? 31 LYS A HA 1
ATOM 153 H HB2 . LYS A 1 10 ? 5.095 20.297 7.989 1.00 11.84 ? 31 LYS A HB2 1
ATOM 154 H HB3 . LYS A 1 10 ? 4.845 19.624 6.594 1.00 11.84 ? 31 LYS A HB3 1
ATOM 155 H HG3 . LYS A 1 10 ? 3.583 18.958 9.011 1.00 16.16 ? 31 LYS A HG3 1
...
ATOM 906 C CB A LEU A 1 67 ? 9.852 30.557 9.574 0.58 7.21 ? 88 LEU A CB 1
ATOM 907 C CB B LEU A 1 67 ? 9.702 30.490 9.619 0.42 7.95 ? 88 LEU A CB 1
ATOM 908 C CG A LEU A 1 67 ? 9.813 29.047 9.670 0.58 8.31 ? 88 LEU A CG 1
ATOM 909 C CG B LEU A 1 67 ? 10.971 29.982 8.986 0.42 9.81 ? 88 LEU A CG 1
ATOM 910 C CD1 A LEU A 1 67 ? 9.578 28.514 8.249 0.58 9.18 ? 88 LEU A CD1 1
ATOM 911 C CD1 B LEU A 1 67 ? 11.829 29.107 9.891 0.42 11.31 ? 88 LEU A CD1 1
ATOM 912 C CD2 A LEU A 1 67 ? 11.120 28.495 10.234 0.58 10.62 ? 88 LEU A CD2 1
ATOM 913 C CD2 B LEU A 1 67 ? 10.661 29.159 7.714 0.42 10.24 ? 88 LEU A CD2 1
ATOM 914 H H . LEU A 1 67 ? 7.931 30.972 11.238 1.00 8.53 ? 88 LEU A H 1
ATOM 915 H HA A LEU A 1 67 ? 10.570 30.856 11.476 1.00 8.95 ? 88 LEU A HA 1

```

Essa proteína também já figurou em alguns trabalhos anteriores do nosso grupo, então já dispúnhamos de familiaridade com a sequência e as propriedades estruturais dessa proteína que facilitaram a visualização e discussão dos resultados. Além disso, trata-se de uma proteína que pertence à classe estrutural  $\alpha+\beta$ , que contém várias alternâncias de segmentos de  $\alpha$ -hélices e fitas- $\beta$  ou folhas- $\beta$ .

Assim como no caso anterior, o primeiro passo do estudo de caso foi analisar o estado do tensor de saída depois das quatro camadas InceptiGOR8. A Figura 5.17A mostra o mapa de calor resultante dessa análise.

Assim como no caso anterior, observamos a formação de padrões sucessivos no desenho dos sinais, devido a aplicação dos *kernels* cada vez maiores nos caminhos paralelos das operações

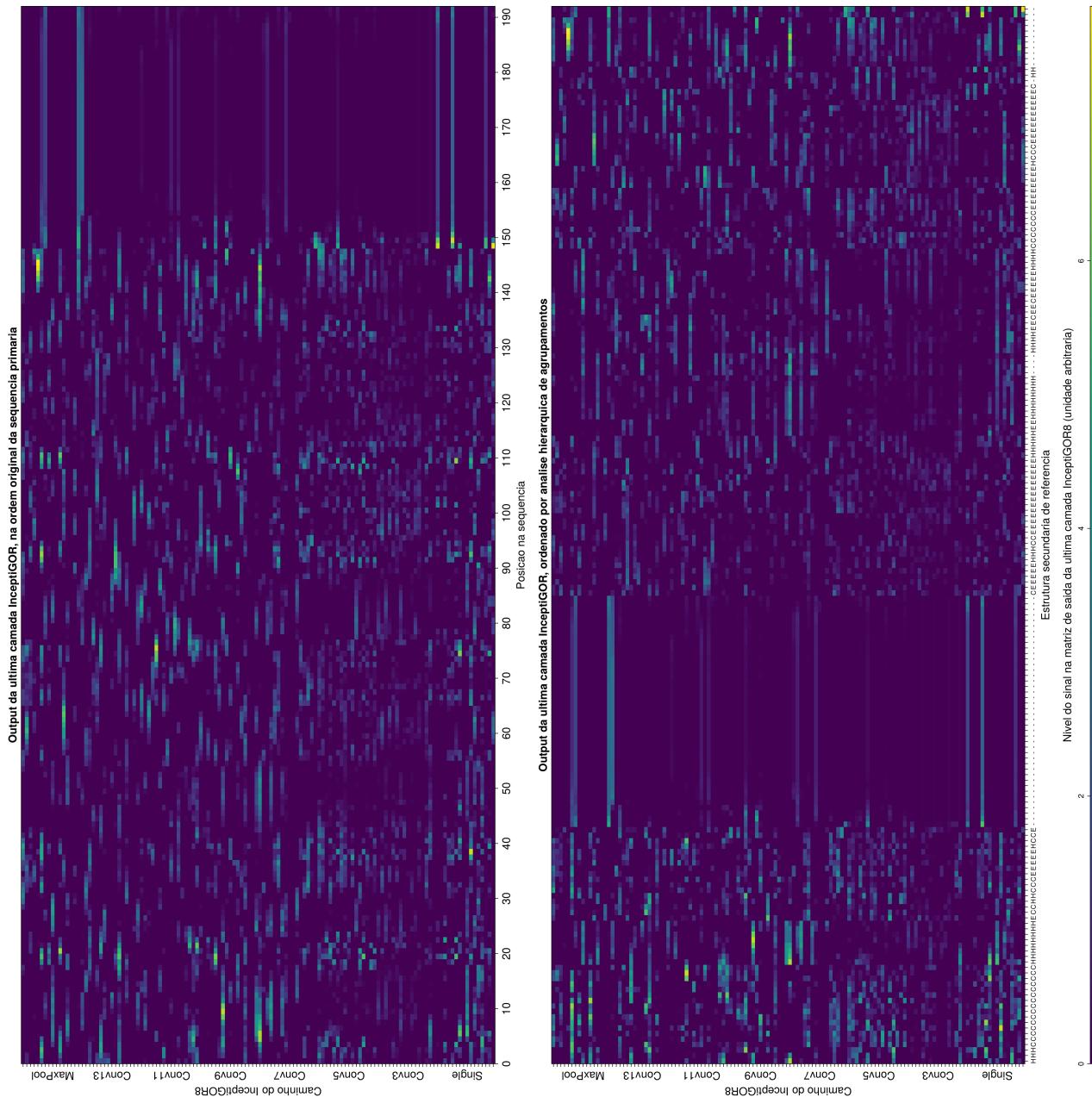


Figura 5.17: Sinal de saída da última camada InceptiGOR8 da parte convolucional do modelo TintiNet.jl para o alvo 5CXOB01.

convolucionais. Nota-se novamente que alguns sinais já se sobressaem na sequência antes mesmo do tensor adentrar a camada Transformer, e que, de forma similar, aminoácidos diferentes possuem sinais diferentes, pois já existe um certo grau de informação estrutural parcialmente codificada no vetor numérico de cada aminoácido. Ao repetir a análise de HCA para modificar a ordem do eixo horizontal, construímos o gráfico que pode ser apreciado na Figura 5.17B. Mesmo sendo um alvo muito mais complexo que a CATH-1C75A01, também conseguimos perceber aqui a formação de

padrões claros nos *clusters* obtidos. Por exemplo, destacamos que muitos dos resíduos sem estrutura secundária definida, e que não são todos vizinhos na sequência primária, mas são todos membros de pequenas voltas entre segmentos da estrutura de barril-beta da SALBIII, se agrupam em um pequeno *cluster* a esquerda do gráfico (20-22, 75-77, 92-94, 109-111), e há agrupamentos bem coesos de resíduos que já alcançariam uma classificação de alto desempenho mesmo se a decisão ocorresse logo após a saída das camadas InceptiGOR8.

