

**Gerador de proteínas para o tratamento da
Hemofilia tipo B baseado em *Deep
Reinforcement Learning***

Arthur Moriggi Pimenta

DISSERTAÇÃO APRESENTADA AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA UNIVERSIDADE DE SÃO PAULO
PARA OBTENÇÃO DO TÍTULO DE
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Roberto Hirata Junior

São Paulo
Novembro de 2023

**Gerador de proteínas para o tratamento da
Hemofilia tipo B baseado em *Deep
Reinforcement Learning***

Arthur Moriggi Pimenta

Esta é a versão original da dissertação
elaborada pelo candidato Arthur
Moriggi Pimenta, tal como
submetida à Comissão Julgadora.

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0
(Creative Commons Attribution 4.0 International License)*

Lista de abreviaturas

MSA	Alinhamento múltiplo de sequências (<i>Multiple Sequence Alignment</i>)
IS	Idêntidade de Sequência
TM-Score	Escore de Modelagem de Template (<i>Template Modeling Score</i>)
LLM	Modelo Grande de Linguagem (<i>Large Language Model</i>)
PPO	Otimização de Política Proxima (<i>Proximal Policy Optimization</i>)
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo

Lista de figuras

1	Projeto de Sequências	2
2	Projeto de Sequências baseado em busca	3
3	Projeto de Sequências baseado em aprendizado profundo	4
4	Projeto de Sequências proposto	5
2.1	Componentes de uma proteína	10
2.2	Cada aminoácido tem uma abreviação de uma ou três letras - figura 3-2 adaptada B. ALBERTS, 2002.	10
2.3	Exemplo de conformação protéica B. ALBERTS, 2002 - adaptado	11
2.4	Atuação do FIXa na ativação do FX	11
2.5	Interação entre FIXa e FVIIIa, formando o complexo tenase. Adaptação da figura 3 de M. H. B. e. AL, 2024	12
2.6	Estrutura alvo - Domínio protease do Fator IXa	12
2.7	A maior parte dos resíduos do FIXa possuem um baixo CS. Os maiores valores estão concentrados em cerca de 12 resíduos espalhados ao longo da sequência.	16
2.8	Distâncias entre pares de aminoácidos LOPES TIAGO J. S., 2021	17
2.9	Arquitetura de uma rede neural - BISHOP, 2006 adaptado	19
2.10	Neurônio processando sinal FIGUEIREDO, 2018	19
3.1	Obtenção das condições iniciais	23
3.2	Construção e uso do <i>Encoder</i>	25
3.3	Escolha da dimensionalidade dos Embeddings	25
3.4	Exemplo de mutação. O aminoácido A na posição 3 está sendo substituído pelo o aminoácido C	26
3.5	Arquitetura da rede neural do Agente (<i>GenSeq</i>)	27
3.6	Primeiro estágio de treinamento	28
3.7	Comparação do Agente pré treinado com um Agente aleatório	29
3.8	Média móvel das recompensas por episódio - treino	30

3.9	Segundo estágio - Treino	31
3.10	Comparação do Agente treinado com um Agente aleatório	32
3.11	Média móvel das recompensas por episódio - treino	33
3.12	Gerando sequência otimizada	34
4.1	Variação da similaridade entre as sequências geradas e o FIX	35
4.2	Alinhamento entre FIX (em azul) e a proteína ID 34 (em vermelho)	36
4.4	Comparação do IM entre as proteínas geradas e o FIX	37
4.5	IM calculado para cada grupo de alelo	38
4.6	Resultado do <i>Docking</i> entre as proteínas geradas e o fator VII	40

Lista de tabelas

3.1	Hiperparâmetros do PPO	26
3.2	Parâmetros preliminares	29
3.3	Parâmetros preliminares	32
4.1	Generated proteins	36
4.2	Grupos de alelos considerados e a frequência com que ocorrem em seres humanos.	36

Sumário

Introdução	1
0.1 Projeto de sequências	1
0.1.1 Baseado em busca	2
0.1.2 Baseado em aprendizado profundo	3
0.2 Proposta	4
0.3 Objetivos	5
1 Revisão Bibliográfica	7
1.0.1 Predição de estruturas	7
2 Fundamentação teórica	9
2.1 Proteína	9
2.1.1 Estrutura proteica	10
2.1.2 O Fator IX	11
2.1.3 Docking	13
2.1.4 Resposta Imunológica	14
2.1.5 Mutação	14
2.1.6 Similariedade entre estruturas	14
2.1.7 Conservation Score	16
2.1.8 Similariedade entre aminoácidos	16
2.2 Aprendizado por reforço profundo	17
2.2.1 Processo de Decisão de Markov	17
2.2.2 Redes Neurais	18
2.2.3 Método Actor-Critic	20
2.2.4 PPO	20
2.2.5 Escalonamento Multidimensional (MDS)	21
3 Metodologia	23
3.1 Condições iniciais	23

3.2	Treinamento	24
3.2.1	Ambiente de aprendizado	24
3.2.2	Agente	26
3.2.3	Treinamento - Estágio I	27
3.2.4	Treinamento - Estágio II	30
3.3	Geração de sequências	33
4	Resultados	35
4.1	Resposta imunológica	36
4.2	Docking	39
5	Conclusão	41
 Apêndices		
 Anexos		
 Referências		
		43

Introdução

Em andamento

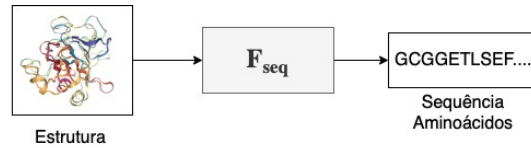
A hemofilia tipo B é uma doença hereditária rara, caracterizada por uma deficiência no fator IX de coagulação sanguínea. Os pacientes com a doença enfrentam um risco maior de sofrer com hemorragias graves, tanto internas quanto externas, podendo levar a complicações debilitantes e até mesmo à morte [PM, 2020](#).

Imagem do fator IX completo

Embora tenham sido obtidos avanços notáveis no tratamento da doença nas últimas décadas, muitos desafios ainda persistem. O tratamento tradicional da hemofilia envolve a infusão de fatores de coagulação recombinantes ou derivados do plasma sanguíneo [GOUW, 2013](#). Apesar de sua eficácia na prevenção de hemorragias, esse tratamento possui limitações, tais como a necessidade de infusões frequentes, a possibilidade de desenvolvimento de inibidores anticoagulantes e, principalmente, os altos custos envolvidos [MANCUSO, 2005](#). **Pegar algum artigo que fale sobre a instabilidade do fator IX, que faz com que o tratamento seja caro e sobre a resposta imune.** O projeto de sequências proteicas surge como uma abordagem que tem potencial de reduzir substancialmente os custos associados ao tratamento da doença, ao criar proteínas sob medida com eficácia comparável.

0.1 Projeto de sequências

O projeto de sequências se refere ao processo de criar ou otimizar uma sequência de aminoácidos de modo a produzir uma proteína cuja estrutura tridimensional possua características desejadas. Esta é uma tarefa desafiadora que, a depender do tamanho da sequência, é inviável de se resolver através de uma busca por força bruta, devido às inúmeras permutações possíveis que se pode obter com os 20 aminoácidos conhecidos na natureza [WENZE DING e GONG, 2022](#).



O projeto de sequências se resume a desenvolver uma função F_{seq} que mapeia uma estrutura protéica tridimensional em uma sequência de aminoácidos.

Figura 1: *Projeto de Sequências*

0.1.1 Baseado em busca

O projeto de sequências baseado em busca, em geral, define a função F_{seq} através do seguinte pipeline:

1. Dada uma estrutura proteica, é definida uma sequência de aminoácidos inicial, baseada em uma heurística H .
2. A sequência serve de entrada para a função objetivo F_{obj} que calcula a métrica a ser otimizada pelo pipeline.
3. Se a sequência atual é melhor do que a sequência final, em termos da métrica calculada pela F_{obj} , então o Agente A define a sequência atual como a nova sequência final. Além disso, A também atualiza a sequência atual a partir de mutações.
4. O processo se repete a partir do item 2 até que seja atingido um critério de parada. Geralmente é baseado no número de iterações e/ou na métrica objetivo.

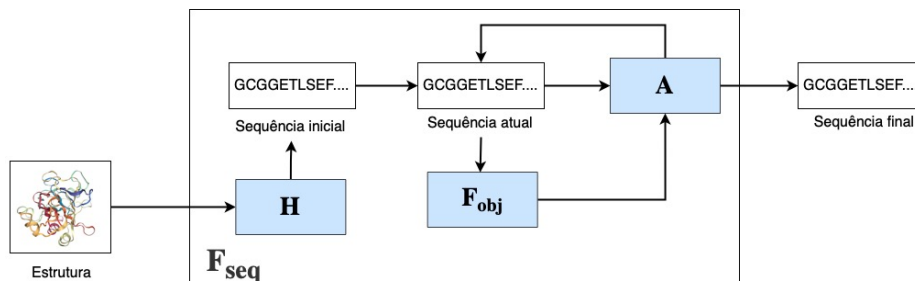


Figura 2: Projeto de Sequências baseado em busca

O *Rosetta* B, 2019 se baseia em um pipeline como este, onde a heurística H define como partida uma sequência de uma proteína homóloga ou estruturalmente semelhante a estrutura alvo. Sua função objetivo F_{obj} mede a energia livre da proteína, considerando diversos fatores como interações de *Van der Waals*, interações eletrostáticas, ligações de hidrogênio, solubilidade, entre outros. O *Rosetta* busca minimizar a energia livre calculada pela F_{obj} . Portanto, seu Agente A decide que a sequência final passa a ser a atual se houver um decréscimo na energia livre. Além disso, atribui modificações na sequência atual de forma aleatória, utilizando o Método de Monte Carlo (MMC).

Existem abordagens cujo o objetivo da busca se concentra em maximizar a similaridade entre as proteínas, como adotado por xx. Com o objetivo de criar novas proteínas nunca antes vistas na natureza, ANISHCHENKO, 2021 parte de uma sequência aleatória de aminoácidos, i.e, não possui uma heurística inicial. A função M , assim como o *Rosetta*, consiste de um MCC. Já a função objetivo F_{obj} calcula a divergência de *Kullback-Leibler* entre ...

0.1.2 Baseado em aprendizado profundo

O projeto de sequências baseado em aprendizado profundo define a função F_{seq} a partir de redes neurais artificiais. O J. D. e. AL, 2022 por exemplo, determina a F_{seq} como uma rede neural profunda, denominada *ProteinMPNN*, que mapeia de forma direta a estrutura alvo à sequência de aminoácidos. A rede é construída através de uma *Message Passing Neural Network* (MPNN) composta por uma arquitetura *encoder-decoder* que se baseia nas características da estrutura - distância e orientação dos átomos no espaço - para fazer previsões J. D. e. AL, 2022.

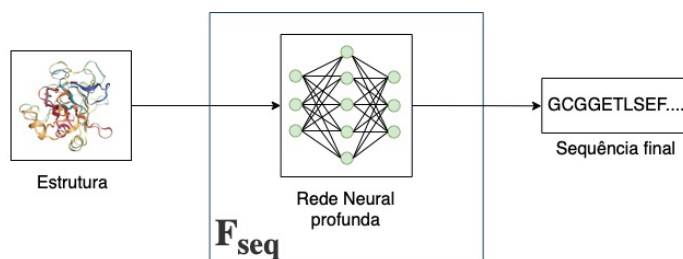


Figura 3: Projeto de Sequências baseado em aprendizado profundo

Falar sobre o RL in latent space e mencionar alg. genetico

0.2 Proposta

Além de possivelmente provocar uma resposta imune no paciente, o tratamento da hemofilia tipo B, baseado na infusão do Fator IX, é custoso devido ao pouco tempo de vida do Fator IX na corrente sanguínea, demandando infusões frequentes. Neste sentido, o presente trabalho se propõe a responder a seguinte pergunta científica: É possível, através de projeto de sequências, projetar uma proteína que desempenhe a mesma função do Fator IX mas com um perfil imunológico melhor e que demande menos infusões no tratamento da hemofilia tipo B?

Para responder a esta questão, formulamos a hipótese de que, se existir uma proteína alternativa superior ao Fator IX, ela deverá ser estruturalmente similar. Com base nisso, pretendemos selecionar, dentre um conjunto de proteínas estruturalmente semelhantes ao Fator IX, aquelas que satisfaçam os critérios estabelecidos, caso existam.

