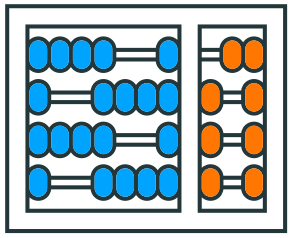


Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

# Classificação de Estruturas e Funções de Proteínas

## MO640 - Biologia Computacional / MC668 - Bioinformática

Gabriel Bianchin de Oliveira

2024

Instituto de Computação

# Roteiro

Proteínas

Transformers

Alinhamento Local

Classificação de Estruturas Secundárias

Classificação de Estruturas Terciárias

Classificação de Funções

# Proteínas

# Proteínas

- Proteínas são macromoléculas essenciais para as atividades biológicas.
- Elas são geradas utilizando informação codificada nos genes.
- Proteínas são formadas por aminoácidos, unidos por ligações peptídicas.

# Aminoácidos

- Cada aminoácido é formado por um carbono alfa, ligado a quatro grupos:
  - Grupo amino ($NH_2$).
  - Grupo carboxila (COOH).
  - Átomo de hidrogênio.
  - Cadeia lateral variável.
- As ligações peptídicas são formadas entre o átomo de carbono do grupo carboxila de um aminoácido e o átomo de nitrogênio do grupo amino do aminoácido adjacente.

# Aminoácidos

Ao todo, existem 20 aminoácidos:

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

**Tabela 1:** Lista dos vinte aminoácidos e as siglas correspondentes.

# Estruturas das Proteínas

As proteínas possuem quatro estruturas:

- Estrutura primária: sequência de aminoácidos.
- Estruturas secundárias: estrutura tridimensional de cada aminoácido.
- Estrutura terciária: estrutura geral tridimensional da proteína.
- Estrutura quaternária: complexo de duas ou mais proteínas.

# Estruturas das Proteínas

Por que estudar estruturas tridimensionais das proteínas?

- O sequenciamento se tornou barato nas últimas décadas.
- Determinar estruturas tridimensionais é custoso:
  - Cristalografia de radiografia.
  - Espectroscopia de ressonância magnética nuclear.
  - Microscopia eletrônica.
- Estruturas tridimensionais impactam na função da proteína.