Procedendo finalmente à análise dos escores de atenção das camadas Transformer, a investigação de alto nível contendo as médias das matrizes de atenção **recebida** por cada posição da sequência está exibida na Figura 5.18, à qual foi também adicionada a sequência FASTA da proteína, a estrutura secundária SS3 de referência e a estimativa do modelo. A partir de um exame visual da Figura 5.18, selecionamos 2 cabeças de atenção para fazer uma análise completa da matriz de atenção, objetivando entender se a atenção observada está sendo generalizada ao longo da sequência, ou local, e, no segundo caso, a que nível de distanciamento na estrutura primária as interações estão ocorrendo. Conforme destaque na Figura 5.19, **selecionamos as cabeças 2 e 6 da camada Transformer 1**.

Produzimos a Figura 5.19 com os mapas de calor das matrizes de atenção das cabeças selecionadas, e adicionamos algumas poses ilustrativas da estrutura cristalográfica da proteína para auxiliar a visualização dos resultados da análise. Primeiramente, analisando o painel 5.19A, percebemos que o centro de interação mais intenso observado anteriormente na Figura 5.18 é na verdade uma interação de muito curto alcance na diagonal principal do mapa de calor, entre os primeiros resíduos próximos ao N-terminal. Esse tipo de interação é difícil de interpretar, mas apresenta um resultado valioso para a compreensão da arquitetura concebida como um todo: embora seja verdade que a maioria das relações observadas na rede Transformer sejam de médio e longo alcance, podem existir casos em que a rede convolucional não vai desonerar completamente o componente de autoatenção de abstrair informação de curto alcance. Essas situações não são problemas para a rede, por serem potenciais “desvios” do design original; nem prejudicam o cômputo da estrutura secundária. É da natureza da camada Transformer poder capturar relações do contexto global, sejam elas distantes ou próximas, se necessário for, e se houver algum fenômeno relevante para ser contabilizado que não foi totalmente processado pelas camadas convolucionais.

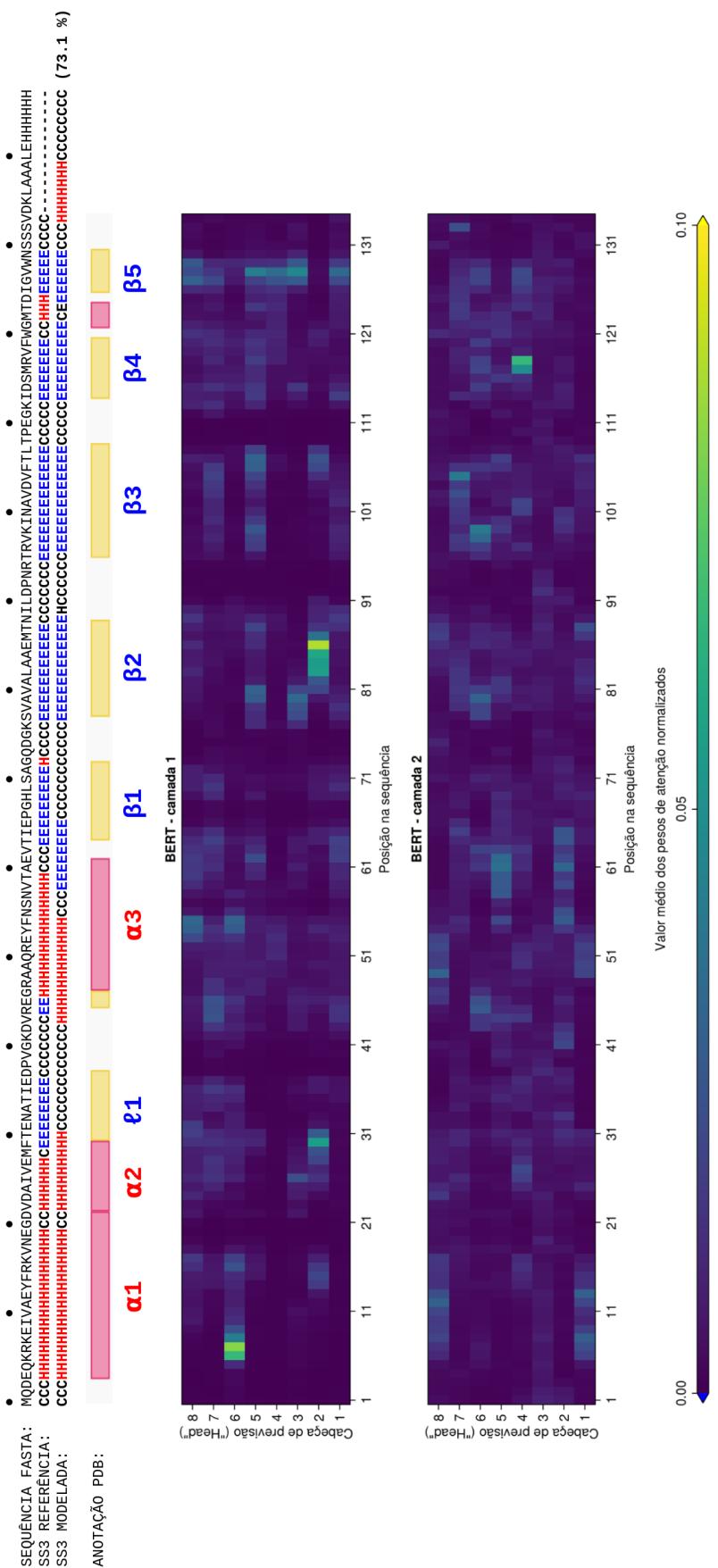


Figura 5.18: Pesos de atenção médios recebidos por cada aminoácido da sequência do alvo CATH-5CXOB01 em cada cabeça de detecção das camadas BERT em tempo de inferência de SS3

Fora da diagonal principal na Figura 5.18A, o segundo maior sinal ocorre entre os resíduos 14 e 7. Um exame das propriedades de sequência e do estado de referência SS3 não revelou nenhum motivo razoável para essa relação, que inclusive pareceu inicialmente curiosa, uma vez que ocorreu entre dois resíduos de polaridades diferentes, ambos em posições internas no mesmo segmento de estrutura secundária. Procedemos novamente então, assim como no alvo CATH-1C75A01, a um exame da estrutura terciária.

Esse exame da estrutura terciária revelou que o resíduo ARG7, apesar de ser uma arginina próxima ao N-terminal em conformação helical, apresenta uma área superficial acessível ao solvente surpreendentemente baixa para sua identidade e posição na sequência. Procurando entender as interações não-covalentes em questão, entendemos que na realidade essa ARG7 em específico está envolvida em diversas ligações de hidrogênio que só são possíveis com a manutenção do *motif* de barril- $\beta$  da SALBIII. Dessa maneira, essa ARG7 estabiliza e é estabilizada por essa interface entre o N-terminal e uma região de volta (*turn*) contendo resíduos distantes mais de 70 aminoácidos na sequência primária (Figura 5.18C). Imaginamos que, se a TYR14 estava dando muito peso para o sinal da ARG7 na matriz de atenção, deveria também existir um motivo estrutural por trás disso, que será retomado adiante. Ainda no painel A, destacamos também uma aparente interação entre os resíduos 18/19 e 54.