Explicar que, para simplificar o problema, vamos projetar apenas um dominio, que é o mais relevante Imagem do FIX destacando o dominio que vamos projetar

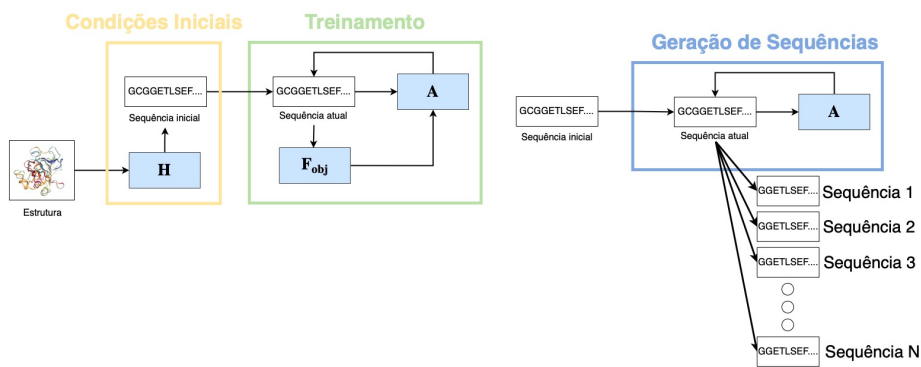
De modo a obter esse conjunto de proteínas, será desenvolvido um pipeline que combina a arquitetura do projeto de sequências baseado em busca e técnicas de aprendizado profundo. O processo será dividido em três módulos: Condições Iniciais, Treinamento e Geração de Sequências.

O módulo de Condições iniciais consiste na obtenção da sequência inicial de aminoácidos através da heurística H , bem como o cálculo do erro inicial associado a esta sequência,

usando a função objetivo F_{obj} . Vamos utilizar o *ProteinMPNN* como heurística H e a métrica de similaridade estrutural *Template Matching Score* (*TMScore*) como função objetivo.

Diferente do uso de agentes que realizam mutações aleatórias, como no *Rosetta*, no módulo de Treinamento, treinaremos uma rede neural profunda para atuar como o Agente A , capaz de realizar mutações que otimizem a similaridade com a estrutura alvo, isto é, o Fator IX.

Após o treinamento, no módulo de Geração de Sequências, espera-se que o Agente seja capaz de gerar várias proteínas estruturalmente semelhantes à estrutura alvo por meio de mutações direcionadas. A partir das sequências geradas, avaliaremos a existência de candidatas viáveis para substituir o Fator IX no tratamento da hemofilia tipo B.



- o *ProteinMPNN* será utilizado como heurística H .
- Na função objetivo F_{obj} será calculada uma métrica de similaridade entre a estrutura da proteína atual e a estrutura alvo que pretendemos maximizar: *Template Matching Score* - *TMScore*.
- O Agente A será uma rede neural profunda, nomeada de *GenSeq*, treinada para receber uma sequência de aminoácidos e prever a mutação que maximiza a similaridade com a estrutura alvo, i.e, otimiza F_{obj} . O treinamento do A será feito com aprendizado por reforço profundo, utilizando a técnica *Proximal Policy Optimization* - PPO
- Depois de treinado, A será utilizado como gerador de proteínas estruturalmente semelhantes ao Fator IX.

Figura 4: Projeto de Sequências proposto

0.3 Objetivos

Definimos os seguintes objetivos:

1. Obter um conjunto de sequências de aminoácidos que mimetizem a estrutura do fator de coagulação IX baseado na métrica TM-Score.
2. Avaliar, dentro do conjunto de proteínas obtidas em 1, quais possuem potencial de substituir o Fator IX de modo tornar o tratamento de Hemofilia tipo B mais acessível.
3. Desenvolver um *pipeline* genérico que produza sequências que mimetizem uma estrutura qualquer.

Capítulo 1

Revisão Bibliográfica

TO DO LOPES TIAGO J. S., 2021

1.0.1 Predição de estruturas

TO DO

Capítulo 2

Fundamentação teórica

Em andamento

Neste capítulo vamos apresentar os fundamentos teóricos utilizados neste trabalho acerca de proteínas e aprendizado por reforço profundo.

2.1 Proteína

Uma proteína é uma macromolécula biológica composta por cadeias de aminoácidos. Constituem a maior parte da massa seca de uma célula, podendo desempenhar funções enzimáticas, estruturais, imunológicas, de transporte, entre outras [B. ALBERTS, 2002](#).

Existem milhares de proteínas diferentes em uma célula, onde cada uma é composta por uma sequência distinta dos 20 tipos de aminoácidos encontrados na natureza, unidos através de ligações peptídicas. Por conta disto, as proteínas são também conhecidas como polipeptídeos [B. ALBERTS, 2002](#).

A cadeia polipeptídica é composta por três componentes: a cadeia principal, as cadeias laterais e as ligações peptídicas. A cadeia principal é também referida como o esqueleto da proteína e é definida como uma série repetitiva e encadeada de átomos de carbono, nitrogénio e oxigénio. Anexadas a cadeia principal, encontram-se as cadeias laterais. Estas não estão envolvidas nas ligações peptídicas e são responsáveis por caracterizar as principais propriedades da proteína [B. ALBERTS, 2002](#).

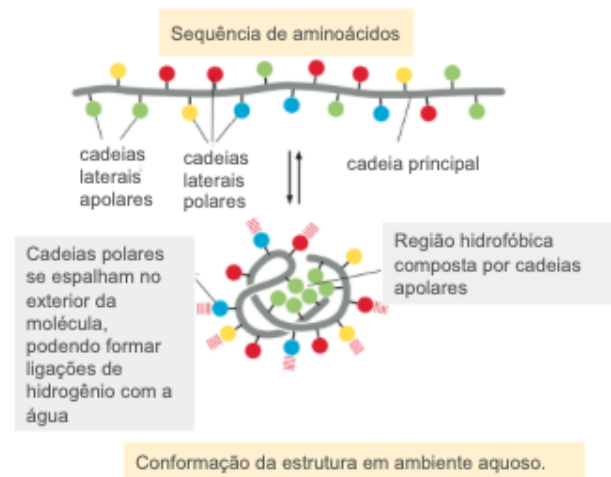


Figura 2.3: Exemplo de conformação protéica *B. ALBERTS, 2002 - adaptado*

2.1.2 O Fator IX

O Fator IX (FIX) é uma proteína essencial para o processo de coagulação. Quando ativado (FIXa), atua em conjunto com o Fator VIIIa (FVIIIa) para formar um complexo conhecido como tenase intrínseco, que é responsável por ativar o Fator X. Este processo é fundamental para a geração de trombina e formação de coágulos *M. H. B. e. AL, 2024*. A ausência ou disfunção do FIX compromete o processo de ativação do FX, impossibilitando uma coagulação saudável, o que configura a hemofilia tipo B.

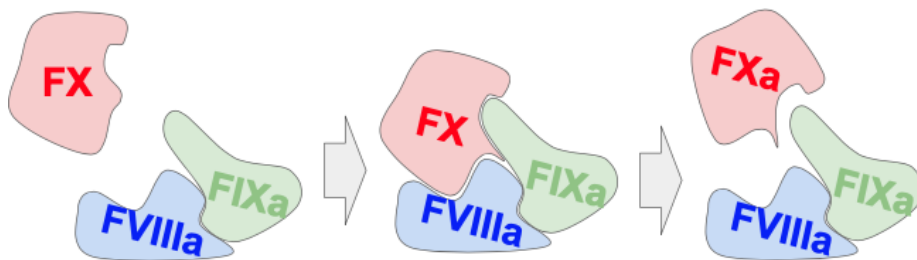


Figura 2.4: Atuação do FIXa na ativação do FX

O FIXa é composta por diferentes domínios, cada um desempenhando um papel essencial nas interações moleculares que permitem a formação do complexo tenase intrínseco. Na região N-terminal, o domínio Gla (gamma-carboxiglutâmico) é responsável por ligar o FIXa a superfícies de membranas fosfolipídicas, uma interação dependente de íons de cálcio, fundamental para posicionar o complexo de coagulação no local da lesão vascular *M. H. B. e. AL, 2024*. Os domínios de fator de crescimento epidérmico (EGF1 e EGF2) desempenham funções complementares: o EGF1 facilita o reconhecimento de cofatores e substratos, enquanto o EGF2 interage com o domínio A3 do Fator VIIIa, contribuindo para a montagem do complexo tenase *M. H. B. e. AL, 2024*. O papel mais crítico para a estabilidade e a funcionalidade do complexo tenase reside no domínio protease, cuja interação com

o domínio A2 do FVIIIa permite a ativação conformacional necessária para uma catálise eficiente do FX. M. H. B. e. AL, 2024 explica que variantes do FIXa, como o FIX-Padua, que modificam o domínio protease, apresentam maior eficiência catalítica, destacando seu papel central no projeto de proteínas otimizadas para terapia de hemofilia.

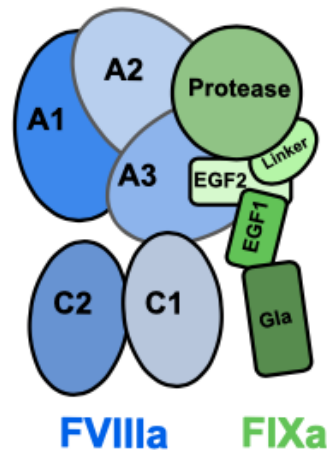


Figura 2.5: Interação entre FIXa e FVIIIa, formando o complexo tenase. Adaptação da figura 3 de M. H. B. e. AL, 2024

A estrutura do domínio protease é a estrutura alvo do projeto de sequências proposto por este trabalho. É composta por 235 resíduos. Logo, o espaço de possíveis novas sequências com este tamanho, considerando 20 diferentes aminoácidos, é de 20^{235} .

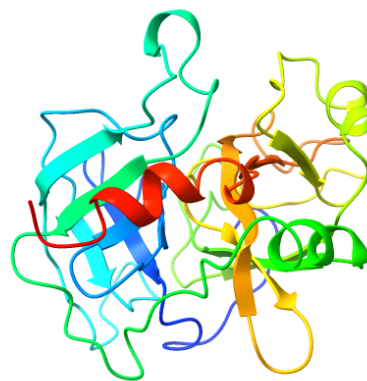


Figura 2.6: Estrutura alvo - Domínio protease do Fator IXa

As proteínas projetadas neste estudo serão desenvolvidas para mimetizar a estrutura do domínio protease. O objetivo é identificar, entre as variantes geradas, aquelas capazes de estabelecer interações mais estáveis com o FVIIIa, aprimorando a estabilidade do complexo tenase intrínseco. Adicionalmente, espera-se que as novas estruturas apresentem um perfil

imunogênico reduzido em comparação ao FIXa nativo, contribuindo para maior segurança e eficácia terapêutica.

2.1.3 Docking

A análise de docking é uma técnica computacional amplamente utilizada no estudo de interações biomoleculares, particularmente para prever a orientação e a afinidade entre duas moléculas, como proteínas. Nesse sentido, é essencial para avaliar como uma proteína substituta do FIXa interage com o FVIIIa. O docking molecular simula o reconhecimento entre uma molécula "receptor" e outra "ligante". No caso de interações proteína-proteína, o receptor e o ligante são ambos macromoléculas, cujas superfícies e conformações influenciam a formação de um complexo estável. O objetivo do docking é identificar as poses mais favoráveis termodinamicamente, ou seja, aquelas que minimizam a energia livre do sistema.

Durante o docking, diversas poses do ligante em relação ao receptor são geradas. Ferramentas como RosettaDock ou AutoDock realizam essa busca, explorando translações, rotações e flexibilidade local. Cada pose é avaliada com base em métricas energéticas, como a *Contact Molecular Surface* (CMS), *Interface Buried SASA* (IBSASA), *Delta Gibbs Free Energy of Binding* (DDG) e *Surface Area Potential Score* (SAP Score).

A CMS é uma medida desenvolvida para superar as limitações de métricas tradicionais, como Shape Complementarity (SC) e Delta Solvent Accessible Surface Area (Delta SASA) [COVENTRY, 2021](#). Enquanto SC analisa a complementaridade geométrica entre superfícies moleculares e Delta SASA calcula a área de superfície acessível ao solvente antes e depois da complexação, ambas possuem limitações na identificação de interfaces bem empacotadas. Em contraste, o CMS combina a área de contato com um fator de penalização baseado na distância entre triângulos de superfície, oferecendo uma representação mais precisa da qualidade do empacotamento na interface [COVENTRY, 2021](#).

A IBSASA é frequentemente usada para estimar a extensão de área de superfície enterrada durante a formação de um complexo proteico, correlacionando-se com a estabilidade de ligação. No entanto, sozinha, essa métrica pode ser enganosa em interfaces mal organizadas, onde grandes áreas de contato podem não resultar em interações eficazes [COVENTRY, 2021](#).

Por outro lado, a DDG fornece uma medida direta do quão favorável energeticamente é uma interação, com valores mais negativos indicando maior estabilidade termodinâmica [COVENTRY, 2021](#). A DDG, calculada frequentemente com o software Rosetta, permite a otimização de sequências e conformações para maximizar a energia de ligação.

O SAP Score avalia a hidrofobicidade com base na superfície acessível ao solvente, sendo uma ferramenta crucial para prever a propensão à agregação e a estabilidade de complexos [COVENTRY, 2021](#). Alterações no SAP Score após a ligação, conhecidas como Delta SAP Score, fornecem insights sobre a contribuição hidrofóbica para a força motriz da interação.