# Estruturas das Proteínas

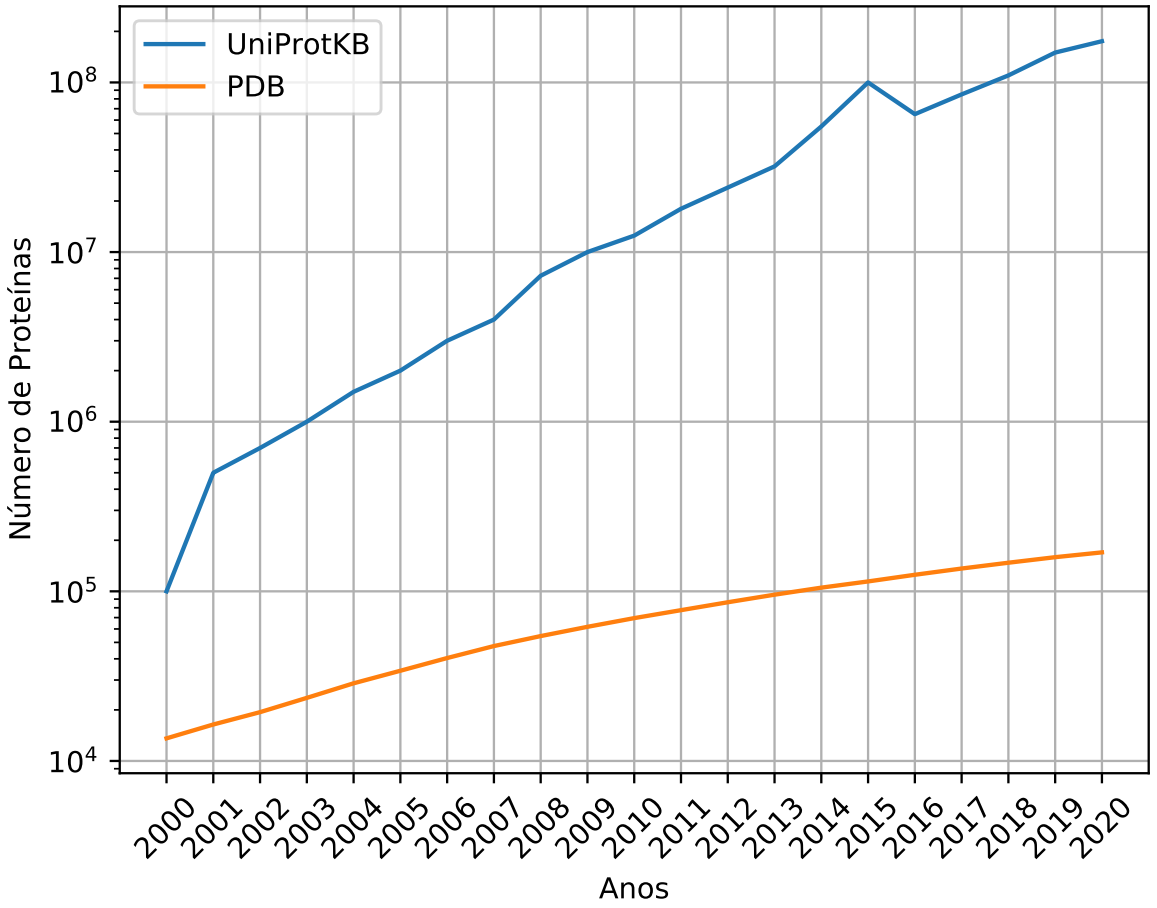


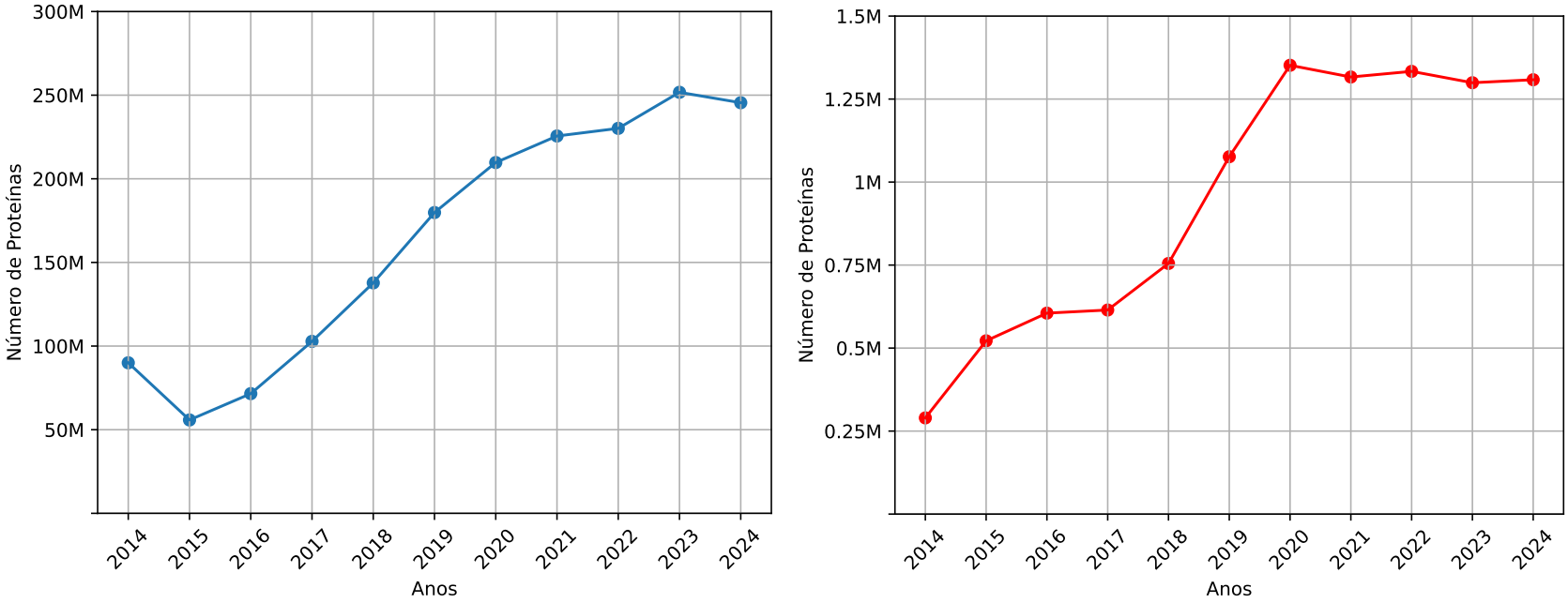

**Figura 1:** Número de proteínas sequenciadas na base UniProtKB e proteínas com estruturas secundárias definidas na base PDB entre 2000 e 2020.

# Funções das Proteínas

Por que estudar funções das proteínas?

- Assim como a análise de estruturas tridimensionais, funções são custosas de serem analisadas.
- Funções indicam o que as proteínas fazem.
- Desenvolvimento de novas aplicações:
  - Medicamentos.
  - Biossensores.
  - Estudo de enzimas.

# Funções das Proteínas




**(a)** Proteínas sequenciadas no UniProtKB. **(b)** Proteínas analisadas manualmente.

**Figura 2:** Quantidade de proteínas sequenciadas na base UniProtKB e proteínas analisadas manualmente na base GOA.

# Métodos Computacionais

Devido ao custo dos métodos de análise em laboratório, métodos computacionais vem sendo aplicados para a análise de estruturas tridimensionais e determinação de funções:

- Aprendizado de Máquina com foco em Transformers.
- Alinhamento Local.

# Transformers

# Redes Recorrentes

- Modelos de aprendizado de máquina para sequências:
  - Redes neurais recorrentes (RNN).
  - Módulos de memória (LSTM e GRU).
  - Redes recorrentes bidirecionais (BRNN).