Em relação a cabeça de previsão 2 (Figura 5.18B), conseguimos notar uma grande faixa de aminoácidos dando altos escores de atenção para o resíduo 30 (inclusive TYR14, responsável pelo segundo maior escore - o escore máximo foi recebido de VAL 18). O resíduo 30 é uma fenilalanina, posicionado exatamente em uma faixa de transição entre uma alfa-hélice e uma fita-beta, o que a princípio justificaria sua importância, e o motivo pelo qual recebe um escore de importância tão alto.

Essas relações aparentemente esparsas começam a ficar mais interessantes conforme vamos combinando pequenas observações sobre interações entre pares de aminoácidos. No caso da SALBIII, ficamos surpresos ao perceber que, na realidade, os escores de atenção mais notáveis estavam, indiretamente, descrevendo diversas interações entre resíduos apolares (sobretudo aromáticos) no núcleo hidrofóbico da molécula (Figura 5.18D). A esse núcleo, pertencem os já mencionados TYR14, PHE30 e VAL18, além de TYR54, PHE55 e PHE106, todos com escores de atenção encadeados entre as diversas matrizes de atenção) Ainda mais notável foi perceber que, de

todos esses aminoácidos, pelo menos dois deles apontados por escores de atenção fazem parte do sítio catalítico da enzima (TYR14 e TYR54).

Por fim, ainda na cabeça de previsão 2, percebemos uma concentração de escores de atenção entre os resíduos 80 e 90, seguindo uma espécie de padrão “ziguezague”. Esse desenho pode estar indicando o uso de informação local para a classificação de fitas- $\beta$ . No entanto, chama atenção a intensidade da atenção recebida pelos resíduos 86 e 87, principalmente em relação a maneira como o sinal é distribuído. GLU86 recebe a maior intensidade, mas mais difusa, enquanto MET87 notavelmente recebe os dois escores máximos, embora na figura com as médias reportadas termine com um valor agregado inferior ao de GLU86. Curiosamente, o PDB possui uma anotação especificamente no resíduo 87 para o qual existe uma previsão de sítio de ligação (*binding site*).

O caso CATH-5CXOB01, uma proteína maior e mais complexa, apresentou padrões menos claros e menos separáveis que o primeiro caso, CATH-1C75A01. No entanto, ambos os casos produziram exemplos em que modelos treinados em uma tarefa de abstrair propriedades *unidimensionais* geraram escores de atenção nas camadas *Transformer* que, quando adequadamente extraídos e interpretados, sempre apontaram para posições da sequência que tinham alguma relevância na estrutura *tridimensional*. Evidentemente, é fundamental manter a expectativa da extensão dessa alegação, e que explorá-la ou demonstrá-la do ponto de vista formal não é um objetivo deste trabalho; buscou-se aqui apenas uma análise exploratória, uma evidência potencialmente geradora de futuras hipóteses, e uma oportunidade de utilizar modelos de IA para **iluminar** esse processo.

### 5.5.3 Observações gerais sobre os estudos de caso

Neste estudo, utilizamos um modelo seq2seq com mecanismos de auto-atenção para traduzir sequências de proteínas em suas estruturas secundárias correspondentes. A análise dos escores de atenção revelou um foco pronunciado nos terminais de regiões com estruturas secundárias definidas e centros de interações de médio e longo alcance dentro da estrutura terciária, incluindo em alguns casos a concentração de escores de atenção em resíduos pertencentes a sítios ativos ou suas vizinhanças diretas. Essas descobertas oferecem *insights* intrigantes sobre a dinâmica de aprendizagem do modelo e o potencial significado biológico desses fragmentos de sequência no dobramento e função de proteínas.

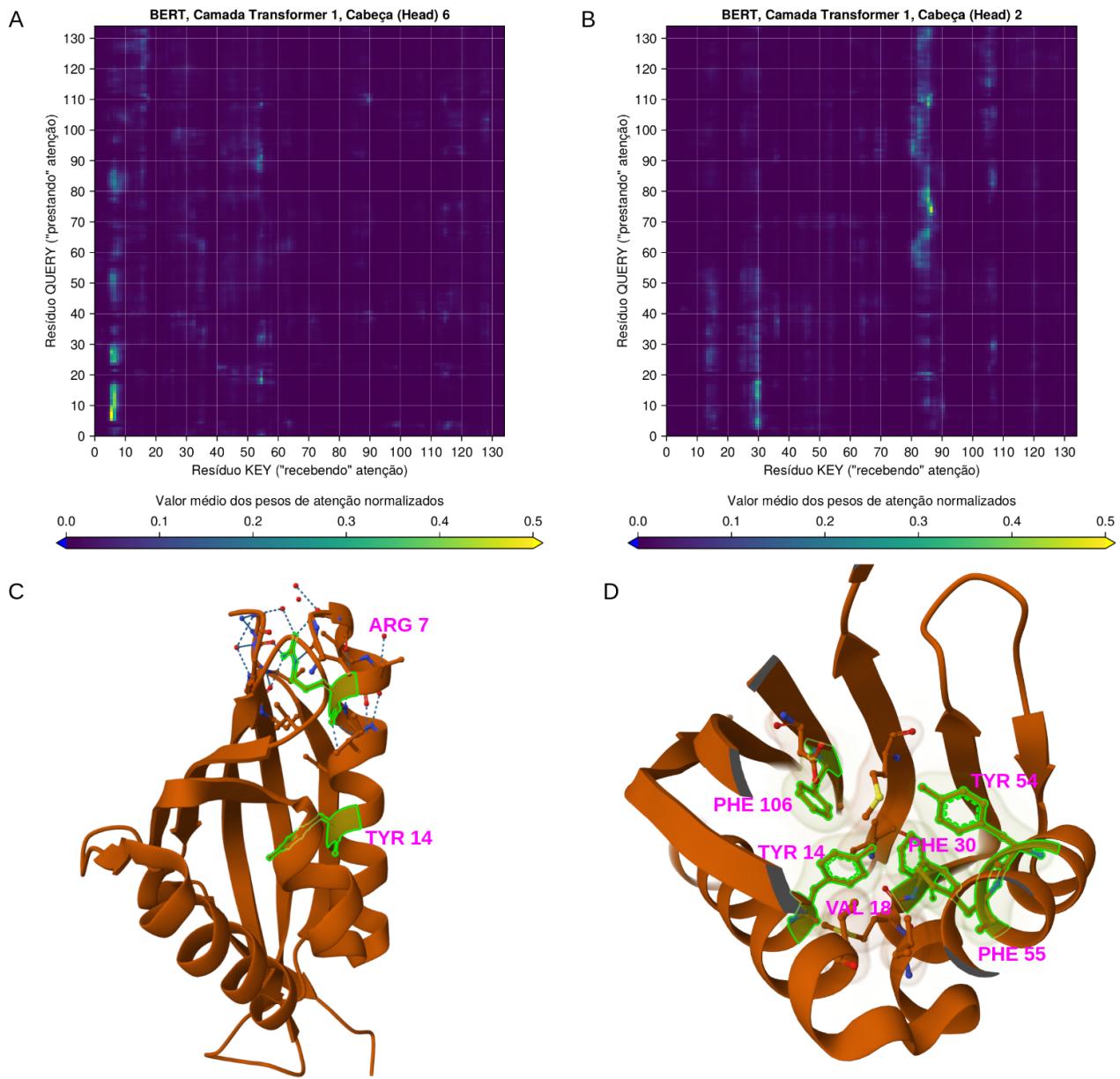


Figura 5.19: Matrizes de atenção das cabeças 6 (A) e 2 (B) da primeira camada Transformer do componente BERT da TintiNet.jl durante a inferência de estrutura secundária categórica (SS3) para o alvo CATH-5CXOB01. (C) Detalhe estrutural da interação entre ARG7 e outros resíduos com grande grau de separação na sequência primária. (D) Detalhe do núcleo hidrofóbico do alvo CATH-5CXOB01 e alguns de seus componentes com cadeias laterais aromáticas. Imagem gerada com Mol\* Viewer [90]

A atenção observada em relação aos terminais das regiões da estrutura secundária é especialmente importante. Estudos de propensão [88] já indicavam que a periodicidade da propensão dos aminoácidos a aparecer em certas regiões dos fragmentos de estrutura secundária era sensível às extremidades, e é um consenso entre a comunidade científica da previsão de estrutura secundária

que acertar previsões no centro dos fragmentos é mais fácil que nos terminais [91]. No entanto, esses fatores são consequências de um fenômeno causal talvez menos interessante, mas mais fatal: para que a cabeça de detecção após a camada BERT, que é do tipo convolucional, modifique a classificação gerada a partir de um terminal de estrutura secundária, alterações agudas e sustentadas nos sinais de saída precisam ser realizadas a partir de determinados pontos da sequência para realmente afetar a distribuição de probabilidades na ponta do modelo. É interessante, no entanto, observar isso acontecendo na prática com tanta transparência, principalmente nos alvos menores, a exemplo do CATH-1C75A01.