2.1.4 Resposta Imunológica

A resposta imune desencadeada por uma proteína terapêutica é um fator crítico que pode influenciar diretamente sua segurança e eficácia. No contexto do projeto de uma proteína substituta do FIXa, a análise dessa resposta imune é especialmente relevante para evitar complicações associadas à formação de inibidores, uma limitação comum no tratamento de hemofilia. Esses inibidores são anticorpos neutralizantes que se ligam à proteína administrada, reduzindo sua eficácia e exigindo estratégias de tratamento alternativas mais caras e menos seguras.

A imunogenicidade está intrinsecamente relacionada à presença de epítomos imunogênicos na sequência da proteína. Esses epítomos são regiões específicas que se ligam a moléculas de MHC classe II em células que apresentam antígenos (APCs), facilitando a ativação de linfócitos T auxiliares [BJOERN PETERS e SETTE, 2020](#). A afinidade de ligação é um fator determinante na indução da resposta imune. Altas afinidades entre peptídeos e MHC-II aumentam a probabilidade de reconhecimento imune e subsequente ativação de células T [BJOERN PETERS e SETTE, 2020](#).

A predição de afinidade de ligação entre epítomos e MHC-II emprega ferramentas computacionais baseadas em aprendizado de máquina, como o NetMHCIIpan, que simulam interações entre peptídeos derivados da proteína e uma variedade de alelos de MHC. A sequência da proteína é fragmentada em peptídeos sobrepostos com comprimentos típicos de 8 a 15 aminoácidos, e a afinidade de cada peptídeo é quantificada por meio do valor de IC_{50} (concentração inibitória de 50%). Peptídeos com valores de IC_{50} baixos ($IC_{50} < 50$ nM) indicam maior afinidade de ligação e, portanto, são mais propensos a serem reconhecidos como epítomos imunogênicos [BJOERN PETERS e SETTE, 2020](#).

Portanto, o objetivo ao projetar a proteína substituta do FIXa é não apenas otimizar a estabilidade de ligação ao FVIIIa, mas também evitar a apresentação de novos epítomos imunogênicos que poderiam levar à formação de inibidores. A redução da imunogenicidade melhora a adesão ao tratamento e reduz complicações, consolidando a importância da integração de predições imunológicas no pipeline de projeto de proteínas terapêuticas.

2.1.5 Mutação

Uma mutação no contexto de proteínas refere-se a uma modificação na sequência de aminoácidos. Por menor que seja esta modificação, a mutação pode acarretar em uma alteração na estrutura tridimensional da proteína, uma vez que a ordem que os aminoácidos estão dispostos na sequência influencia diretamente na conformação da estrutura. Neste trabalho, vamos assumir que uma mutação consiste na substituição de apenas um aminoácido por outro na sequência.

2.1.6 Similaridade entre estruturas

Para projetar proteínas semelhantes ao FIXa, é necessário definir uma métrica de similaridade entre estruturas. Duas métricas amplamente utilizadas são o *Root-Mean-Square Deviation* (RMSD) e o *Template Modeling Score* (TM-score) [ZHANG, 2004](#), cada uma com características próprias que influenciam sua adequação a diferentes cenários de

comparação estrutural.

O **RMSD** é definido como a raiz quadrada da média dos quadrados das distâncias entre pares de átomos correspondentes em duas estruturas alinhadas:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N d_i^2}, \quad (2.1)$$

onde N representa o número de pares de átomos alinhados e d_i é a distância entre os i -ésimos átomos nas estruturas comparadas. Embora seja uma métrica intuitiva, o RMSD é altamente sensível a desalinhamentos locais e não é normalizado pelo comprimento da proteína, o que limita sua eficácia ao comparar proteínas de diferentes tamanhos.

O **TM-score**, por sua vez, mede a similaridade estrutural global, superando limitações do RMSD. ZHANG, 2004 a formulou da seguinte maneira:

$$\text{TM-score} = \frac{1}{L_{\text{ref}}} \sum_{i=1}^{L_{\text{model}}} \frac{1}{1 + \left(\frac{d_i}{d_0(L_{\text{ref}})} \right)^2}, \quad (2.2)$$

onde L_{ref} é o número de resíduos na estrutura de referência, L_{model} é o número de resíduos no modelo comparado, e d_i é a distância entre os átomos C_α alinhados.

ZHANG, 2004 define d_0 em função de L_{ref} da seguinte maneira:

$$d_0(L_{\text{ref}}) = 1.24 \times (L_{\text{ref}} - 15)^{1/3} - 1.8. \quad (2.3)$$

Este parâmetro adapta a escala de distâncias consideradas significativas para o alinhamento estrutural, ajustando-se ao comprimento da proteína de referência, L_{ref} . O termo $(L_{\text{ref}} - 15)^{1/3}$ reflete uma relação empírica entre o tamanho da proteína e as distâncias típicas entre resíduos alinhados, modelando o crescimento da distância permissível conforme a proteína aumenta de tamanho. A constante multiplicativa 1.24 e o deslocamento -1.8 também foram determinados empiricamente para fornecer um valor de d_0 que otimiza a correlação entre o TM-score e a percepção biológica de similaridade estrutural. Essa função adapta o limiar de normalização das distâncias entre pares de resíduos alinhados, garantindo que proteínas de diferentes comprimentos sejam comparadas de forma justa. Proteínas maiores toleram distâncias absolutas maiores entre resíduos alinhados sem que o TM-score seja severamente penalizado ZHANG, 2004.

O TMScore varia entre 0 e 1, onde valores próximos a 1 indicam alta similaridade topológica ZHANG, 2004. Se a distância entre cada par alinhado de átomos C_α for igual a 0, então o TMScore é igual a 1, indicando sobreposição completa entre as macromoléculas. Em contra partida, quanto maiores são as distâncias, menor é o TMScore (mais próximo de 0).

Devido à sua robustez na avaliação da similaridade estrutural global e à sua interpretabilidade, com valores limitados entre 0 e 1, adotaremos o TM-score como a métrica principal

a ser otimizada neste projeto. O cálculo desta medida será feito a partir da plataforma desenvolvida por [ZHANG, 2022](#).

2.1.7 Conservation Score

O *Conservation Score* (CS) é uma métrica que estima a importância ou contribuição individual de cada resíduo da sequência na caracterização da estrutura da proteína. Resíduos cruciais na estrutura da proteína tendem a ser conservados, visto que alterações neles podem impactar significativamente a função proteica. O score é obtido levando-se em conta a frequência com que certas combinações de aminoácidos ocorrem em proteínas homólogas ao longo da evolução das espécies [EDDY, 2004](#). Para este trabalho, utilizamos o CS calculado pelo servidor ConsurfDB [BEN CHORIN, 2020](#), tendo a estrutura do FIXa de coagulação como entrada.

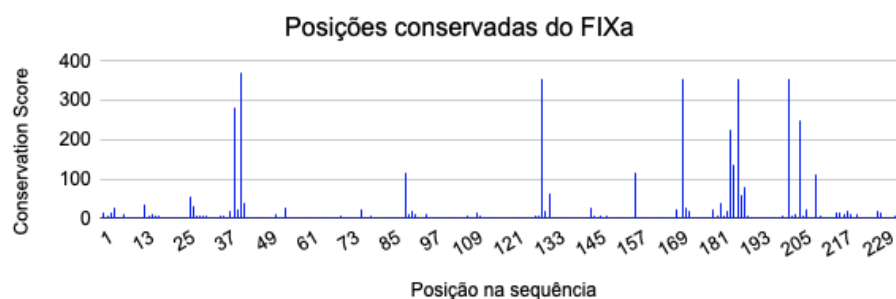


Figura 2.7: A maior parte dos resíduos do FIXa possuem um baixo CS. Os maiores valores estão concentrados em cerca de 12 resíduos espalhados ao longo da sequência.

A identificação de posições conservadas é um aspecto fundamental ao projetar novas proteínas, pois fornece informações essenciais para evitar mutações em regiões críticas, preservando assim a estabilidade estrutural e a função biológica da molécula.

2.1.8 Similaridade entre aminoácidos

Da mesma forma que regiões com altos score de conservação devem ser evitadas, convém evitar mutações que substituem aminoácidos muito diferentes. Nesse sentido, para mensurar o nível de similaridade entre os aminoácidos, utilizamos uma métrica de distância fundamentada no trabalho de [LOPES TIAGO J. S., 2021](#), que propõe uma abordagem vetorial para representar as propriedades de cada aminoácido. Nesse estudo, cada aminoácido foi descrito por um vetor de 544 dimensões, incorporando uma ampla gama de características físico-químicas e estruturais. A construção desses vetores foi realizada com o auxílio do pacote seqinR [CHARIF, 2007](#), uma ferramenta especializada em análise de sequências biológicas. O pacote seqinR utiliza uma base de dados abrangente contendo propriedades como peso molecular, hidrofobicidade, polaridade, acessibilidade à superfície, entre outras, para derivar descritores numéricos que caracterizam as propriedades de cada aminoácido.

A partir dessa representação, [LOPES TIAGO J. S., 2021](#) introduziu uma matriz de distâncias, construída a partir das distâncias euclidianas de cada par de vetores, fornecendo uma

medida quantitativa da dissimilaridade entre aminoácidos.

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0	1.5	1.57	1.51	1.52	1.59	1.51	1.52	1.52	1.46	1.48	1.51	1.49	1.47	1.6	1.38	1.49	1.44	1.55	1.59
C	1.5	0	1.6	1.54	1.48	1.68	1.51	1.46	1.61	1.49	1.48	1.59	1.34	1.5	1.61	1.53	1.54	1.42	1.57	1.56
D	1.57	1.6	0	1.48	1.65	1.61	1.55	1.62	1.59	1.67	1.65	1.51	1.46	1.53	1.63	1.52	1.58	1.6	1.72	1.62
E	1.51	1.54	1.48	0	1.62	1.63	1.53	1.6	1.58	1.63	1.58	1.5	1.51	1.47	1.61	1.56	1.58	1.57	1.7	1.63
F	1.52	1.48	1.65	1.62	0	1.63	1.48	1.39	1.61	1.45	1.48	1.56	1.6	1.54	1.61	1.58	1.52	1.42	1.53	1.49
G	1.59	1.68	1.61	1.63	1.63	0	1.58	1.6	1.57	1.65	1.68	1.53	1.47	1.57	1.59	1.54	1.62	1.57	1.77	1.59
H	1.51	1.51	1.55	1.53	1.48	1.58	0	1.49	1.51	1.52	1.46	1.48	1.44	1.46	1.52	1.51	1.54	1.47	1.62	1.5
I	1.52	1.46	1.62	1.6	1.39	1.6	1.49	0	1.58	1.42	1.42	1.5	1.58	1.51	1.57	1.57	1.53	1.14	1.55	1.45
K	1.52	1.61	1.59	1.58	1.61	1.57	1.51	1.58	0	1.59	1.61	1.48	1.54	1.45	1.49	1.44	1.49	1.55	1.65	1.56
L	1.46	1.49	1.67	1.63	1.45	1.65	1.52	1.42	1.59	0	1.5	1.58	1.64	1.56	1.62	1.47	1.52	1.43	1.53	1.54
M	1.48	1.48	1.65	1.58	1.48	1.68	1.46	1.42	1.61	1.5	0	1.62	1.55	1.52	1.65	1.6	1.6	1.48	1.57	1.57
N	1.51	1.59	1.51	1.5	1.56	1.53	1.48	1.5	1.48	1.58	1.62	0	1.4	1.45	1.52	1.35	1.49	1.56	1.62	1.54
P	1.49	1.34	1.46	1.51	1.6	1.47	1.44	1.58	1.54	1.64	1.55	1.4	0	1.45	1.57	1.38	1.46	1.55	1.65	1.58
Q	1.47	1.5	1.53	1.47	1.54	1.57	1.46	1.51	1.45	1.56	1.52	1.45	1.45	0	1.52	1.43	1.44	1.49	1.59	1.53
R	1.6	1.61	1.63	1.61	1.61	1.59	1.52	1.57	1.49	1.62	1.65	1.52	1.57	1.52	0	1.53	1.56	1.51	1.73	1.55
S	1.38	1.53	1.52	1.56	1.58	1.54	1.51	1.57	1.44	1.47	1.6	1.35	1.38	1.43	1.53	0	1.21	1.49	1.55	1.5
T	1.49	1.54	1.58	1.58	1.52	1.62	1.54	1.53	1.49	1.52	1.6	1.49	1.46	1.44	1.56	1.21	0	1.43	1.53	1.56
V	1.44	1.42	1.6	1.57	1.42	1.57	1.47	1.14	1.55	1.43	1.48	1.56	1.55	1.49	1.51	1.49	1.43	0	1.59	1.4
W	1.55	1.57	1.72	1.7	1.53	1.77	1.62	1.55	1.65	1.53	1.57	1.62	1.65	1.59	1.73	1.55	1.53	1.59	0	1.59
Y	1.59	1.56	1.62	1.63	1.49	1.59	1.5	1.45	1.56	1.54	1.57	1.54	1.58	1.53	1.55	1.5	1.56	1.4	1.59	0

Figura 2.8: Distâncias entre pares de aminoácidos LOPES TIAGO J. S., 2021

2.2 Aprendizado por reforço profundo

Nesta seção vamos introduzir os conceitos envolvidos no aprendizado por reforço profundo, que formam a base teórica da técnica que iremos utilizar: Proximal Policy Optimization (PPO). Para tal, será fornecido um contexto sobre Processo de Decisão de Markov (MDP), Redes Neurais (NN) e o método Ator-Crítico (AC).