# Redes Recorrentes

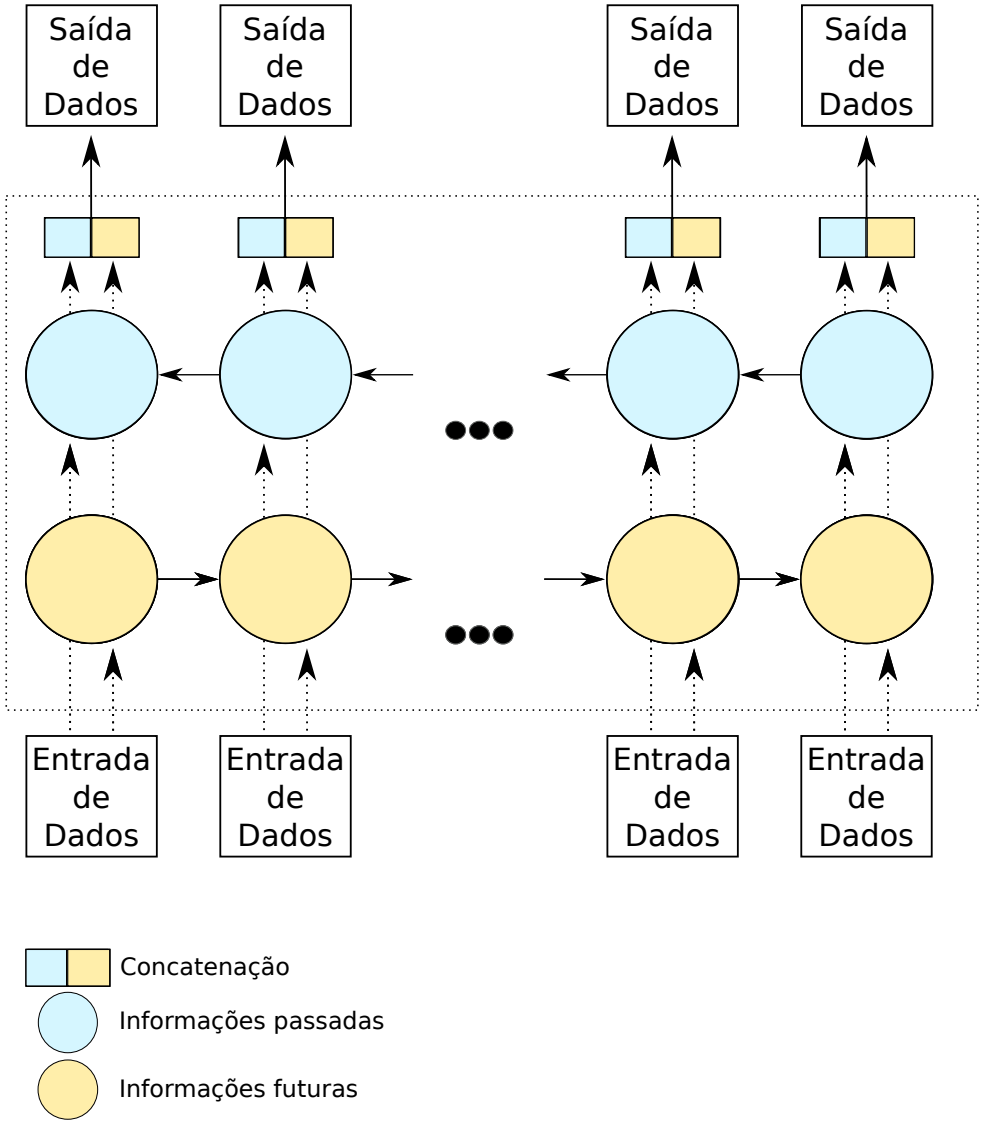



**Figura 3:** Redes Bidirecionais Recorrentes.

# Problemas com os Modelos Existentes

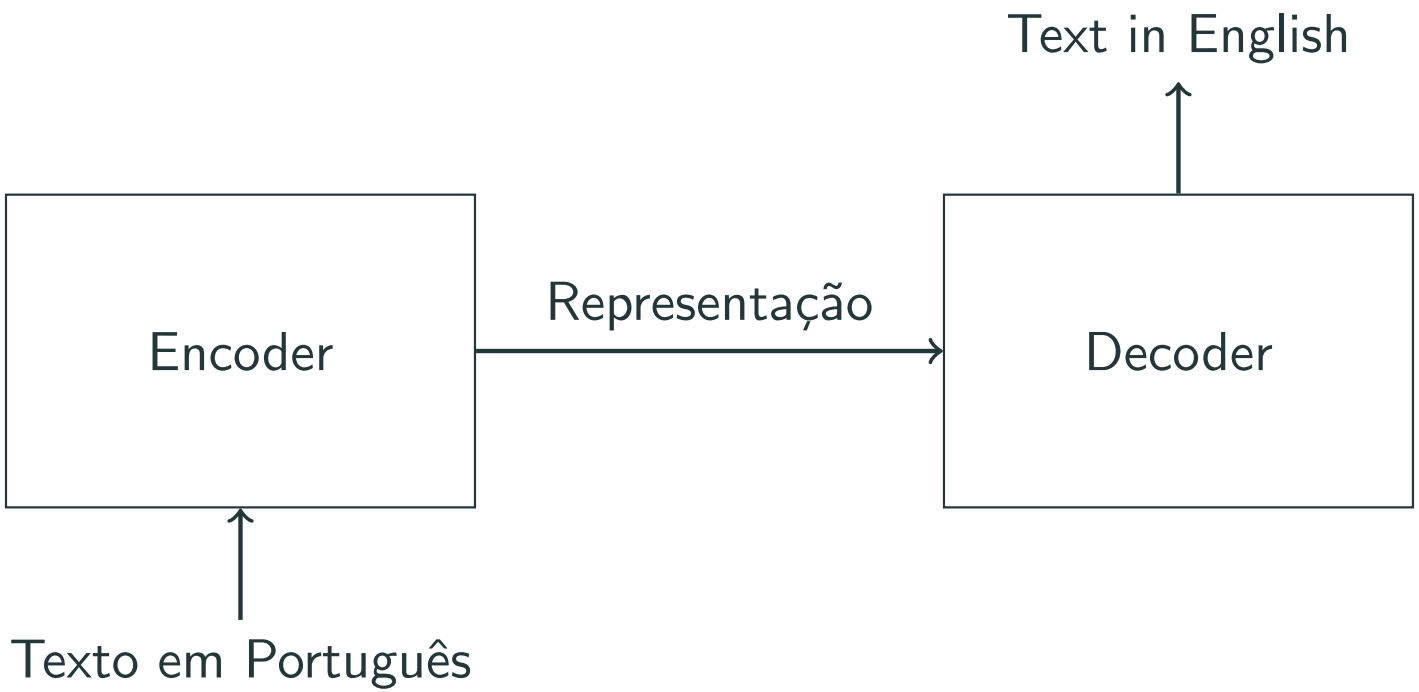
- Problemas com a abordagem existente, principalmente com relação de termos distantes.
- Como resolver esse problema?
  - Atenção e Transformers!