Muito mais intrigante é a atenção que emerge nos centros de interações de longo alcance, *sobretudo* nos casos em que se alinha com locais em que são mediadas por outras moléculas ligantes. Essas observações não só se alinham com o conhecimento geral e estabelecido de que contatos de longo alcance são fundamentais para o estabelecimento da estrutura protéica (inclusive secundária), mas mais que isso, **podem estar sugerindo uma aprendizagem implícita do contexto terciário, mesmo quando o treinamento ocorre principalmente em traduções de estruturas de sequência para secundária.**

As implicações dos nossos desenvolvimentos podem ir além da finalidade original modelo computacional, sugerindo potencial relevância biológica. Por exemplo, o foco apresentado do mecanismo de atenção nos centros de interação pode sugerir pontos de partida para estudos do papel crítico de determinados resíduos na regulação alostérica, principalmente nos casos onde os membros dos sítios se encontram a grandes separações na estrutura primária. Esta visão poderia ser aproveitada em estudos futuros para identificar potenciais locais alostéricos ou para projetar proteínas com propriedades dinâmicas desejadas.

Em retrospecto, nossa análise dos escores de atenção em um modelo seq2seq para estimativa de propriedades unidimensionais para sequências proteicas revelou padrões que ressoam com princípios fundamentais na estrutura e dinâmica de proteínas. Estas descobertas não só esclarecem os mecanismos de aprendizagem do modelo, mas também - *de forma modesta, sem pretensão e apenas como uma ferramenta exploratória* - podem ajudar a informar hipóteses sobre a potencial importância biológica dos resíduos destacados.

## 5.6 Limitações

Fundamentalmente, a limitação mais imediata do modelo TintiNet.jl, que se aplica tanto ao modelo classificador quanto ao modelo regressor, certamente é a restrição da sequência de entrada ao tamanho máximo de 192 aminoácidos a partir do N-terminal. É importante mencionar aqui que existem maneiras bem documentadas de se sobrepor a essa limitação sem necessariamente aumentar o número de parâmetros do modelo ou, a memória alocada em GPU, supondo uma disponibilidade muito superior de tempo computacional. Uma possível maneira [92] de fazê-lo seria criar uma espécie de aminoácido artificial que sinalizaria ao modelo o “*inicio da proteína*” e o “*final da proteína*” na sequência de *input*. Ele seria um caractere da sequência assim como os demais aminoácidos, e o modelo naturalmente aprenderia que trata-se de uma extremidade. Depois de adicionar esse aminoácido artificial aos terminais da sequência, o próximo passo seria aplicar logo no início uma codificação posicional. Posteriormente, dividir as sequências em conjuntos de treino e validação e, ao invés de utilizar as sequências inteiras, amostrar e reamostrar diversos fragmentos de igual tamanho máximo de todas as possíveis sequências - grandes e pequenas, contendo ou não terminais, no inicio ou no final, e controlando diversos parâmetros dessa amostragem. O número de amostras, ao final, seria muito superior ao que foi utilizado neste projeto, o que inviabilizaria o desenvolvimento da metodologia dentro das condições em que ocorreram aqui, mas seria eventualmente possível.

Em relação especificamente ao modelo de classificação, a TintiNet.jl avança sobre, mas não soluciona definitivamente a persistente dificuldade dos estimadores de estrutura secundária baseados em sequência única de classificar adequadamente as estruturas do tipo fita- $\beta$ . De fato, retomando as figuras de mérito da classificação, especificamente na Figura 5.3, há 72 domínios de proteínas, correspondendo a aproximadamente 2.2% do conjunto VF3162, para as quais a previsão de SS3 erra mais de 50% das classificações. Uma inspeção mais minuciosa dos casos em que o modelo erra mais revela que tratam-se de domínios quase exclusivamente de classe  $\beta$ , ou com grande preponderância deste tipo de estrutura secundária na estrutura cristalográfica.

Além disso, a rede sofre dos problemas naturais de viés de população naturais na maioria dos modelos de *Machine Learning*. Inclusive, um exame minucioso das figuras 5.7 e 5.8 mostra que existe uma correlação muito grande entre a qualidade da previsão de determinada estrutura para determinado aminoácido, e a abundância dessa estrutura para esse aminoácido. Mesmo assim, certamente as figuras de mérito superam muito o que seria esperado num caso em que esse viés

fosse grave. Existem grupos de pesquisa contemporâneos empreendendo maneiras de corrigir esses tipos de desbalanceamento entre classes, usando técnicas como *contrastive learning* [93], mas essas ainda são técnicas complexas que não tivemos recursos para aprender, desenvolver e implementar dentro do escopo deste trabalho.

Por fim, em relação à interpretação do modelo, é importante observar as limitações da interpretação dos escores de atenção. Primeiro, vale ressaltar que o objetivo do trabalho não era criar uma “IA interpretável” no sentido técnico da palavra, mas *aproveitar* a transparência do modelo para ensaiar pequenos empreendimentos de interpretação. Segundo, embora tenham aqui fornecido uma janela para o processo de tomada de decisão do modelo, os padrões de atenção não equivalem necessariamente à relevância biológica e devem ser interpretados com cautela. Trabalhos futuros, se existirem, deverão persistir e se aprofundar no objetivo de validar o significado estrutural destas regiões focadas na atenção, inclusive se possível através de colaborações com laboratórios que permitam estudos experimentais, ou possivelmente explorar a integração dos princípios da biologia estrutural na arquitetura do modelo para melhorar a sua interpretabilidade e precisão.