2.2.1 Processo de Decisão de Markov

O Processo de Decisão de Markov (MDP) é utilizado para modelar processos de forma probabilística. Ele é chamado de markoviano pois a distribuição de probabilidade de um estado depende apenas do estado anterior e da ação selecionada. Uma ação dá origem a um novo estado ao passo que promove alterações no estado atual. Um Agente seleciona cada ação com base em uma política (π) que mapeia estados a ações. Em um MDP, o processo evolui em etapas, ou *steps*, à medida que o agente toma ações. A sequência de estados, desde o inicial até o final, é chamada de episódio. A qualidade de cada ação é avaliada com base no quão benéfico é o estado alcançado. De acordo com PELLEGRINI, 2007, o MDP pode ser definido como uma tupla $\langle S, A, T, R \rangle$ onde:

- S é o conjunto de possíveis estados;
- A é o conjunto de ações que interferem no processo;
- $T : S \times A \times S \rightarrow [0, 1]$ é uma função que quantifica a probabilidade de mudança do estado $s \in S$ para o estado $s' \in S$, dado a ação $a \in A$ selecionada. É representado por $T(s' | s, a)$;

- $r : S \times A \rightarrow \mathbb{R}$ é uma função que mede a recompensa por selecionar a ação $a \in A$ quando o processo está no estado $s \in S$.

De modo a encontrar um bom π , é necessário definir uma maneira de comparar diferentes políticas. Uma técnica comum é comparar a esperança da recompensa acumulada descontada, ou Valor de Retorno (R), que cada política produziu em um mesmo episódio.

$$R = E \left[\sum_{k=1}^z \gamma^{k-1} r_k \right] \quad (2.4)$$

Sendo z o tamanho do episódio, k o índice do *step* e o r_k a recompensa obtida em k . A constante γ é um valor entre 0 e 1, chamado de fator de desconto. Ele é responsável por atribuir um peso maior a recompensas obtidas no início do episódio.

Além disso, existem outras duas métricas importantes para se avaliar a qualidade de uma política: a Função de Valor (V) e a função de Valor da Ação (Q). A primeira estima a vantagem de estar em um estado s , calculando a esperança de R dado o estado inicial $s_0 = s$.

$$V_{\pi}(s) = E[R | s_0 = s] \quad (2.5)$$

A segunda estima a vantagem de selecionar a ação a estando no estado s . Em outras palavras, é definida como a esperança de R dado o estado inicial e a ação inicial:

$$Q_{\pi}(s, a) = E[R | s_0 = s, a_0 = a] \quad (2.6)$$

Neste trabalho vamos modelar tanto a política π quanto a Função de Valor V utilizando redes neurais profundas.

2.2.2 Redes Neurais

Inspirado no sistema neural humano, uma rede neural artificial consiste em um conjunto de nós, chamados de neurônios, interconectados por arestas [BISHOP, 2006](#). Em geral, é organizada em camadas, que incluem três tipos: a camada de entrada, a camada oculta e a camada de saída, conforme ilustrado na figura [2.9](#).

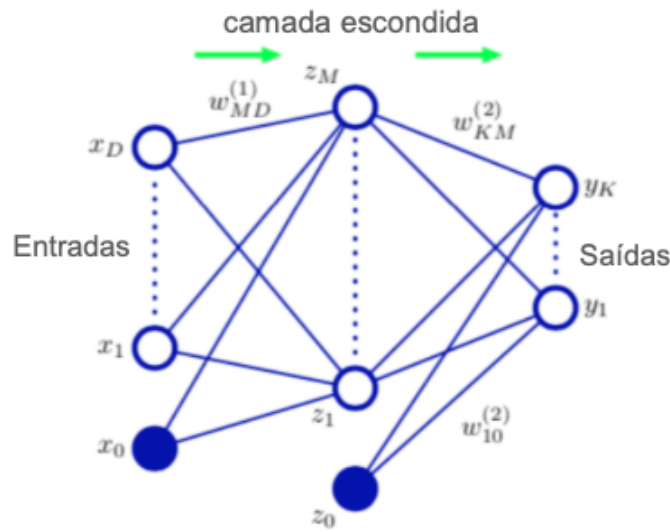


Figura 2.9: Arquitetura de uma rede neural - *BISHOP, 2006 adaptado*

Sinais fluem entre os neurônios através das arestas que estão associadas a um peso (w) que quantifica a sua importância. Nos neurônios é feito o processamento dos diversos sinais recebidos que, em seguida, aplica-se uma função, conhecida como função de ativação. Esta pode ser simplesmente uma combinação linear das entradas ponderadas pelos seus respectivos pesos, como pode ser função não linear como sigmoid, tangente hiperbólica, entre outras *BISHOP, 2006*.

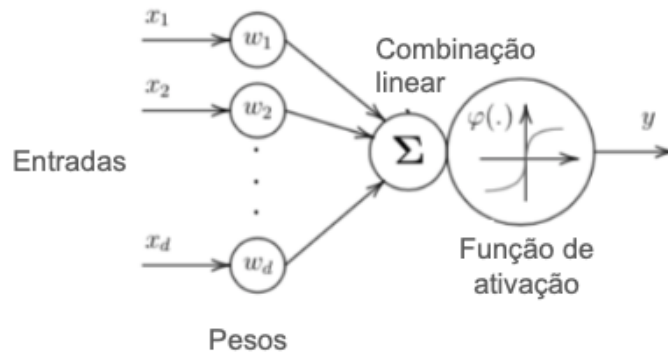


Figura 2.10: Neurônio processando sinal *FIGUEIREDO, 2018*

Os pesos das arestas são ajustados durante a fase de treinamento, de modo a orientar a rede a produzir a saída esperada para uma entrada específica. Nesta fase, é aplicado uma função de perda baseado na saída da rede. Existem diversas técnicas possíveis para minimizar a função de perda. Dentre elas, uma das mais tradicionais é a baseada no gradiente descendente, que atualiza os pesos conforme a equação a seguir.

$$w(t+1) = w(t) - n \nabla L(w(t)) \quad (2.7)$$

Onde w é o vetor de pesos, n é uma constante positiva conhecida como taxa de aprendizagem, que quantifica o tamanho do ajuste dos parâmetros a cada iteração e L é a função de perda. Um dos algoritmo mais utilizados para calcular o gradiente da função de perda e atualizar os parâmetros da rede é o *Backpropagation* BISHOP, 2006.

No contexto deste trabalho, as funções de perda estarão fortemente relacionadas ao valor de Retorno R e a função de Valor V apresentadas na seção anterior.

2.2.3 Método Actor-Critic

O método Ator-Crítico (AC) é um procedimento que tem como objetivo otimizar a política de um MDP KONDA e TSITSIKLIS, 2001. Ele é composto por dois componentes principais: o Ator e o Crítico. O Ator é aquele que seleciona as ações baseado nos estados, i.e, desempenha a função da política em um MDP. O Crítico, como o nome sugere, avalia a ação selecionada pelo Ator, estimando o vantagem de estar no estado alcançado. Assim, o Crítico é representado pela função de Valor do MDP. Tanto o Ator quanto o Crítico serão estimados por uma rede neural profunda cujos parâmetros serão ajustados utilizando a técnica *Proximal Policy Optimization* J. S. e. AL, 2017.

2.2.4 PPO

A técnica *Proximal Policy Optimization* é um modelo de aprendizado por reforço profundo que define funções de perda para ajustar os pesos das redes neurais pertencentes à arquitetura do método AC J. S. e. AL, 2017.

Uma vez que o Crítico objetiva estimar o valor de V , o ajuste dos seus parâmetros (W) é feito através de um gradiente descendente a fim de minimizar o quadrado da diferença entre a saída da rede ($\hat{V}(W)$) e o V observado (V^{obs}) J. S. e. AL, 2017:

$$L_t^{VF}(W) = (\hat{V}_t(W) - V_t^{obs})^2 \quad (2.8)$$

Uma das principais motivações do PPO é, durante o processo de otimização do MDP, evitar atualizações de política muito grandes. Para isto, J. S. e. AL, 2017 força com que a razão entre a política atual e a política anterior ($r_t(\Theta)$) seja limitada a um intervalo conveniente.

$$r_t(\Theta) = \frac{\pi_{\Theta}}{\pi_{\Theta_{old}}} \quad (2.9)$$

Desta forma, para ajustar os parâmetros (Θ) do Ator, J. S. e. AL, 2017 define a seguinte função objetivo a ser maximizada por um gradiente ascendente:

$$L^{CLIP}(\Theta) = \hat{E}_t[\min(r_t(\Theta)\hat{A}_t, \text{clip}(r_t(\Theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)] \quad (2.10)$$

Onde ϵ é uma constante que o J. S. e. AL, 2017 sugere ser igual a 0.2. \hat{A}_t é definido como a estimativa da função de vantagem, que corresponde ao benefício de selecionar uma ação a no estado s , em comparação com uma seleção aleatória de uma ação no estado s . Em outras palavras, é a diferença entre Q e V :

$$A = Q(s, a) - V(s) \quad (2.11)$$

A função objetivo $L^{CLIP}(\Theta)$ é a esperança do mínimo de dois termos: $r_t(\Theta)\hat{A}_t$ e $clip(r_t(\Theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t$. O primeiro direciona a política a selecionar ações que maximizam a função de vantagem. O segundo é uma versão truncada do primeiro, onde é aplicado um *clip* no $r_t(\Theta)$ de modo a garantir que a nova política não se distancie muito da política anterior.

Uma vez que o aprendizado das redes do AC são dependentes entre si, [J. S. e. AL, 2017](#) sugere uma função objetivo única para treinar ambos:

$$L_t^{PPO}(\Theta) = \hat{E}_t[L_t^{CLIP}(\Theta) - c_1 L_t^{VF}(\Theta) + c_2 S(s_t)] \quad (2.12)$$

Onde c_1 e c_2 são constantes e S é definido por [J. S. e. AL, 2017](#) como um bônus de entropia que garante que o agente explore suficientemente o ambiente durante o treinamento.

2.2.5 Escalonamento Multidimensional (MDS)

O uso de técnicas de redução de dimensionalidade, como o *Metric Multidimensional Scaling* (MDS), é motivado pela necessidade de construir representações vetoriais de objetos em um espaço de menor dimensão, preservando as distâncias definidas em uma matriz de similaridade ou dissimilaridade. No contexto deste trabalho, utilizamos o MDS para gerar uma representação vetorial para cada aminoácido, respeitando as distâncias previamente definidas na matriz proposta por [LOPES TIAGO J. S., 2021](#). Esta abordagem permite incorporar relações complexas de similaridade em uma forma compatível com métodos de aprendizado de máquina e análise estrutural.

Dado um conjunto de n objetos e uma matriz de distâncias $D = (d_{ij})$, onde cada elemento d_{ij} representa a distância entre o par de objetos i e j , o objetivo do MDS é encontrar uma configuração vetorial $X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ para cada objeto i em um espaço de dimensão m , de modo que a distância euclidiana entre X_i e X_j aproxime d_{ij} . O MDS busca minimizar as distâncias reais e as distâncias no espaço embutido, utilizando a função de perda conhecida como *stress*:

$$Stress(X_1, X_2, \dots, X_n) = \sqrt{\sum_{i \neq j=1}^n (d_{ij} - \|X_i - X_j\|)^2}. \quad (2.13)$$

A minimização desta função é geralmente realizada pelo algoritmo SMACOF (*Scaling by MAjorizing a Complicated Function*), que iterativamente melhora a configuração de pontos em direção a uma solução de menor *stress*, garantindo uma melhor correspondência entre as distâncias calculadas e as observadas [KRUSKAL, 1964](#). Embora outros métodos, como o gradiente descendente, possam ser utilizados, o SMACOF é amplamente preferido devido à sua eficiência e estabilidade.

Neste trabalho, a implementação do MDS foi realizada utilizando a biblioteca `sklearn`,

que fornece uma solução eficiente para problemas de escala multidimensional, baseada nas técnicas originalmente propostas por KRUSKAL, 1964.

Capítulo 3

Metodologia

Neste capítulo será detalhado os três módulos propostos na figura 4: Condições Iniciais, Treinamento e Geração de Sequências.

3.1 Condições iniciais

O objetivo do primeiro módulo (figura 3.1) é obter a sequência e o erro inicial. A primeira é obtida ao se passar a estrutura alvo como entrada do *ProteinMPNN*. Já o erro inicial é definido pela seguinte expressão:

$$Er_0 = 1 - TMScore(E_{FIX}, E_{ini}) \quad (3.1)$$

Onde E_{FIX} é a estrutura alvo, i.e, a estrutura do fator IX de coagulação e E_{ini} é a estrutura vinculada a sequência inicial, predita pelo *ESMFold* - descrito em 1.0.1. O *TMScore* consiste em uma medida de similaridade entre estruturas, detalhada em ??