# Attention is All You Need

- Transformer foi apresentado no artigo *Attention is All You Need* por Vaswani e coautores (2017).
  - Artigo com mais de 140.000 citações.
- Modelo inteiramente baseado em mecanismos de atenção do tipo auto-atenção (*self-attention*).
- Arquitetura base para modelos estado da arte em Processamento de Linguagem Natural, como BERT, GPT e T5.

# Arquitetura Transformer Simplificada

- A arquitetura Transformer original é formada por um bloco de codificação (*encoder*) e um bloco de decodificação (*decoder*).
- O bloco de *encoder* é responsável por receber o texto de entrada e gerar uma representação.
- O bloco de *decoder* é responsável por receber a representação e gerar um texto de saída.



# Tipos de Modelos Transformers

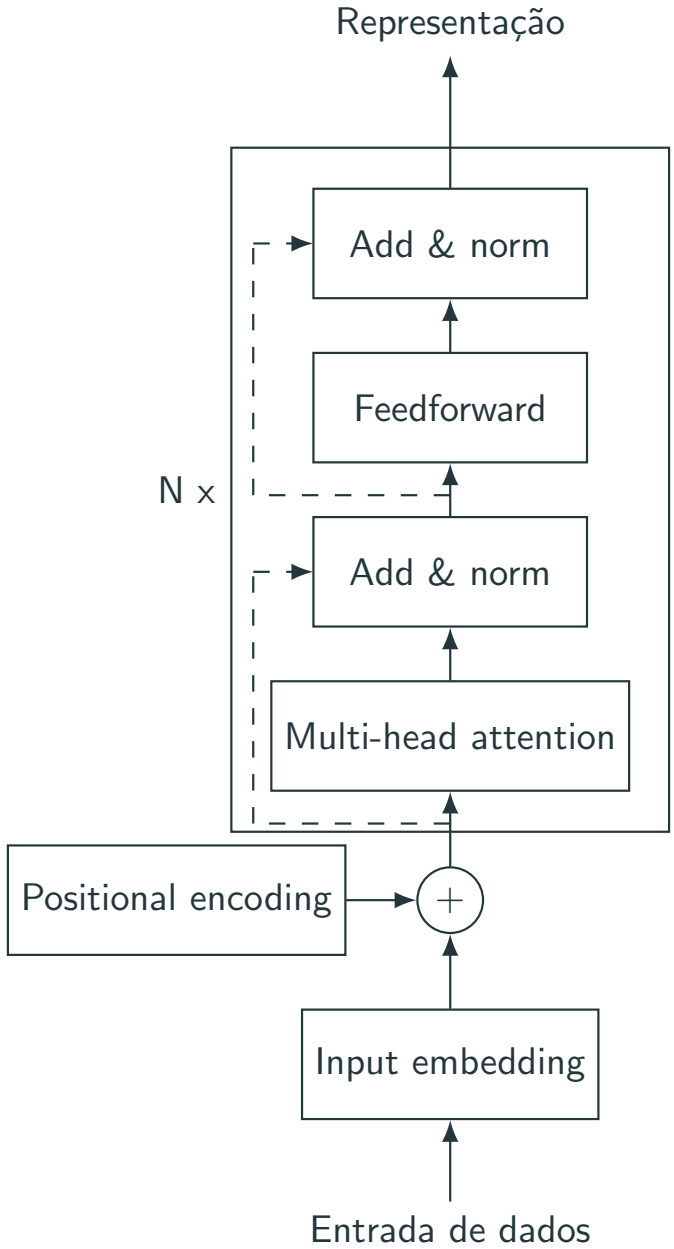
Os modelos baseados em Transformers podem ser divididos em três tipos:

- *Encoder-Decoder*:
  - Responsável por transformar o texto de entrada no texto de saída.
  - Usado em tarefas de tradução, sumarização, e pergunta e resposta.
- *Encoder*:
  - Responsável por entender o contexto do texto de entrada.
  - Usado em tarefas de classificação de texto e reconhecimento de entidades.
- *Decoder*:
  - Responsável por gerar texto.
  - Usado em tarefas de geração de texto, como completar frases e *chatbots*.