## Capítulo 6

# Considerações Finais

Neste trabalho, projetamos e implementamos uma arquitetura denominada TintiNet.jl, uma prova de conceito para minificação de modelos de linguagem para inferência de estrutura de proteínas, que empregou uma combinação de convoluções e Transformers para prever propriedades estruturais de sequências de proteínas independente de informação de homologia. Nossa modelo, desenvolvido na linguagem Julia e avaliado em conjuntos de sequências que não foram visitadas durante o treinamento (VF3162), alcançou as maiores métricas de classificação - Q3 72.69% e SOV 69.32 -, mantendo-se competitivo em termos de métricas de regressão - cMAE do ângulo PHI 24.28°; cMAE do ângulo PSI 42.69°; MAE da acessibilidade ao solvente 28.83 Å<sup>2</sup>.

Como um modelo de sequência única, seu desempenho é independente do comprimento da sequência ou da quantidade de informações evolutivas, e experimentos *de benchmark* computacional mostraram que a aplicação de alguns princípios de design, como evitar RNNs e a economia computacional do conjunto de parâmetros, permitiu que TintiNet.jl produzisse as previsões mais rápidas por sequência de entrada.

### 6.1 Perspectivas

O desenvolvimento e a avaliação do TintiNet.jl representam um avanço significativo na difusão do acesso a modelagem com alto desempenho da estrutura de proteínas, destacando a eficácia de uma abordagem que combina convoluções e Transformers para predizer propriedades estruturais 1D de sequências de proteínas. Os resultados deste trabalho, naturalmente, deixam algumas questões perspectivas que podem gerar novos caminhos para a exploração de modelos de

linguagem minificados e interpretáveis para bioinformática estrutural.

Do ponto de vista mais computacional, apesar dos resultados promissores alcançados pela TintiNet.jl, sempre há espaço para refinamentos adicionais na arquitetura da rede neural. Investigar diferentes configurações de camadas, funções de ativação e outros mecanismos de atenção pode levar a melhorias no desempenho preditivo, na eficiência computacional e na interpretabilidade da rede. Especialmente, com o desenvolvimento recente dos chamados explicadores contrafactual, a interpretação de redes neurais está cada vez mais paramentada de métodos ortogonais para acessar a informação entre as camadas.

Na interface entre as diversas ciências naturais, além de predizer propriedades estruturais de proteínas, poder-se-ia considerar a aplicação da arquitetura TintiNet.jl em outros domínios biomoleculares, como sequências de ácidos nucleicos ou estruturas de carboidratos, podendo ampliar significativamente o escopo e a utilidade do modelo. Além da predição de propriedades estruturais, a TintiNet.jl poderia também ser adaptada para tarefas de design de proteínas, se fosse pré-treinada em uma tarefa de linguagem mascarada (MLT). Investigar como a arquitetura pode ser modificada para suportar tais tarefas pode ser uma área de pesquisa empolgante.

Em resumo, a TintiNet.jl representa uma base sólida para futuras pesquisas na interseção de aprendizado de máquina e biologia estrutural. Ao continuar a explorar essas perspectivas e colaborar com especialistas em biologia e informática, podemos continuar a impulsionar os limites do conhecimento e criar ferramentas mais poderosas e igualmente acessíveis para entender e prever estruturas biomoleculares.

## Bibliografia

- [1] "So much more to know". eng. Em: *Science (New York, N.Y.)* 309.5731 (jul. de 2005), pp. 78–102. ISSN: 1095-9203. DOI: 10.1126/science.309.5731.78b.
- [2] P. Atkins e L. Jones. *Chemical Principles*. W. H. Freeman, 2008. ISBN: 9781429209656. URL: <https://books.google.com/books?id=9VIVAQAAIAAJ>.
- [3] David L Nelson e Michael M Cox. *Lehninger Principles of Biochemistry, Fourth Edition*. Ed. por Freeman. Fourth Edition. 2004.
- [4] D. Voet e J.G. Voet. *Biochemistry, 4th Edition*. John Wiley & Sons, 2010. ISBN: 9781118139936. URL: <https://books.google.com.br/books?id=ne0bAAAAQBAJ>.
- [5] Cyrus Levinthal. "How to fold graciously". Em: *Mossbauer spectroscopy in biological systems* 67 (1969), pp. 22–24.
- [6] Leandro Martínez. "Introducing the Levinthal's Protein Folding Paradox and Its Solution". Em: *Journal of Chemical Education* 91 (nov. de 2014), pp. 1918–1923. DOI: 10.1021/ed300302h.
- [7] Cyrus Levinthal. "Are there pathways for protein folding?" Em: *Journal de Chimie Physique* 65 (1968), pp. 44–45. ISSN: 0021-7689. DOI: 10.1051/jcp/1968650044.
- [8] Christian B. Anfinsen. "Principles that govern the folding of protein chains." Em: *Science (New York, N.Y.)* 181.4096 (1973), pp. 223–30. ISSN: 0036-8075. DOI: 10.1126/SCIENCE.181.4096.223.
- [9] Kevin A. Dill. "Additivity principles in biochemistry." Em: *The Journal of biological chemistry* 272.2 (1997), pp. 701–4. ISSN: 0021-9258. DOI: 10.1074/JBC.272.2.701.
- [10] Ken A. Dill e Justin L. MacCallum. "The protein-folding problem, 50 years on". eng. Em: *Science (New York, N.Y.)* 338.6110 (nov. de 2012), pp. 1042–1046. ISSN: 1095-9203. DOI: 10.1126/science.1219021.

- [11] John C. Kendrew et al. "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis". Em: *Nature* 181.4610 (1958), pp. 662–666. DOI: 10.1038/181662a0.
- [12] Soraya de Chadarevian. "John Kendrew and myoglobin: Protein structure determination in the 1950s". en. Em: *Protein Science : A Publication of the Protein Society* 27.6 (jun. de 2018). Publisher: Wiley-Blackwell, p. 1136. DOI: 10.1002/pro.3417. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5980623/> (acesso em 28/02/2024).
- [13] RCSB Protein Data Bank. *PDB Statistics: Overall Growth of Released Structures Per Year*. en-US. URL: <https://www.rcsb.org/stats/growth/growth-released-structures> (acesso em 28/02/2024).
- [14] Alexander Wlodawer et al. "Protein crystallography for aspiring crystallographers or how to avoid pitfalls and traps in macromolecular structure determination". en. Em: *The FEBS Journal* 280.22 (2013). \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/febs.12495>, pp. 5705–5736. ISSN: 1742-4658. DOI: 10.1111/febs.12495. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/febs.12495> (acesso em 28/02/2024).
- [15] Michael Levitt. "A simplified representation of protein conformations for rapid simulation of protein folding". Em: *Journal of Molecular Biology* 104.1 (jun. de 1976), pp. 59–107. ISSN: 0022-2836. DOI: 10.1016/0022-2836(76)90004-8. URL: <https://www.sciencedirect.com/science/article/pii/0022283676900048> (acesso em 29/02/2024).
- [16] A. Warshel e M. Levitt. "Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme". Em: *Journal of Molecular Biology* 103.2 (mai. de 1976), pp. 227–249. ISSN: 0022-2836. DOI: 10.1016/0022-2836(76)90311-9. URL: <https://www.sciencedirect.com/science/article/pii/0022283676903119> (acesso em 29/02/2024).