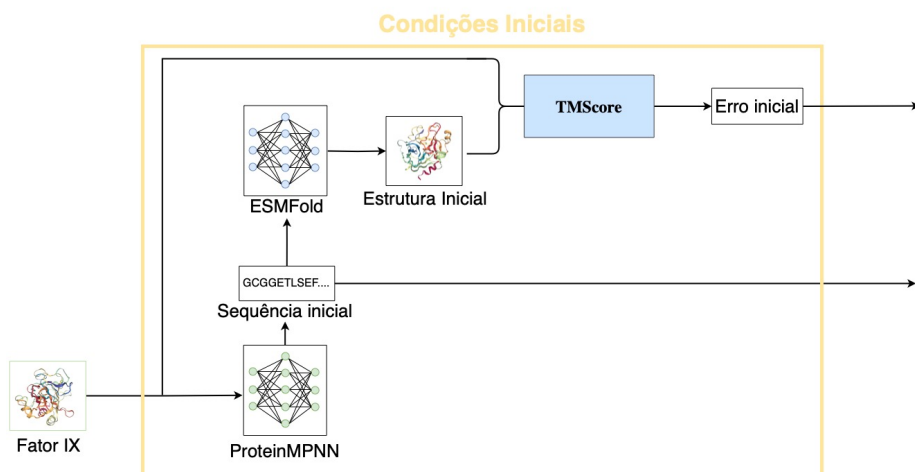


Figura 3.1: Obtenção das condições iniciais

3.2 Treinamento

O módulo de Treinamento foi construído em duas etapas: Estágios I e II, detalhados em 3.2.3 e 3.2.4 respectivamente. Para ambos os estágios, o treinamento se dá através de interações entre dois componentes: O Ambiente de aprendizado e o Agente.

3.2.1 Ambiente de aprendizado

O ambiente de aprendizado é modelado como um MDP e, portanto, é responsável por definir o espaço de ações, o espaço de estados, a função de transição entre estados dado uma ação e a função de recompensa.

Espaço de ações

Definimos uma ação como um vetor bi-dimensional composto pelo índice p de uma posição na sequência e um aminoácido r . A ação será utilizada para realizar uma mutação na proteína, inserindo o r na posição p . De modo a incentivar o Agente a explorar diferentes ações, definimos, para cada iteração, um conjunto de posições e aminoácidos válidos: P_{val} e R_{val} respectivamente. P_{val} é composto por todas as possíveis posições na sequência, com exceção das posições previamente selecionadas dentro do mesmo episódio. Definimos R_{val} sendo igual aos 20 possíveis aminoácidos, com exceção dos aminoácidos que foram selecionados mais de 7 vezes no mesmo episódio. Desta forma, o espaço de ações, para cada iteração i , é definido por:

$$A_i = \{ (p, r) | p \in P_{val_i}, r \in R_{val_i} \} \quad (3.2)$$

Espaço de estados

Definimos o espaço de estados como o conjunto de todas as possíveis sequências de n aminoácidos, onde n é o tamanho da sequência inicial:

$$S = \{ (r_1, r_2, r_3, \dots, r_n) | r_i \in R_{encoded} \forall i \in [1, n] \} \quad (3.3)$$

Onde $R_{encoded}$ é o conjunto dos 20 possíveis aminoácidos, cada um representado por um vetor codificado pelo *Encoder*. O *Encoder* consiste em um dicionário que mapeia cada aminoácido para uma representação vetorial que conserva as distâncias entre cada par de aminoácidos estabelecidas por [LOPES TIAGO J. S., 2021](#). Este mapeamento foi construído através de um MDS utilizando 21 dimensões.

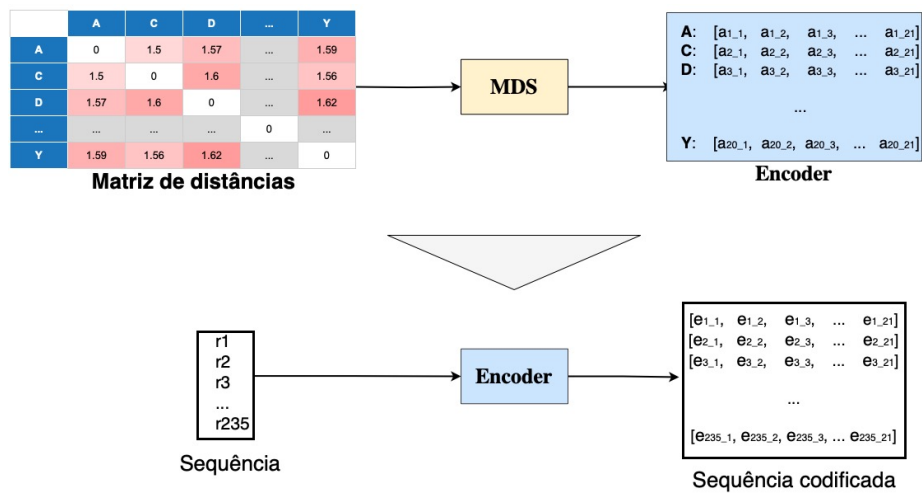


Figura 3.2: Construção e uso do Encoder.

Para escolher a quantidade de dimensões a se utilizar no MDS, definimos o ponto a partir do qual a curva entre *Stress* e dimensionalidade tende a se aproximar assintoticamente do eixo x do plano cartesiano. Este ponto, conforme a figura 3.3, é o que possui 21 dimensões.

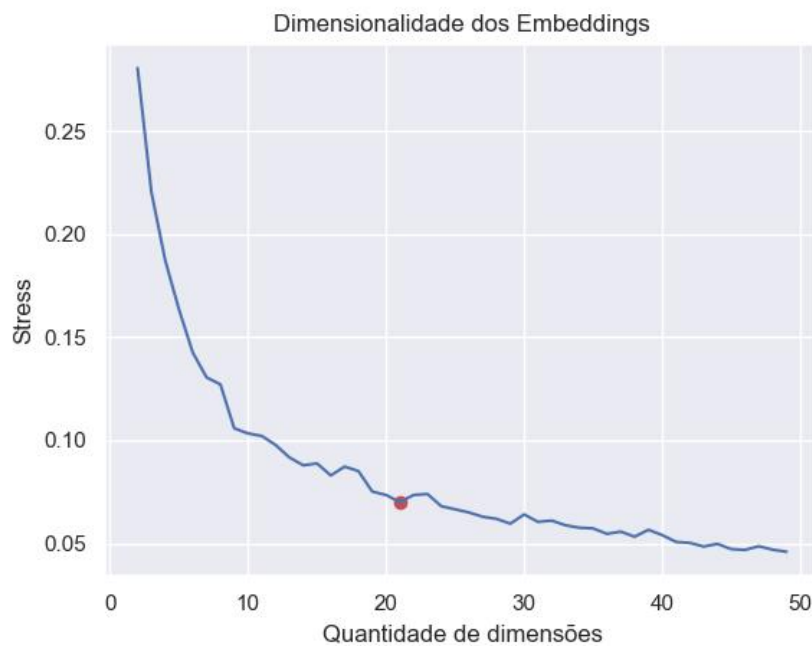


Figura 3.3: Escolha da dimensionalidade dos Embeddings

Função de transição de estados

A função de transição de estados é caracterizada pela geração de uma nova sequência, resultante da mutação definida pelo Agente. Em outras palavras, é uma função que tem como entrada uma ação e o estado atual e como saída o estado resultante da transição.

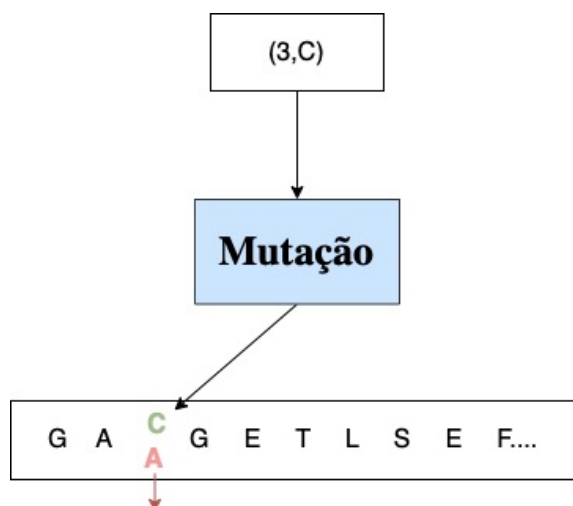


Figura 3.4: Exemplo de muta  o. O amino cido A na pos   o 3 est  sendo substituído pelo o amino cido C

Fun  o de recompensa

J  a fun  o de recompensa calcula a qualidade da a  o selecionada pelo Agente. Nas se   es 3.2.3 e 3.2.4 especificamos a fun  o de recompensa de cada est gio de treinamento.

3.2.2 Agente

O Agente   modelado por uma rede neural, batizada de *GenSeq*, cujos par metros s o otimizados atrav s da t cnica *Proximal Policy Optimization* (PPO), sendo respons vel por decidir qual a melhor muta  o a se fazer, dada a seq  ncia de entrada. Em outras palavras, o Agente deve escolher, dentro do espa o de a   es, a a  o que maximiza a recompensa acumulada. Os hiperpar metros utilizados no Agente s o apresentados na tabela a seguir.

Hiperpar�metro	Valor
N�mero de camadas escondidas	2
N�mero de neur�nios por camada escondida	64
Fun��o de ativa���o	tangente hiperb�lica
Otimizador	Adam
Taxa de aprendizagem	3e-4
N�mero de �pocas	10
Fator de desconto (γ)	0.99

Tabela 3.1: Hiperpar metros do PPO

Ap s ser processada pelo *Encoder*, a seq  ncia passa pela camada de entrada do *GenSeq* e segue para duas camadas escondidas de 64 dimens  es cada. A camada de sa  da, por sua vez, possui 255 unidades, organizadas em duas partes:

1. As primeiras 235 unidades da camadas estão associadas ao primeiro atributo da ação: A posição p da sequência de entrada que deve sofrer mutação. A posição predita pelo Agente corresponde ao índice da unidade de saída com maior valor entre as 235 primeiras.
2. Já as 20 últimas unidades de saída estão associadas ao segundo atributo da ação: O índice do aminoácido r que o Agente sugere inserir na posição p . Este valor corresponde ao índice da unidade de saída com maior valor entre as 20 últimas unidades, subtraído de 235.

Por exemplo, suponha que, entre as 235 primeiras unidades, a de maior valor é a terceira, e que, entre as 20 últimas, o maior valor esteja na unidade 240. Nesse caso, a ação sugerida pelo Agente é fazer uma mutação na posição 3, inserindo o aminoácido de índice 5.

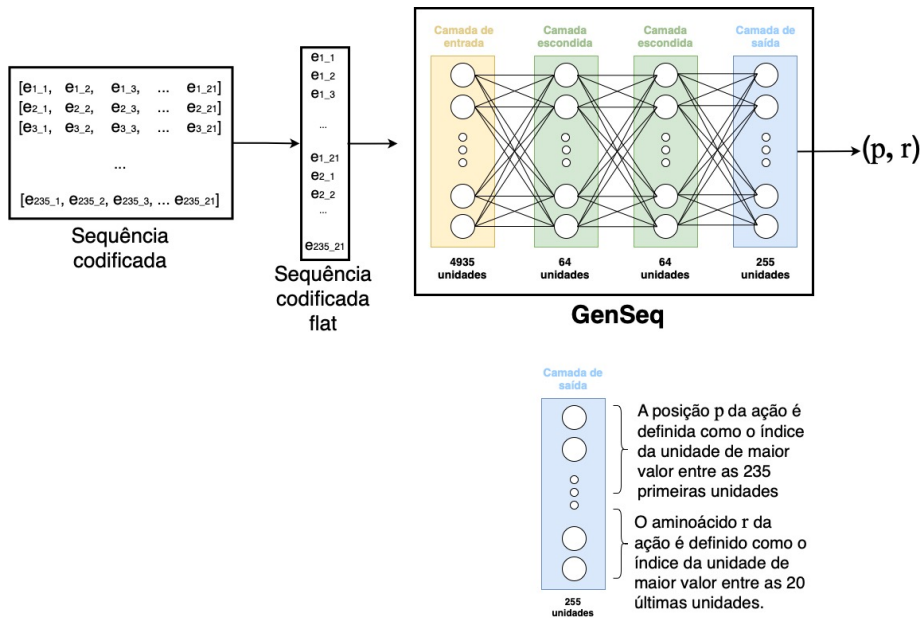


Figura 3.5: Arquitetura da rede neural do Agente (GenSeq)

3.2.3 Treinamento - Estágio I

Tendo em vista o vasto número de ações possíveis, desenvolvemos o Estágio I de treinamento com o objetivo de estimular o Agente a encontrar um subespaço menor de ações para atuar. Para isso, atribuímos recompensas negativas (ou penalizações) para ações em posições da sequência cujo o *Conservation Score* é alto, bem como penalizamos ações onde o aminoácido substituído é muito diferente do que fora inserido. Neste sentido, definimos a seguinte função de recompensa:

$$R_I = -(CS(p) + AminoDist(r, r_{prev})) \quad (3.4)$$

Onde p e r são a posição e o aminoácido selecionados pelo Agente, respectivamente. r_{prev} é o aminoácido que estava na posição p antes da mutação. $CS(p)$ é o *Conservation*

Score normalizado da posição p e $AminoDist(r, r_{prev})$ é a medida de similariedade entre os aminoácidos r e r_{prev} .