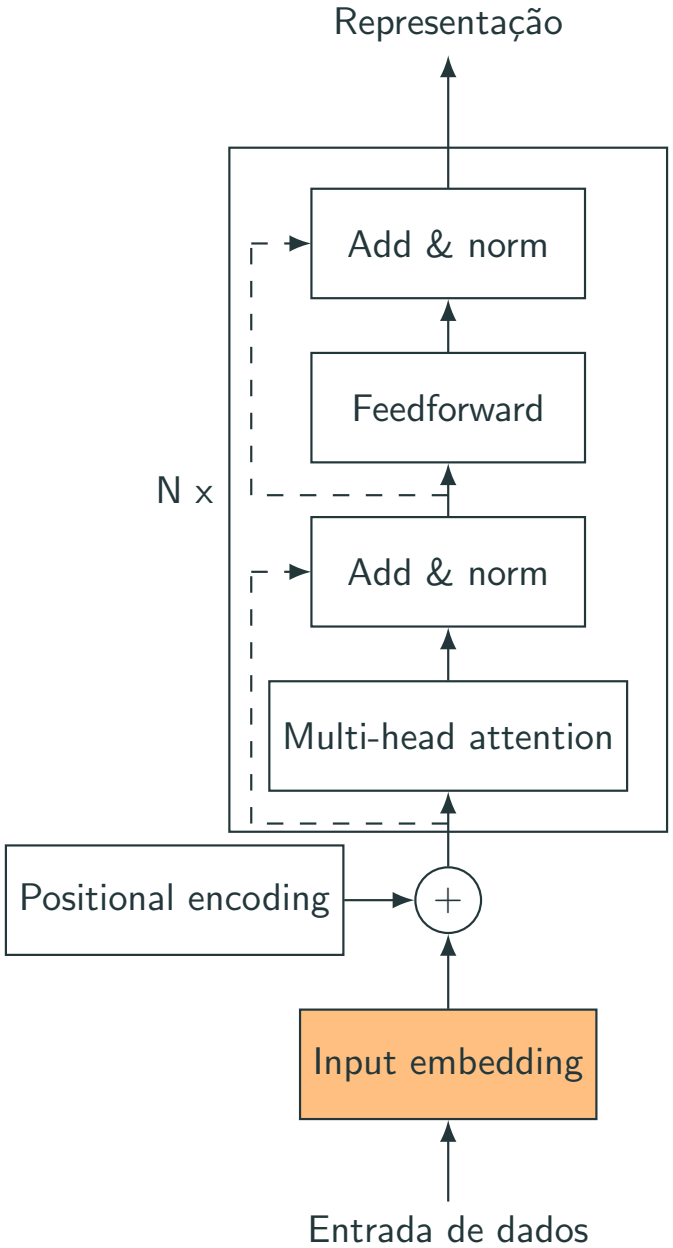
# Tipos de Modelos Transformers

- Qual tipo de modelo pode ser usado para classificação de estruturas e funções de proteínas?
- Queremos obter representações dada a sequência de aminoácidos da proteína, logo arquiteturas *encoder* devem ser usadas para essas tarefas.

# Arquitetura Encoder



# Arquitetura Encoder



# Tokenização

- Modelos computacionais não conseguem lidar com dados textuais diretamente. Portanto, precisamos transformar texto em números.
- Aplicação de tokenizadores:
  - Dicionários que mapeiam palavras ou conjunto de palavras (*tokens*) do vocabulário para representações numéricas inteiras.
- Exemplo:
  - Frase original: ``Os Transformers são incríveis''
  - Tokens: [``Os'', ``Transformers'', ``são'', ``incríveis'']
  - Conversão: [111, 7632, 98, 2205]

# Tokenização

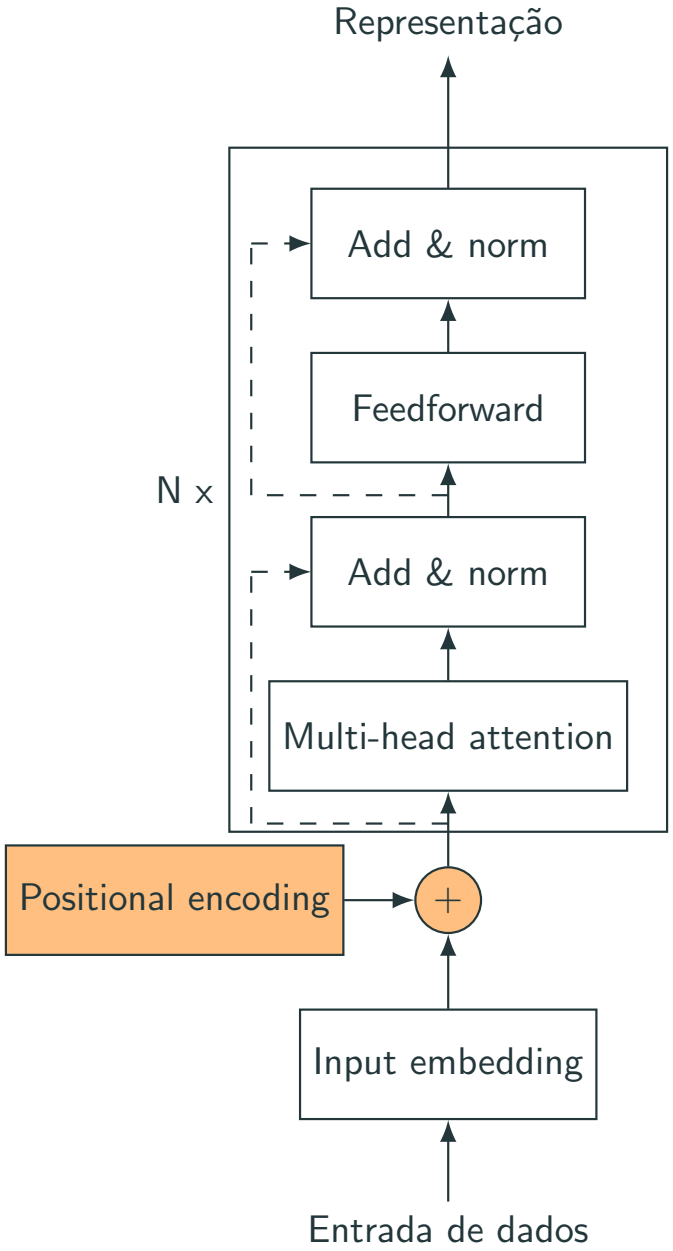
Os modelos podem utilizar *tokens* especiais para indicar condições específicas:

- `[CLS]`, `<CLS>`, `<bos>` e variações para início de texto.
- `<eos>`, `<end>` e variações para final de texto.
- `[PAD]`, `<pad>` e variações para preenchimento.
- `[MASK]`, `<mask>` e variações para mascaramento.