- [17] Michael Levitt e Cyrus Chothia. "Structural patterns in globular proteins". en. Em: *Nature* 261.5561 (jun. de 1976). Number: 5561 Publisher: Nature Publishing Group, pp. 552–558. ISSN: 1476-4687. DOI: 10.1038/261552a0. URL: <https://www.nature.com/articles/261552a0> (acesso em 29/02/2024).
- [18] Cyrus Chothia et al. "Conformations of immunoglobulin hypervariable regions". en. Em: *Nature* 342.6252 (dez. de 1989). Number: 6252 Publisher: Nature Publishing Group, pp. 877–883.

- ISSN: 1476-4687. DOI: 10.1038/342877a0. URL: <https://www.nature.com/articles/342877a0> (acesso em 29/02/2024).
- [19] Rebecca F. Alford et al. "The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design". Em: *Journal of Chemical Theory and Computation* 13.6 (2017). PMID: 28430426, pp. 3031–3048. DOI: 10.1021/acs.jctc.7b00125.
- [20] Guilherme F Bottino et al. "Structural Discrimination Analysis for Constraint Selection in Protein Modeling". Em: *Bioinformatics (Oxford, England)* (jun. de 2021). ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btab425.
- [21] Faruck Morcos et al. "Direct-coupling analysis of residue coevolution captures native contacts across many protein families". eng. Em: *Proceedings of the National Academy of Sciences of the United States of America* 108.49 (dez. de 2011), E1293–1301. ISSN: 1091-6490. DOI: 10.1073/pnas.1111471108.
- [22] Dong Xu e Yang Zhang. "Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field". eng. Em: *Proteins* 80.7 (jul. de 2012), pp. 1715–1735. ISSN: 1097-0134. DOI: 10.1002/prot.24065.
- [23] Yang Zhang, Adrian K. Arakaki e Jeffrey Skolnick. "TASSER: an automated method for the prediction of protein tertiary structures in CASP6". eng. Em: *Proteins* 61 Suppl 7 (2005), pp. 91–98. ISSN: 1097-0134. DOI: 10.1002/prot.20724.
- [24] K. T. Simons et al. "Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions". eng. Em: *Journal of Molecular Biology* 268.1 (abr. de 1997), pp. 209–225. ISSN: 0022-2836. DOI: 10.1006/jmbi.1997.0959.
- [25] Sergey Ovchinnikov [ @sokrypton ]. My slides in Google Slides format: <https://t.co/iNS9xHuEP1>. en. Tweet. Jun. de 2022. URL: <https://twitter.com/sokrypton/status/1536769463604977666> (acesso em 29/02/2024).
- [26] Thomas A Hopf et al. "The EVcouplings Python framework for coevolutionary sequence analysis". Em: *Bioinformatics* 35.9 (mai. de 2019), pp. 1582–1584. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty862. URL: <https://doi.org/10.1093/bioinformatics/bty862> (acesso em 29/02/2024).

- [27] Jianzhu Ma et al. "Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning". eng. Em: *Bioinformatics (Oxford, England)* 31.21 (nov. de 2015), pp. 3506–3513. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btv472.
- [28] David T. Jones. "Protein secondary structure prediction based on position-specific scoring matrices1Edited by G. Von Heijne". Em: *Journal of Molecular Biology* 292.2 (1999), pp. 195–202. ISSN: 0022-2836. DOI: 10.1006/jmbi.1999.3091.
- [29] J A Cuff et al. "JPred: a consensus secondary structure prediction server." Em: *Bioinformatics* 14.10 (jan. de 1998), pp. 892–893. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.10.892. URL: <https://doi.org/10.1093/bioinformatics/14.10.892> (acesso em 29/02/2024).
- [30] Andrew W. Senior et al. "Improved protein structure prediction using potentials from deep learning". eng. Em: *Nature* 577.7792 (jan. de 2020), pp. 706–710. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1923-7.
- [31] Ashish Vaswani et al. *Attention Is All You Need*. arXiv:1706.03762 [cs]. Ago. de 2023. DOI: 10.48550/arXiv.1706.03762. URL: <http://arxiv.org/abs/1706.03762> (acesso em 28/02/2024).
- [32] Wayne Xin Zhao et al. *A Survey of Large Language Models*. arXiv:2303.18223 [cs]. Nov. de 2023. DOI: 10.48550/arXiv.2303.18223. URL: <http://arxiv.org/abs/2303.18223> (acesso em 29/02/2024).
- [33] Ali Madani et al. "Large language models generate functional protein sequences across diverse families". en. Em: *Nature Biotechnology* 41.8 (ago. de 2023). Number: 8 Publisher: Nature Publishing Group, pp. 1099–1106. ISSN: 1546-1696. DOI: 10.1038/s41587-022-01618-2. URL: <https://www.nature.com/articles/s41587-022-01618-2> (acesso em 29/02/2024).
- [34] Mai Ha Vu et al. "Linguistically inspired roadmap for building biologically reliable protein language models". en. Em: *Nature Machine Intelligence* 5.5 (mai. de 2023). Number: 5 Publisher: Nature Publishing Group, pp. 485–496. ISSN: 2522-5839. DOI: 10.1038/s42256-023-00637-1. URL: <https://www.nature.com/articles/s42256-023-00637-1> (acesso em 29/02/2024).
- [35] Noam Chomsky. *On Nature and Language*. Ed. por Adriana Belletti e Luigi Rizzi. Cambridge: Cambridge University Press, 2002. ISBN: 978-0-521-81548-2. DOI:

- 10 . 1017/CB09780511613876. URL: <https://www.cambridge.org/core/books/on-nature-and-language/210D36D24C9F0397E7599D6AB5F2ACB1> (acesso em 29/02/2024).
- [36] N. Chomsky. "Three models for the description of language". Em: *IRE Transactions on Information Theory* 2.3 (set. de 1956). Conference Name: IRE Transactions on Information Theory, pp. 113–124. ISSN: 2168-2712. DOI: 10 . 1109 / TIT . 1956 . 1056813. URL: <https://ieeexplore.ieee.org/document/1056813> (acesso em 29/02/2024).
- [37] Ethan A. Chi, John Hewitt e Christopher D. Manning. *Finding Universal Grammatical Relations in Multilingual BERT*. en. Mai. de 2020. URL: <https://arxiv.org/abs/2005.04511v2> (acesso em 29/02/2024).
- [38] Mohammed AlQuraishi. "Protein-structure prediction revolutionized". eng. Em: *Nature* 596.7873 (ago. de 2021), pp. 487–488. ISSN: 1476-4687. DOI: 10.1038/d41586-021-02265-4.
- [39] Ratul Chowdhury et al. *Single-sequence protein structure prediction using language models from deep learning*. en. Pages: 2021.08.02.454840 Section: New Results. Ago. de 2021. DOI: 10.1101/2021.08.02.454840. URL: <https://www.biorxiv.org/content/10.1101/2021.08.02.454840v1> (acesso em 28/02/2024).
- [40] Minkyung Baek et al. "Accurate prediction of protein structures and interactions using a three-track neural network". eng. Em: *Science (New York, N.Y.)* 373.6557 (ago. de 2021), pp. 871–876. ISSN: 1095-9203. DOI: 10.1126/science.abj8754.
- [41] *The computing power needed to train AI is now rising seven times faster than ever before*. en. URL: <https://www.technologyreview.com/2019/11/11/132004/the-computing-power-needed-to-train-ai-is-now-rising-seven-times-faster-than-ever-before/> (acesso em 28/02/2024).