Neste sentido, o primeiro estágio de treinamento consiste do seguinte procedimento:

1. Os pesos do *GenSeq* são inicializados aleatoriamente.
2. Inicializa o episódio com a sequência inicial e com P_{val} sendo todas as posições da sequência e R_{val} sendo todos os 20 aminoácidos
3. A sequência atual é codificada pelo *Encoder*.
4. *GenSeq* estima a melhor ação (p, r) baseado na sequência codificada.
5. A sequência atual é modificada ao inserir r na posição p .
6. Baseado em (p, r) e r_{prev} , é calculado R_I .
7. p é removido de P_{val} e se a quantidade de vezes que r foi utilizado pelo *GenSeq* é maior ou igual a 7, então r é retirado de R_{val} .
8. Se a iteração atual é igual ao tamanho do *batch*, então *GenSeq* atualiza seus pesos baseado nos valores obtidos de R_I utilizando o *PPO*
9. Se foi atingido o máximo de mutações por episódio, então o episódio é finalziado.
10. Repete a partir do item 2 até atingir o número máximo de iterações.

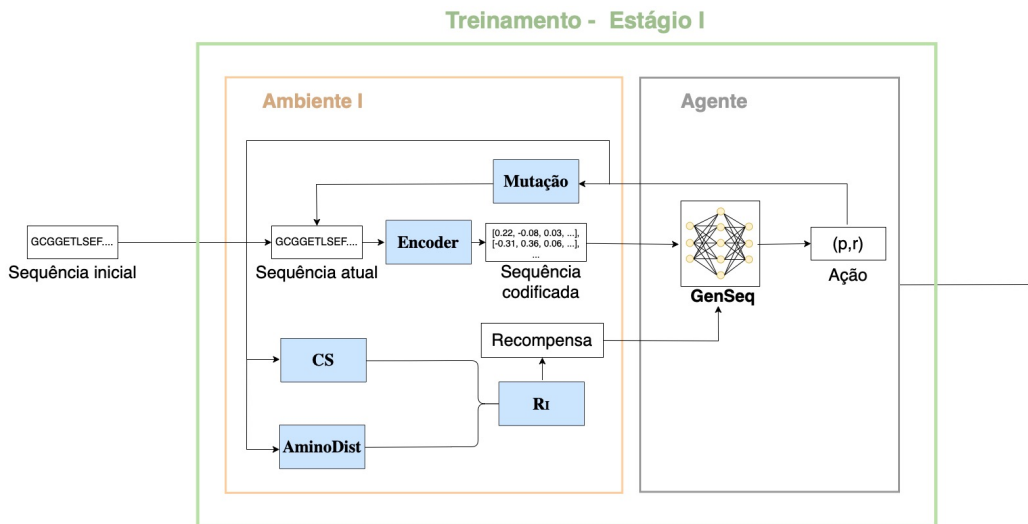


Figura 3.6: Primeiro estágio de treinamento

Neste estágio, o Agente foi treinado com um total de 2 milhões de iterações, podendo executar até 100 mutações por episódio.

Parâmetro	Valor
Número máximo de mutações por episódio	100
Número de iterações	2.000.000
Tamanho do <i>batch</i>	200

Tabela 3.2: *Parâmetros preliminares*

Após finalizado o primeiro estágio, comparamos o desempenho do Agente com o desempenho de uma política que define ações aleatoriamente, onde observamos que a mediana da distribuição de recompensas do Agente é consideravelmente superior ao do Agente aleatório.

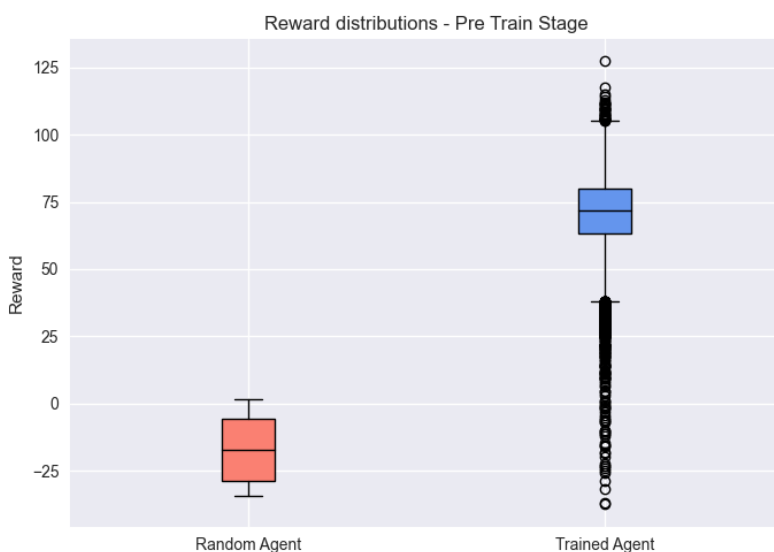


Figura 3.7: *Comparação do Agente pré treinado com um Agente aleatório*

Observa-se que o Agente atingiu um platô de recompensas no ambiente I com cerca de 2500 episódios, como ilustrado em 3.8.

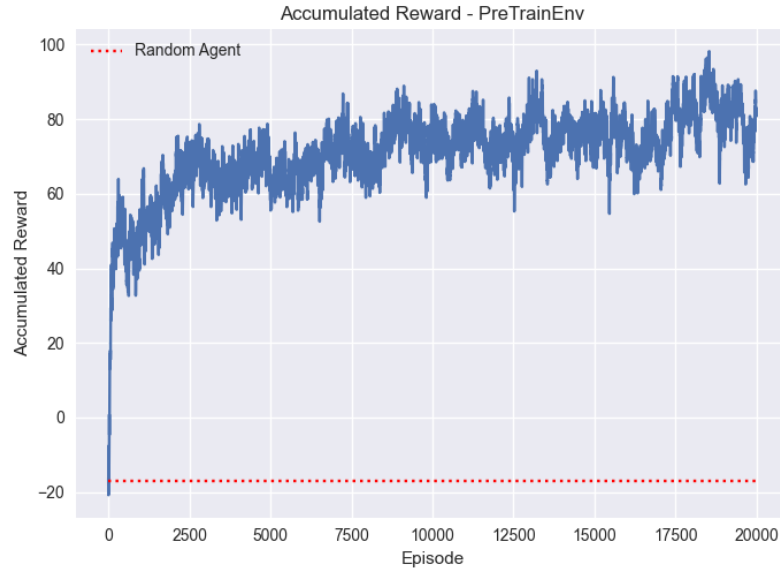


Figura 3.8: Média móvel das recompensas por episódio - treino

3.2.4 Treinamento - Estágio II

O objetivo do segundo estágio de treinamento é fazer com que o agente pré treinado aprenda a escolher ações que maximizem o *TM-Score* entre a estrutura gerada e a alvo, i.e, minimizem o erro $1 - TMScore$. Para isto, definimos a função de recompensa como a soma de 3 termos:

$$R_{II} = -aEr_i + b(Er_0 - Er_i) + c(Er_{i-1} - Er_i) \quad (3.5)$$

Onde Er_i e Er_{i-1} são os erros da iteração i e $i-1$ respectivamente, e a , b e c são constantes. O primeiro termo, $-aEr_i$, é a penalização pelo erro atual. O segundo determina uma penalização se o erro atual for maior que o inicial e uma recompensa caso contrário. Análogamente, o terceiro termo penaliza se o erro atual for maior que o anterior e recompensa se for menor. Como o objetivo principal é encontrar sequências com erro menores que o inicial, atribuímos um peso maior ao segundo termo, i.e, $b > a$ e $b > c$. Além disso, consideramos $c > a$ para valorizar reduções do erro ao longo do episódio.

Introduzimos também o conceito de vitória e derrota a cada episódio. Consideramos vitória caso o erro final seja menor que o inicial e derrota caso contrário. De modo a valorizar a vitória e penalizar a derrota, ao final de cada episódio é adicionado uma recompensa adicional dada por $2500(Er_0 - Er_f)$, onde Er_f é o erro final, i.e, da última iteração do episódio. Cada episódio é finalizado ao se atingir um número máximo de iterações, ou um limiar de erro. Foi definido um limite superior para o erro de modo a evitar que o Agente atinja erros muito grandes em um mesmo episódio, e um limite inferior para valorizar episódios em que o Agente foi capaz de encontrar "atalhos", i.e, conseguiu reduzir o erro consideravelmente em poucas iterações.

Neste sentido, o segundo estágio de treinamento consiste do seguinte procedimento:

1. Inicializa o episódio com a sequência e o erro inicial; P_{val} sendo todas as posições da sequência; R_{val} sendo todos os 20 aminoácidos.
2. A sequência atual é codificada pelo *Encoder*.
3. *GenSeq* estima a melhor ação (p, r) baseado na sequência codificada.
4. A sequência atual é modificada ao inserir r na posição p .
5. É predito a estrutura da sequência atual utilizando o *ESMFold*
6. Baseado no *TMScore* entre a estrutura do Fator IX e da estrutura atual, no erro inicial e no erro da iteração anterior, é calculado R_{II} .
7. p é removido de P_{val} e, se a quantidade de vezes que r foi utilizado pelo *GenSeq* é maior ou igual a 7, então r é retirado de R_{val} .
8. Se a iteração atual é igual ao tamanho do *batch*, então *GenSeq* atualiza seus pesos baseado nos valores obtidos por R_{II} utilizando o *PPO*
9. Se foi atingido o máximo de mutações por episódio, ou um limiar de erro, então o episódio é finalizado e é atribuído a recompensa adicional igual a $2500(Er_0 - Er_f)$.
10. Repete a partir do item 1 até atingir o número máximo de iterações.

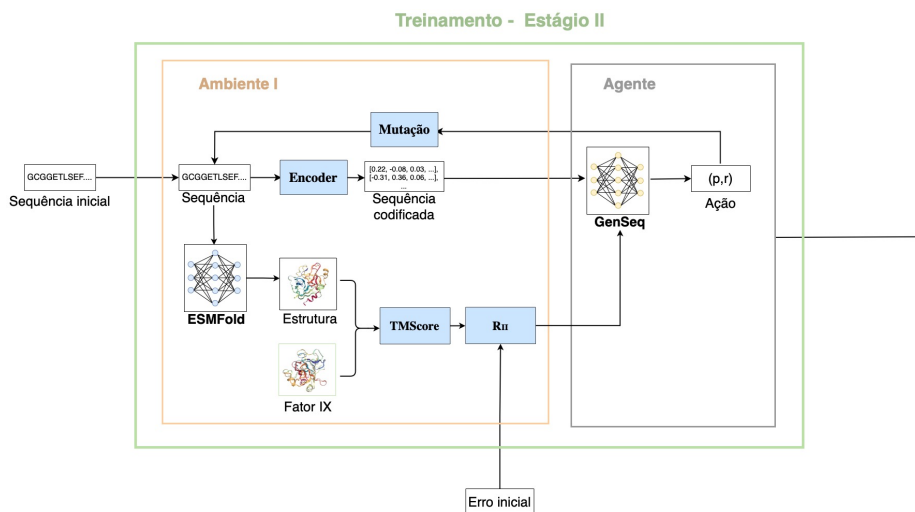


Figura 3.9: Segundo estágio - Treino

Neste estágio, o Agente foi treinado com um total de 60 mil de iterações, podendo executar até 45 mutações por episódio.

Parâmetro	Valor
Número máximo de mutações por episódio	45
Erro mínimo no episódio	0.05
Erro máximo no episódio	0.083
Número de iterações	60.000
a	12
b	200
c	50

Tabela 3.3: *Parâmetros preliminares*

Assim como no pré treino, a distribuição das recompensas do Agente ao longo do treinamento é bem superior às recompensas de um Agente aleatório.

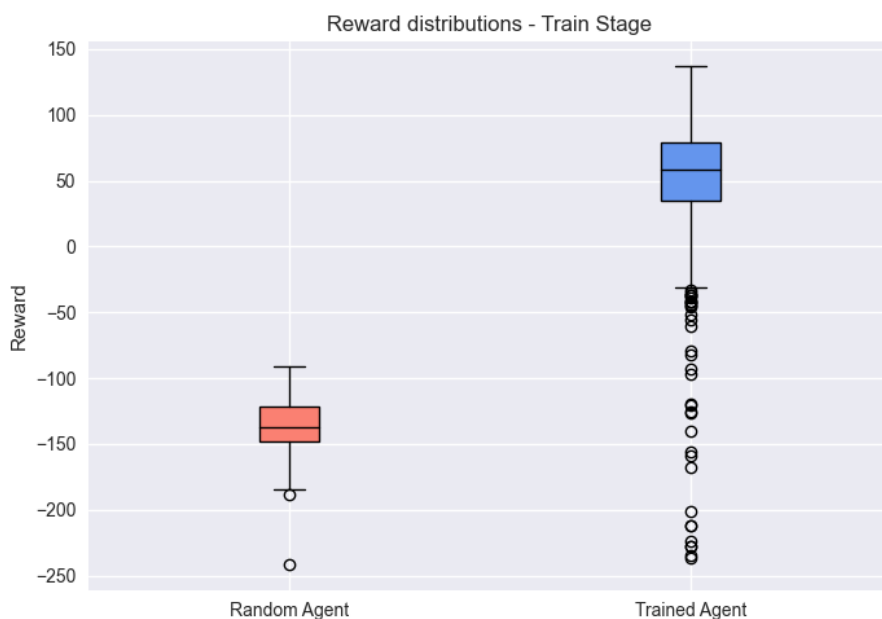


Figura 3.10: *Comparação do Agente treinado com um Agente aleatório*

Repare que, devido ao pré treino, as recompensas médias do Agente são consideravelmente superiores à recompensa média do Agente aleatório desde os primeiros episódios.