# Input Embedding

- Após a aplicação de tokenizadores, cada *token* é representado por um único valor inteiro.
- Na etapa de *input embedding*, cada valor inteiro único é transformado em um vetor de alta dimensão de valores reais.
- Isso permite que o modelo trabalhe em um espaço real contínuo:
  - Facilita a generalização.
  - Captura a relação semântica entre palavras.
  - Permite o entendimento melhor do contexto.

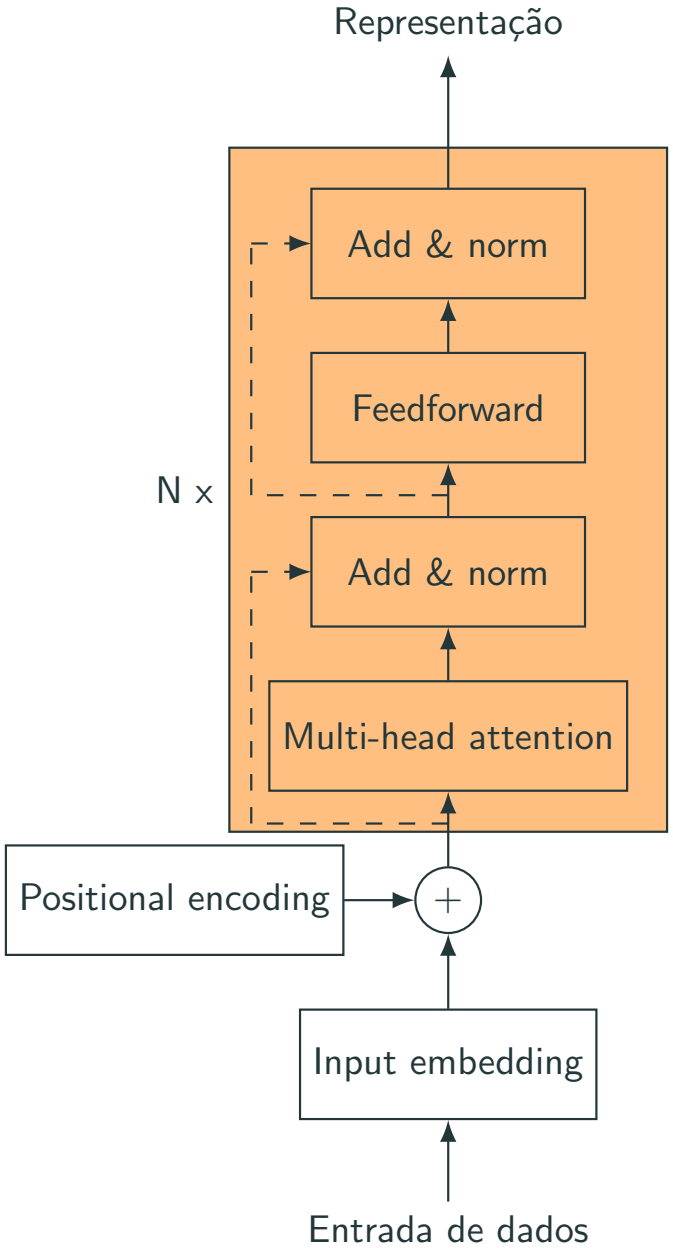
# Arquitetura Encoder



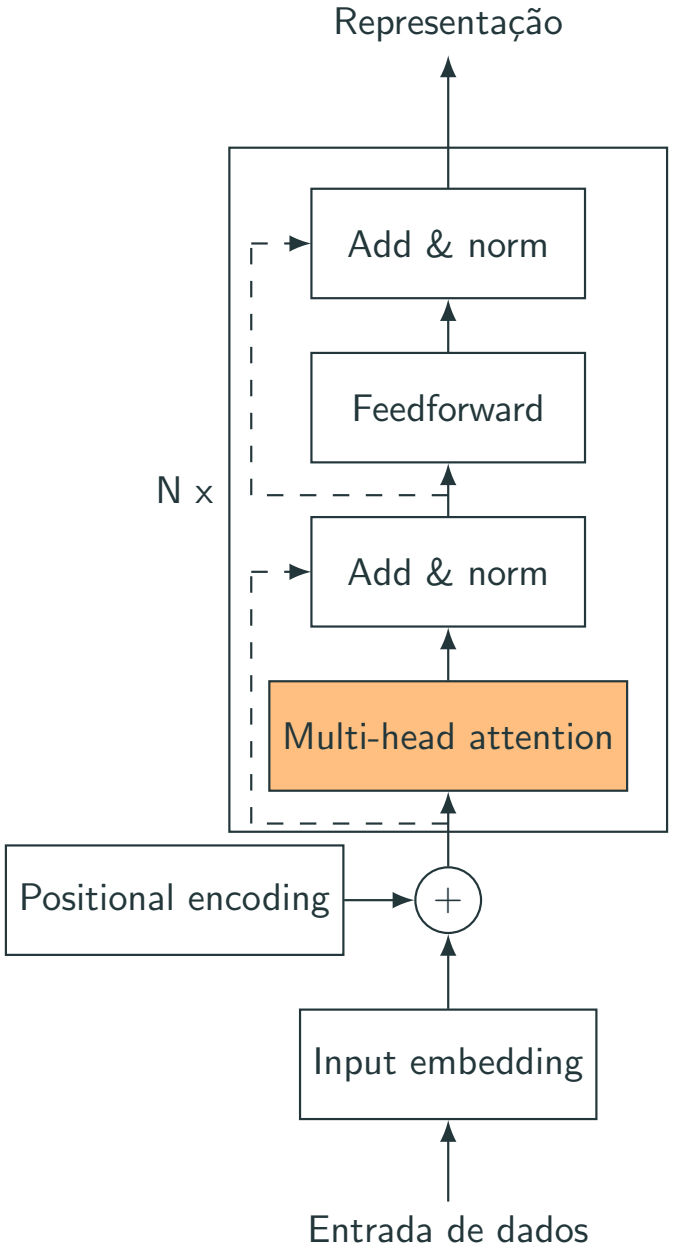
# Positional Encoding

- Como entender a ordem das palavras?
- *Positional encoding* é um codificador de posição para indicar a localização de cada palavra.
- A codificação por posição é somada com o vetor de representação de cada *token*.
- Para obter o valor de codificação por posição, Vaswani e coautores (2017) utilizaram a função seno considerando a posição do *token* no dado de entrada.