- [42] A. G. Murzin et al. "SCOP: a structural classification of proteins database for the investigation of sequences and structures". eng. Em: *Journal of Molecular Biology* 247.4 (abr. de 1995), pp. 536–540. ISSN: 0022-2836. DOI: 10.1006/jmbi.1995.0159.
- [43] Jianlin Cheng et al. "Estimation of model accuracy in CASP13". en. Em: *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019). \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25767>, pp. 1361–1377. ISSN: 1097-0134. DOI: 10 . 1002 . 1002 / prot . 25767. URL:

- <https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25767> (acesso em 28/02/2024).
- [44] Ricardo N. Dos Santos et al. "Enhancing protein fold determination by exploring the complementary information of chemical cross-linking and coevolutionary signals". eng. Em: *Bioinformatics (Oxford, England)* 34.13 (jul. de 2018), pp. 2201–2208. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bty074.
- [45] Ricardo Nascimento Dos Santos et al. "Coevolutionary Signals and Structure-Based Models for the Prediction of Protein Native Conformations". eng. Em: *Methods in Molecular Biology (Clifton, N.J.)* 1851 (2019), pp. 83–103. ISSN: 1940-6029. DOI: 10.1007/978-1-4939-8736-8\_5.
- [46] Tuo Zhang et al. "Intrinsically semi-disordered state and its role in induced folding and protein aggregation". eng. Em: *Cell Biochemistry and Biophysics* 67.3 (2013), pp. 1193–1205. ISSN: 1559-0283. DOI: 10.1007/s12013-013-9638-0.
- [47] Soumya Mishra, Loren L. Looger e Lauren L. Porter. "Inaccurate secondary structure predictions often indicate protein fold switching". Em: *Protein Science : A Publication of the Protein Society* 28.8 (ago. de 2019), pp. 1487–1493. ISSN: 0961-8368. DOI: 10.1002/pro.3664. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6635839/> (acesso em 28/02/2024).
- [48] Tomasz Smolarczyk, Irena Roterman-Konieczna e Katarzyna Stapor. "Protein Secondary Structure Prediction: A Review of Progress and Directions". en. Em: *Current Bioinformatics* 15.2 (), pp. 90–107. URL: <https://www.eurekaselect.com/article/101564> (acesso em 29/02/2024).
- [49] Peter Y. Chou e Gerald D. Fasman. "Prediction of protein conformation". Em: *Biochemistry* 13.2 (jan. de 1974). Publisher: American Chemical Society, pp. 222–245. ISSN: 0006-2960. DOI: 10.1021/bi00699a002. URL: <https://doi.org/10.1021/bi00699a002> (acesso em 26/07/2024).
- [50] P. Y. Chou e G. D. Fasman. "Empirical Predictions of Protein Conformation". en. Em: *Annual Review of Biochemistry* 47. Volume 47, 1978 (jul. de 1978). Publisher: Annual Reviews, pp. 251–276. ISSN: 0066-4154, 1545-4509. DOI:

- 10.1146/annurev.bi.47.070178.001343. URL: <https://www.annualreviews.org/content/journals/10.1146/annurev.bi.47.070178.001343> (acesso em 26/07/2024).
- [51] J. Garnier, D. J. Osguthorpe e B. Robson. "Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins". Em: *Journal of Molecular Biology* 120.1 (mar. de 1978), pp. 97–120. ISSN: 0022-2836. DOI: 10.1016/0022-2836(78)90297-8. URL: <https://www.sciencedirect.com/science/article/pii/0022283678902978> (acesso em 12/03/2024).
- [52] J. Garnier, D. J. Osguthorpe e B. Robson. "Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins". eng. Em: *Journal of Molecular Biology* 120.1 (mar. de 1978), pp. 97–120. ISSN: 0022-2836. DOI: 10.1016/0022-2836(78)90297-8.
- [53] Sergey Ovchinnikov et al. "Protein structure determination using metagenome sequence data". eng. Em: *Science (New York, N.Y.)* 355.6322 (jan. de 2017), pp. 294–298. ISSN: 1095-9203. DOI: 10.1126/science.aah4043.
- [54] Brian Kuhlman e Philip Bradley. "Advances in protein structure prediction and design". eng. Em: *Nature Reviews. Molecular Cell Biology* 20.11 (nov. de 2019), pp. 681–697. ISSN: 1471-0080. DOI: 10.1038/s41580-019-0163-x.
- [55] Rhys Heffernan et al. "Single-sequence-based prediction of protein secondary structures and solvent accessibility by deep whole-sequence learning". eng. Em: *Journal of Computational Chemistry* 39.26 (out. de 2018), pp. 2210–2216. ISSN: 1096-987X. DOI: 10.1002/jcc.25534.
- [56] Krzysztof Kotowski et al. "ProteinUnet-An efficient alternative to SPIDER3-single for sequence-based prediction of protein secondary structures". eng. Em: *Journal of Computational Chemistry* 42.1 (jan. de 2021), pp. 50–59. ISSN: 1096-987X. DOI: 10.1002/jcc.26432.
- [57] Jaspreet Singh et al. "SPOT-1D-Single: improving the single-sequence-based prediction of protein secondary structure, backbone angles, solvent accessibility and half-sphere exposures using a large training set and ensembled deep learning". eng. Em: *Bioinformatics (Oxford, England)* 37.20 (out. de 2021), pp. 3464–3472. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btab316.
- [58] Seth Cooper et al. "Predicting protein structures with a multiplayer online game". en. Em: *Nature* 466.7307 (ago. de 2010). Number: 7307 Publisher: Nature Publishing Group,

- pp. 756–760. ISSN: 1476-4687. DOI: 10 . 1038 / nature09304. URL: <https://www.nature.com/articles/nature09304> (acesso em 29/02/2024).
- [59] M. Shirts e V. S. Pande. “COMPUTING: Screen Savers of the World Unite!” eng. Em: *Science (New York, N.Y.)* 290.5498 (dez. de 2000), pp. 1903–1904. ISSN: 0036-8075. DOI: 10.1126/science.290.5498.1903.
- [60] Sauptik Dhar et al. *On-Device Machine Learning: An Algorithms and Learning Theory Perspective*. arXiv:1911.00623 [cs, stat]. Jul. de 2020. DOI: 10.48550/arXiv.1911.00623. URL: <http://arxiv.org/abs/1911.00623> (acesso em 29/02/2024).
- [61] Davis Blalock et al. *What is the State of Neural Network Pruning?* arXiv:2003.03033 [cs, stat]. Mar. de 2020. DOI: 10.48550/arXiv.2003.03033. URL: <http://arxiv.org/abs/2003.03033> (acesso em 29/02/2024).
- [62] Sepp Hochreiter e Jürgen Schmidhuber. “Long short-term memory”. Em: *Neural Computation* 9.8 (1997). Place: US Publisher: MIT Press, pp. 1735–1780. ISSN: 1530-888X. DOI: 10.1162/neco.1997.9.8.1735.
- [63] Rhys Heffernan et al. “Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility”. eng. Em: *Bioinformatics (Oxford, England)* 33.18 (set. de 2017), pp. 2842–2849. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btx218.
- [64] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. Em: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. por Alessandro Moschitti, Bo Pang e Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, out. de 2014, pp. 1724–1734. DOI: 10 . 3115 / v1 / D14 - 1179. URL: <https://aclanthology.org/D14-1179> (acesso em 28/02/2024).