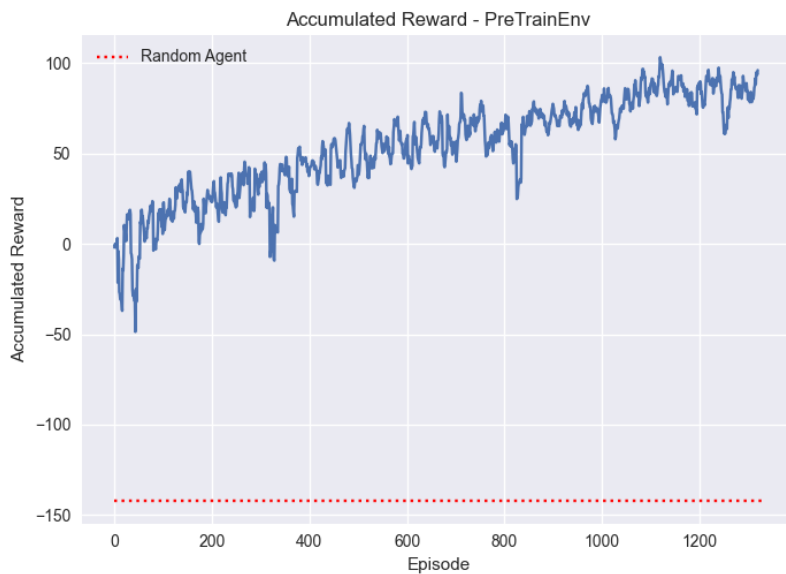


Figura 3.11: Média móvel das recompensas por episódio - treino

3.3 Geração de sequências

Por fim, o módulo de Geração de Sequências utiliza o agente treinado para gerar novas sequências que sejam capazes de mimetizar a estrutura alvo melhor do que a sequência inicial, em termos de *TM-Score*.

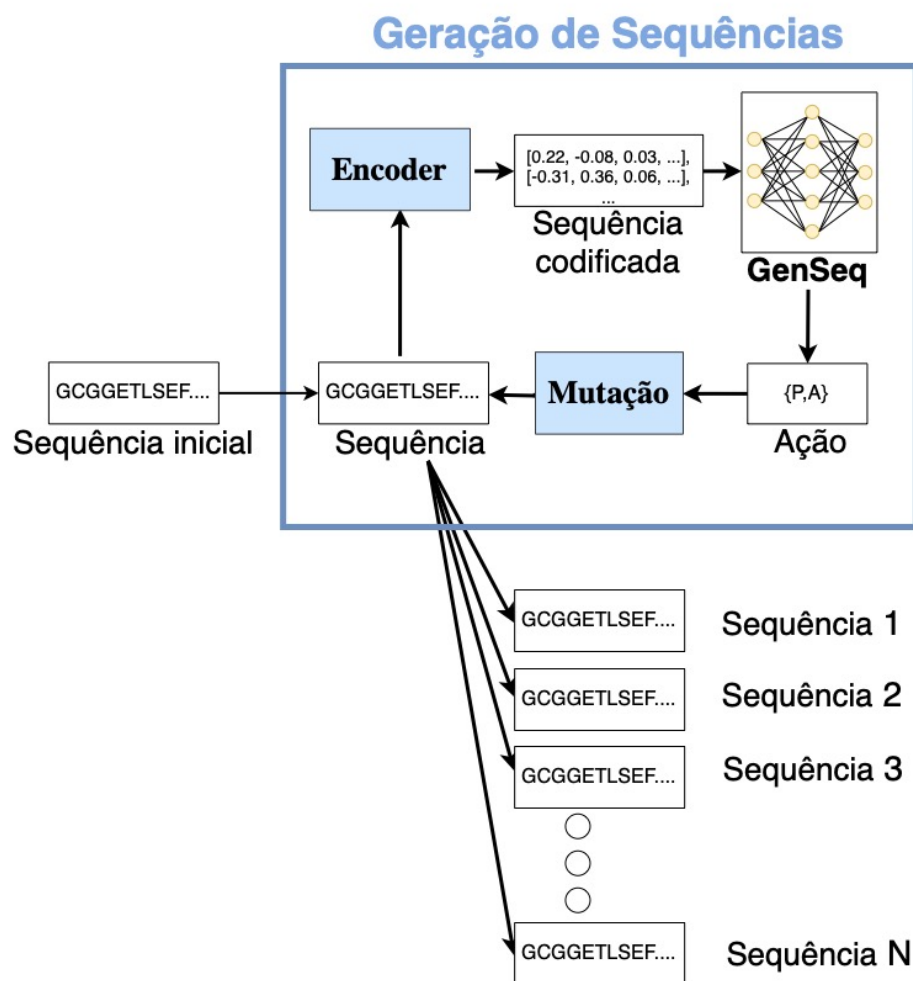


Figura 3.12: Gerando sequência otimizada

Capítulo 4

Resultados

Em andamento

Com o Agente treinado, é possível gerar novas sequências seguindo a arquitetura ilustrada na figura 3.12. O Agente gerou mais de 200 proteínas diferentes a partir de mutações na sequência inicial. De modo a identificar cada proteína, vamos considerar a ordem com que foi gerada como seu respectivo ID. Filtramos as proteínas geradas baseado em um limiar de 92% de TMScore. Com isso, selecionamos as 63 primeiras sequências.

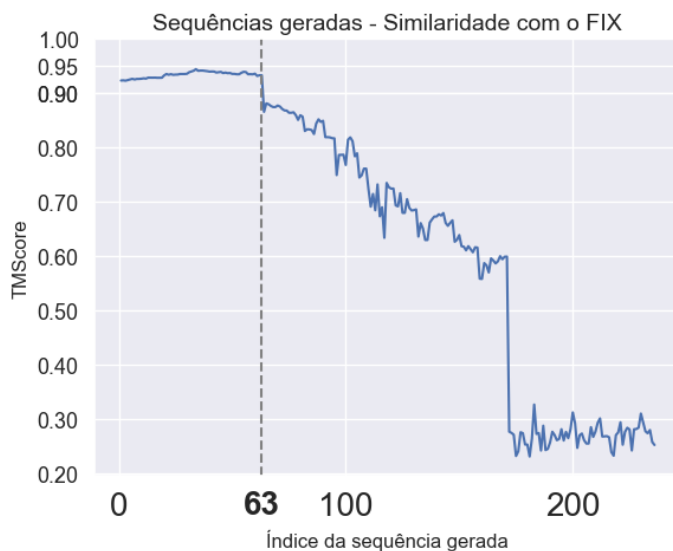


Figura 4.1: Variação da similaridade entre as sequências geradas e o FIX

A proteína gerada que é estruturalmente mais similar ao FIX é a de ID 34, tendo TMScore igual a 94.48% e RMSD de 1.485 angstroms.

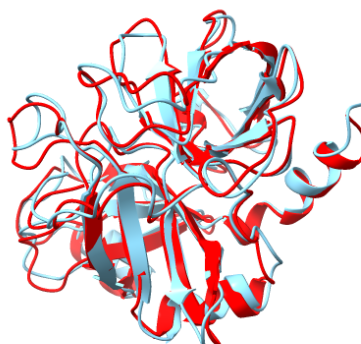


Figura 4.2: Alinhamento entre FIX (em azul) e a proteína ID 34 (em vermelho)

É interessante notar que, apesar da forte semelhança estrutural entre as proteínas, em termos de IS (Idêntidade de Sequência) são sequências muito diferentes, tendo em média apenas 30% de similaridade.

ID	TM-Score	RMSD	IS - seq. original
20	93.33%	1.75 Å	30.64%
30	93.62%	1.68 Å	30.21%
34	94.48%	1.48 Å	29.79%
63	93.36%	1.72 Å	26.81%

Tabela 4.1: Generated proteins

4.1 Resposta imunológica

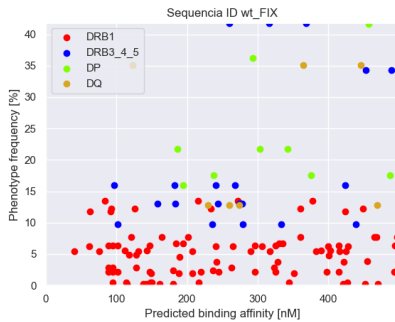
Corrigir Para mensurar a propensão das proteínas geradas provocarem uma resposta imunológica no organismo, calculamos o *binding affinity* de cada proteína com os epitopes de quatro grupos diferentes de alelos apresentados na tabela 4.2

Grupo	Frequência em seres humanos
DP	25%
DQ	20%
DRB1	5%
DRB3_4_5	21%

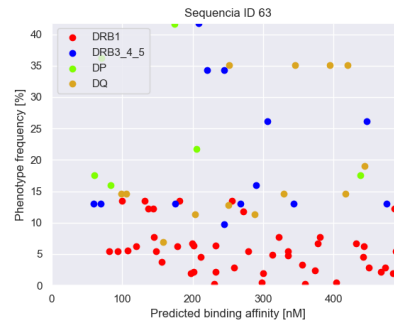
Tabela 4.2: Grupos de alelos considerados e a frequência com que ocorrem em seres humanos.

Para esta análise, levamos em consideração apenas epitopes associados a valores de *binding affinity* menores que 500nm, indicando ligação forte. Nas figuras a seguir é possível visualizar a quantidade, *binding affinity* e frequência de ocorrência dos epitopes por cada grupo de alelo. Ao se comparar o resultado do FIX e da proteína de ID 63, é possível

notar uma redução significativa da quantidade de epítopes com ligação forte, influenciado principalmente pelo resultado do grupo DRB1.



(a) Resposta imune FXI



(b) Resposta imune ID 63

A propensão a provocar uma resposta imune não está apenas relacionada a quantidade de epítopes, mas também é diretamente proporcional a frequência de ocorrência dos alelos e inversamente proporcional ao *binding affinity*. Neste sentido, para mensurar qual proteína induz a uma melhor resposta imune, propomos a métrica *IM*, que consiste na média da razão entre *binding affinity* (Af_i) e frequência (Fr_i), dividido pela quantidade de epítopos (N):

$$IM = \frac{1}{N} \sum_{i=1}^N \frac{1}{N} \frac{Af_i}{Fr_i} \quad (4.1)$$

Desta forma, a proteína associada ao maior IM possui o menor risco de induzir a uma resposta imune, i.e, possui a resposta imune mais favorável. De modo geral, as 4 últimas sequências geradas possuem um IM superior ao do FIX: IDs 60, 61, 62 e 63.

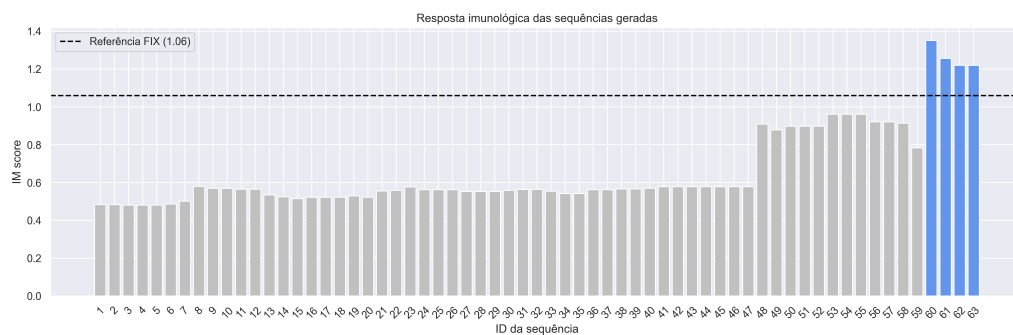


Figura 4.4: Comparação do IM entre as proteínas geradas e o FIX

Calculando o IM para cada grupo de alelo separadamente, observa-se que grande parte das sequências geradas possuem uma resposta imune mais favorável que o FIX para os grupos DRB3, DRB4 e DRB5. Para o grupo DRB1, as sequências a partir do ID 48 em diante obtiveram IM superior ao FIX. Já para os grupos DQ e DP, nenhuma sequência superou o IM do FIX.