# Arquitetura Encoder



# Arquitetura Encoder



# Mecanismo de Auto-Atenção

`O menino jogou o disco para o cachorro e ele foi buscar`

O pronome `ele` se refere a quem?

- Menino?
- Disco?
- Cachorro?

O mecanismo de auto-atenção auxilia o modelo a entender a relação entre as palavras.

# Mecanismo de Auto-Atenção

Como isso é feito?

- Dado o vetor que representa cada *token*, três novos vetores são gerados:
  - Consulta.
  - Chave.
  - Valor.
- Os pesos para gerar os vetores são aprendidos durante o processo de treinamento, aplicados ao vetor de entrada da camada.

# Mecanismo de Auto-Atenção

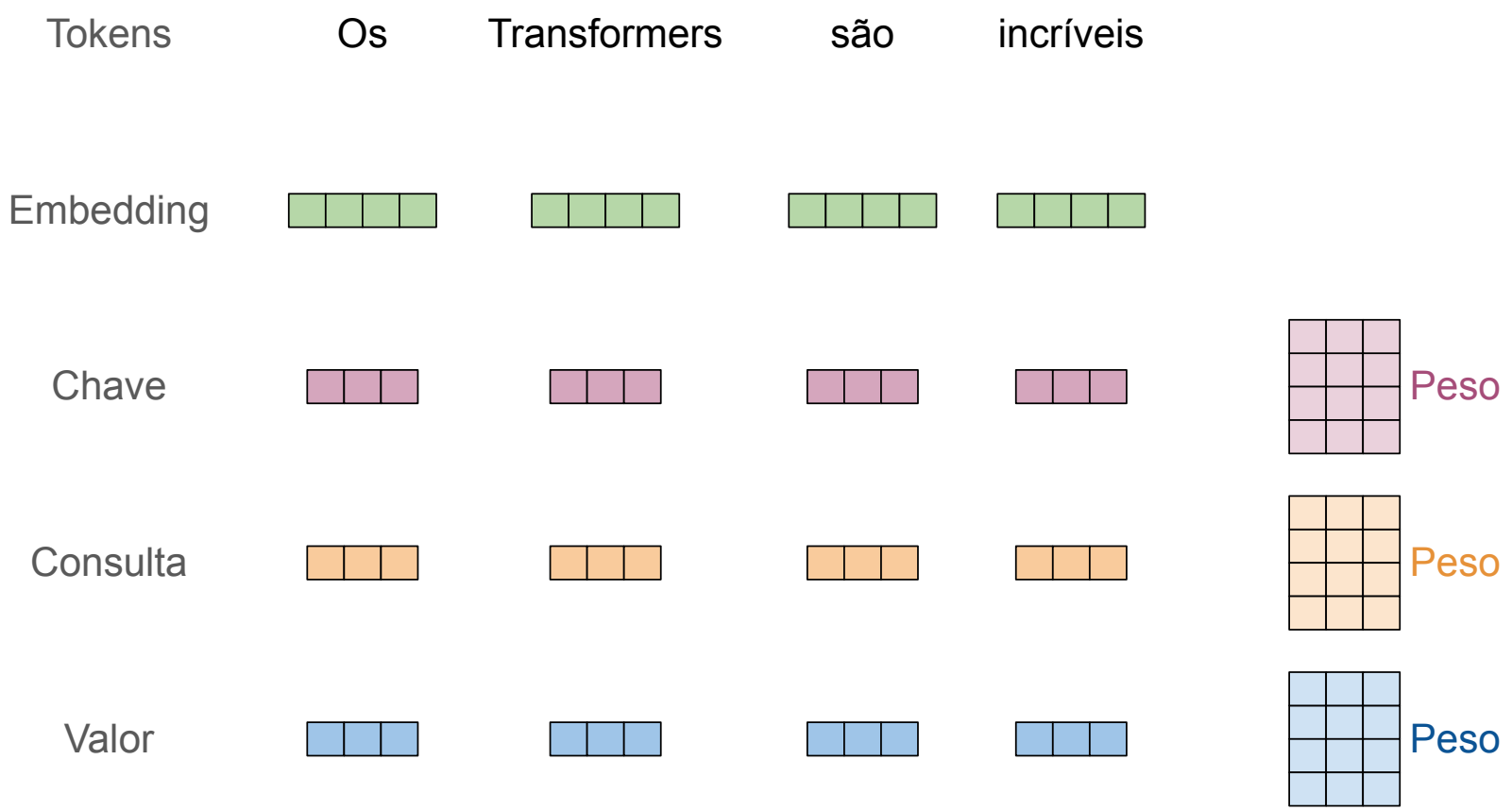



**Figura 4:** Geração de vetores de chave, consulta e valor para cada *token*.

# Mecanismo de Auto-Atenção

- A partir dos três vetores para cada *token*, calculamos a similaridade entre os pares de palavras.
- Para isso, para cada *token*, fazemos a multiplicação de matrizes entre a consulta deste *token* com a chave transposta de cada palavra da entrada de dados.