- [65] Ilya Sutskever, Oriol Vinyals e Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. arXiv:1409.3215 [cs]. Dez. de 2014. DOI: 10.48550/arXiv.1409.3215. URL: <http://arxiv.org/abs/1409.3215> (acesso em 28/02/2024).
- [66] Zhixing Tan et al. *Neural Machine Translation: A Review of Methods, Resources, and Tools*. arXiv:2012.15515 [cs]. Dez. de 2020. DOI: 10.48550/arXiv.2012.15515. URL: <http://arxiv.org/abs/2012.15515> (acesso em 28/02/2024).

- [67] Olaf Ronneberger, Philipp Fischer e Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv:1505.04597 [cs]. Mai. de 2015. DOI: 10.48550/arXiv.1505.04597. URL: <http://arxiv.org/abs/1505.04597> (acesso em 29/02/2024).
- [68] Ye Zhang e Byron Wallace. *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*. arXiv:1510.03820 [cs]. Abr. de 2016. DOI: 10.48550/arXiv.1510.03820. URL: <http://arxiv.org/abs/1510.03820> (acesso em 28/02/2024).
- [69] Wenjie Luo et al. "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". Em: *Advances in Neural Information Processing Systems*. Ed. por D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2016/file/c8067ad1937f728f51288b3eb986afaa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2016/file/c8067ad1937f728f51288b3eb986afaa-Paper.pdf).
- [70] Jack Hanson et al. "Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks". eng. Em: *Bioinformatics (Oxford, England)* 35.14 (jul. de 2019), pp. 2403–2410. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bty1006.
- [71] Nicolas Carion et al. *End-to-End Object Detection with Transformers*. arXiv:2005.12872 [cs] version: 3. Mai. de 2020. DOI: 10.48550/arXiv.2005.12872. URL: <http://arxiv.org/abs/2005.12872> (acesso em 12/03/2024).
- [72] Ian Sillitoe et al. "CATH: increased structural coverage of functional space". Em: *Nucleic Acids Research* 49.D1 (nov. de 2020), pp. D266–D273. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa1079. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7778904/> (acesso em 28/02/2024).
- [73] Helen M. Berman et al. "The Protein Data Bank". Em: *Nucleic Acids Research* 28.1 (2000), pp. 235–242. DOI: 10.1093/nar/28.1.235.
- [74] Robbie P. Joosten et al. "A series of PDB related databases for everyday needs". Em: *Nucleic Acids Research* 39.Database issue (jan. de 2011), pp. D411–D419. ISSN: 0305-1048. DOI: 10.1093/nar/gkq1105. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013697/> (acesso em 28/02/2024).

- [75] W. Kabsch e C. Sander. "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features". eng. Em: *Biopolymers* 22.12 (dez. de 1983), pp. 2577–2637. ISSN: 0006-3525. DOI: 10.1002/bip.360221211.
- [76] D. Frishman e P. Argos. "Knowledge-based protein secondary structure assignment". eng. Em: *Proteins* 23.4 (dez. de 1995), pp. 566–579. ISSN: 0887-3585. DOI: 10.1002/prot.340230412.
- [77] Thomas A. Christensen II et al. *BioJulia/BioAlignments.jl: v3.1.0*. Jan. de 2023. DOI: 10.5281/zenodo.7502559. URL: <https://doi.org/10.5281/zenodo.7502559>.
- [78] Christian Szegedy et al. "Going deeper with convolutions". Em: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.
- [79] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". Em: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.
- [80] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Em: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. por Jill Burstein, Christy Doran e Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, jun. de 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://aclanthology.org/N19-1423> (acesso em 28/02/2024).
- [81] Jeff Bezanson et al. "Julia: A Fresh Approach to Numerical Computing". Em: *Siam Review* 59.1 (jan. de 2017). Publisher: Society for Industrial and Applied Mathematics, pp. 65–98. ISSN: 0036-1445. DOI: 10.1137/141000671.
- [82] Mike Innes. "Flux: Elegant machine learning with Julia". en. Em: *Journal of Open Source Software* 3.25 (mai. de 2018), p. 602. ISSN: 2475-9066. DOI: 10.21105/joss.00602. URL: <https://joss.theoj.org/papers/10.21105/joss.00602> (acesso em 28/02/2024).
- [83] Jonathan Laurent. *jonathan-laurent/AlphaZero.jl*. original-date: 2019-09-04T02:00:41Z. Fev. de 2024. URL: <https://github.com/jonathan-laurent/AlphaZero.jl> (acesso em 28/02/2024).

- [84] Michael Remmert et al. "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment". eng. Em: *Nature Methods* 9.2 (dez. de 2011), pp. 173–175. ISSN: 1548-7105. DOI: 10.1038/nmeth.1818.
- [85] Baris E. Suzek et al. "UniRef: comprehensive and non-redundant UniProt reference clusters". eng. Em: *Bioinformatics (Oxford, England)* 23.10 (mai. de 2007), pp. 1282–1288. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btm098.
- [86] B. Rost, C. Sander e R. Schneider. "Redefining the goals of protein secondary structure prediction". eng. Em: *Journal of Molecular Biology* 235.1 (jan. de 1994), pp. 13–26. ISSN: 0022-2836. DOI: 10.1016/s0022-2836(05)80007-5.
- [87] A. Zemla et al. "A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment". eng. Em: *Proteins* 34.2 (fev. de 1999), pp. 220–223. ISSN: 0887-3585. DOI: 10.1002/(sici)1097-0134(19990201)34:2<220::aid-prot7>3.0.co;2-k.
- [88] Nicholus Bhattacharjee e Parbati Biswas. "Position-specific propensities of amino acids in the β-strand". eng. Em: *BMC structural biology* 10 (set. de 2010), p. 29. ISSN: 1472-6807. DOI: 10.1186/1472-6807-10-29.
- [89] Fahim Farzadfar et al. "Beta-sheet capping: signals that initiate and terminate beta-sheet formation". eng. Em: *Journal of Structural Biology* 161.1 (jan. de 2008), pp. 101–110. ISSN: 1047-8477. DOI: 10.1016/j.jsb.2007.09.024.
- [90] David Sehnal et al. "Mol\* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures". Em: *Nucleic Acids Research* 49.W1 (jul. de 2021), W431–W437. ISSN: 0305-1048. DOI: 10.1093/nar/gkab314. URL: <https://doi.org/10.1093/nar/gkab314> (acesso em 22/04/2024).
- [91] Yuedong Yang et al. "Sixty-five years of the long march in protein secondary structure prediction: the final stretch?" eng. Em: *Briefings in Bioinformatics* 19.3 (mai. de 2018), pp. 482–494. ISSN: 1477-4054. DOI: 10.1093/bib/bbw129.
- [92] R. Sennrich, B. Haddow e A. Birch. "Improving Neural Machine Translation Models with Monolingual Data". Em: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. por K. Erk e N. Smith. Berlin, Germany: Association for Computational Linguistics, ago. de 2016, pp. 86–96. DOI: 10.18653/v1/P16-1009. URL: <https://aclanthology.org/P16-1009>.

- [93] Tianhao Yu et al. "Enzyme function prediction using contrastive learning". Em: *Science* 379.6639 (mar. de 2023). Publisher: American Association for the Advancement of Science, pp. 1358–1363. DOI: 10 . 1126 / science . adf2465. URL: <https://www.science.org/doi/10.1126/science.adf2465> (acesso em 22/04/2024).