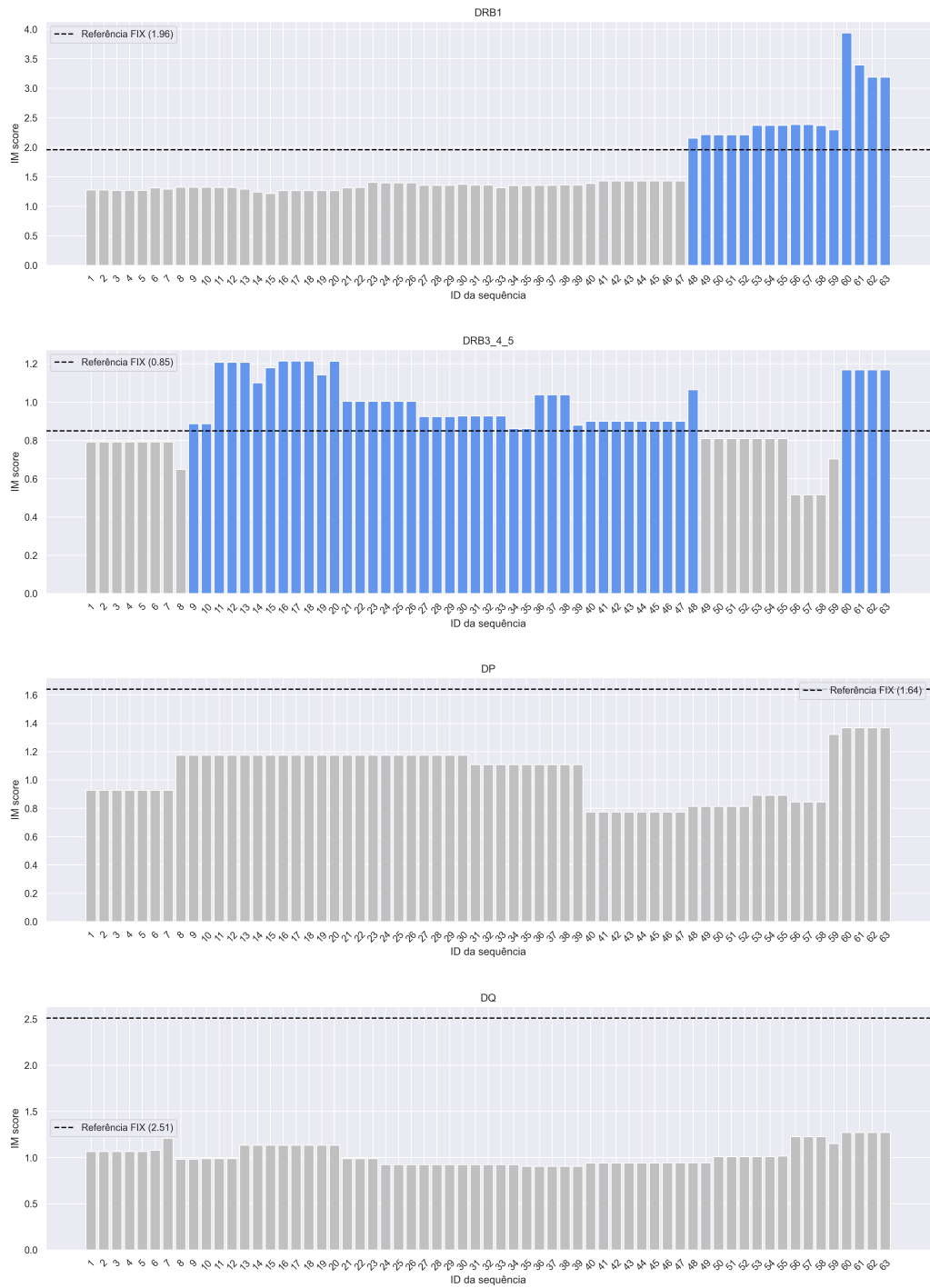


Figura 4.5: IM calculado para cada grupo de alelo

4.2 Docking

Explicar Docking no background, explicando a interação entre FIX e FVIII. Explicar as métricas Corrigir os percentuais com as novas imagens

Ao projetar proteínas capazes de substituir o FIX no tratamento da Hemofilia B, um aspecto crítico é garantir que as proteínas geradas possuem uma interação eficaz ou até mais favorável com o FVIII se comparado ao FIX. Para isto, realizamos análises de *docking* molecular de cada sequência gerada com o FVIII, calculando métricas que capturam diferentes características da interface proteína-proteína.

- *Contact Molecular Surface*: obtivemos 19 proteínas (30% do total) superiores ao FIX em termos de CMS, sugerindo interfaces de ligação maiores para essas sequências.
- *DDG*: cerca de 32% das proteínas geradas atingiram um DDG mais favorável (menor) do que o FIX, indicando ligações mais estáveis.
- *Interface Buried Sasa*: 33% das proteínas obtiveram um SASA superior ao FIX.
- *SAP Score Target*: apenas 2% das sequências projetadas atingiram valores superiores ao FIX.

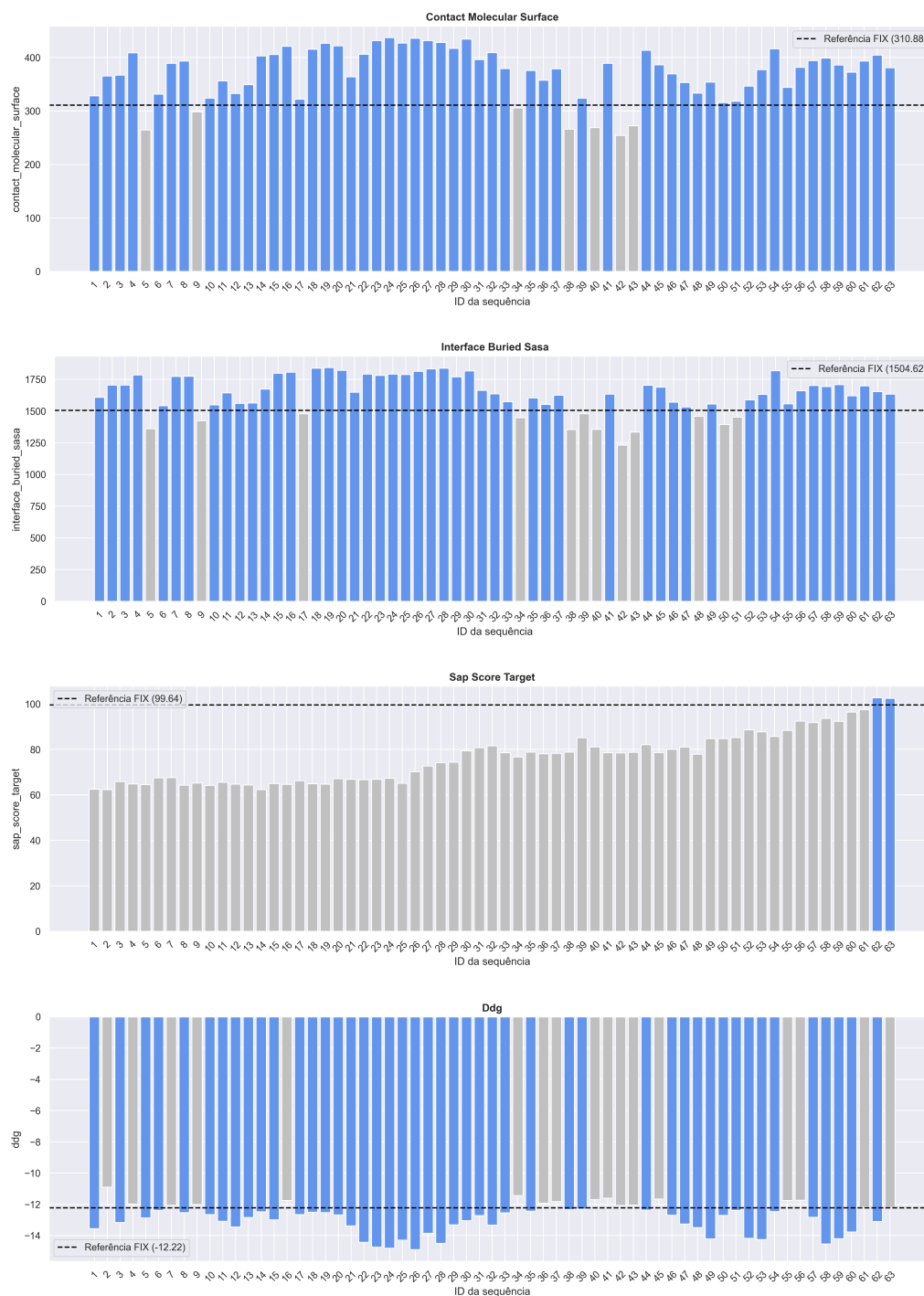


Figura 4.6: Resultado do Docking entre as proteínas geradas e o fator VII

Capítulo 5

Conclusão

Em andamento

Referências

- [J. D. e. AL 2022] J. Dauparas et AL. “Robust deep learning–based protein sequence design using proteinmpnn”. *Science* (2022) (citado na pg. 3).
- [J. S. e. AL 2017] John Schulman et AL. “Proximal policy optimization algorithms”. *ArXiv* (2017) (citado nas pgs. 20, 21).
- [M. H. B. e. AL 2024] Mettine H.A. Bos et AL. “Blood coagulation factor ix: structural insights impacting hemophilia b therapy”. *American Society for Hematology Journal BLOOD* (2024) (citado nas pgs. 11, 12).
- [ANISHCHENKO 2021] et al. ANISHCHENKO I. “De novo protein design by deep network hallucination”. *Nature* (2021) (citado na pg. 3).
- [B 2019] Kuhlman B. “Designing protein structures and complexes with the molecular modeling program rosetta”. *J Biol Chem* (2019) (citado na pg. 3).
- [B. ALBERTS 2002] et al. B. ALBERTS. *Molecular Biology of the Cell*. Garland Science, 2002 (citado nas pgs. 9–11).
- [BEN CHORIN 2020] A. et al BEN CHORIN. “Consurf-db: an accessible repository for the evolutionary conservation patterns of the majority of pdb proteins”. *Protein Sci* (2020) (citado na pg. 16).
- [BISHOP 2006] M. Christopher BISHOP. *Pattern Recognition and Machine Learning*. New York: Springer, 2006 (citado nas pgs. 18–20).
- [BJOERN PETERS e SETTE 2020] Morten Nielsen BJOERN PETERS e Alessandro SETTE. “T cell epitope predictions”. *Annual Reviews* (2020) (citado na pg. 14).
- [CHARIF 2007] J. CHARIF D. Lobry. “R. in structural approaches to sequence evolution”. *Springer* (2007) (citado na pg. 16).
- [COVENTRY 2021] Brian COVENTRY. “Learning How to Make Mini-Proteins that Bind to Specific Target Proteins”. Tese de dout. University of Washington, 2021 (citado na pg. 13).
- [EDDY 2004] S. R. EDDY. “Where did the blosum62 alignment score matrix come from?” *Nature Biotechnology* (2004) (citado na pg. 16).

- [FIGUEIREDO 2018] Jonathas FIGUEIREDO. “Aplicação de algoritmos de aprendizagem por reforço para controle de navios em águas restritas”. *Universidade de São Paulo* (2018) (citado na pg. 19).
- [GOUW 2013] et al. GOUW S. C. “Factor viii products and inhibitor development in severe hemophilia a”. *New England Journal of Medicine* (2013) (citado na pg. 1).
- [KONDA e TSITSIKLIS 2001] Vijay KONDA e John TSITSIKLIS. “Actor-critic algorithms”. *Society for Industrial and Applied Mathematics* 42 (abr. de 2001) (citado na pg. 20).
- [KRUSKAL 1964] J.B. KRUSKAL. “Nonmetric multidimensional scaling: a numerical method”. *Psychometrika* (1964) (citado nas pgs. 21, 22).
- [LOPES TIAGO J. S. 2021] Rios Ricardo et al. LOPES TIAGO J. S. Nogueira Tatiane. “Prediction of hemophilia a severity using a small-input machine-learning framework”. *Nature* (2021) (citado nas pgs. 7, 16, 17, 21, 24).
- [MANCUSO 2005] et al. MANCUSO M. E. “Environmental risk factors for inhibitor development in children with hemophilia a: a case-control study”. *Haematologica* (2005) (citado na pg. 1).
- [PELLEGRINI 2007] Jerônimo PELLEGRINI. *Processos de Decisão de Markov: um tutorial*. XIV. RITA, 2007 (citado na pg. 17).
- [PM 2020] Mannucci PM. “Emophilia therapy: the future has begun.” *Haematologica* (2020) (citado na pg. 1).
- [WENZE DING e GONG 2022] Kenta Nakai WENZE DING e Haipeng GONG. “Protein design via deep learning”. *Oxford Academic* (2022) (citado na pg. 1).
- [ZHANG 2022] et al. ZHANG C. “Us-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes”. *Nat Methods* (2022) (citado na pg. 16).
- [ZHANG 2004] et al. ZHANG Y. “Scoring function for automated assessment of protein structure template quality”. *Proteins: Structure, Function, and Bioinformatics* (2004) (citado nas pgs. 14, 15).