# Mecanismo de Auto-Atenção

Exemplo de calculo de similaridade com o *token* `Transformers`. O mesmo processo ocorre para os outros *tokens*.

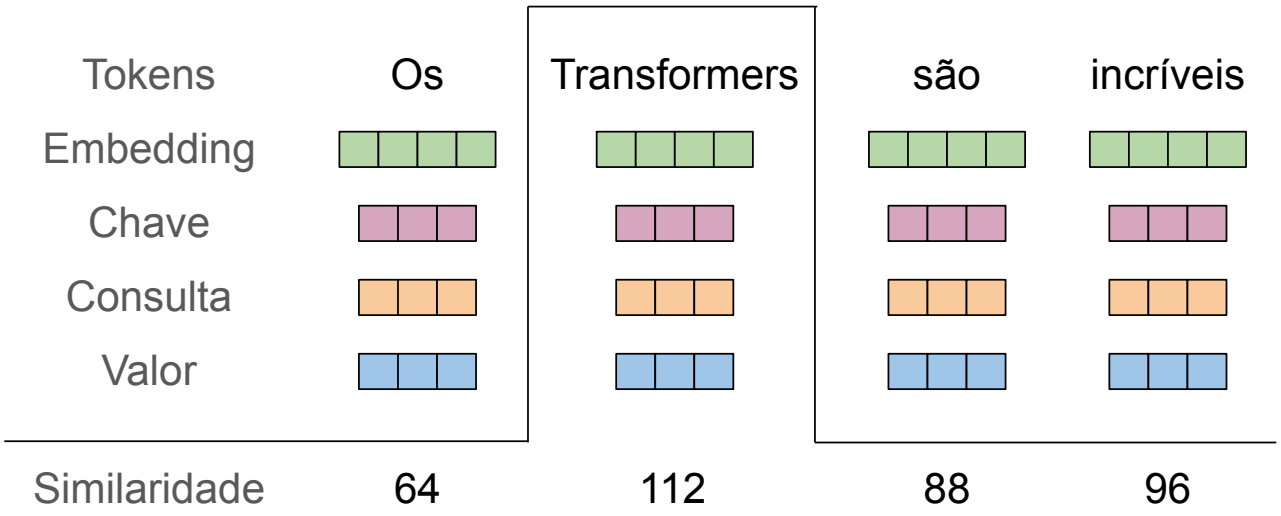



**Figura 5:** Cálculo da similaridade dos *tokens* com `Transformers`.

# Mecanismo de Auto-Atenção

Após o processo de similaridade entre cada par de *token*, a normalização é aplicada para gerar resultados mais estáveis

# Mecanismo de Auto-Atenção

Exemplo de normalização dada a similaridade dos *tokens* com o *token* `Transformers`. O mesmo processo ocorre para os outros *tokens*.

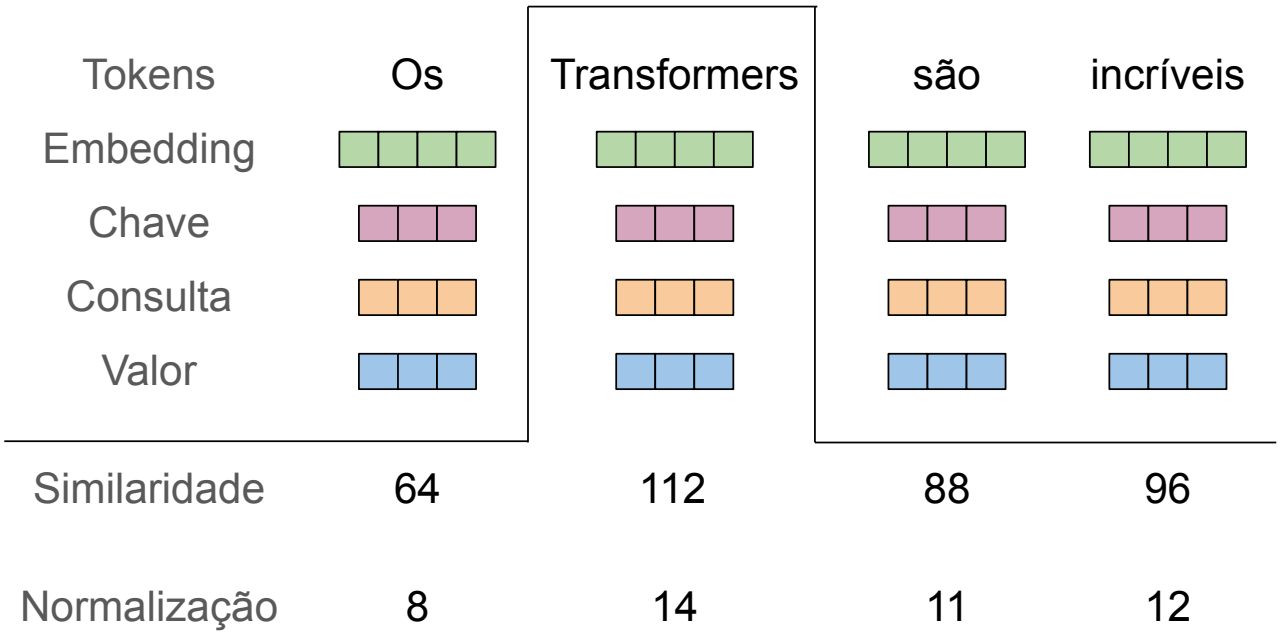



**Figura 6:** Normalização da similaridade dos *tokens* com `Transformers`.

# Mecanismo de Auto-Atenção

- Após a normalização, a função *softmax* é utilizada.
- Desta forma, a relação entre os *tokens* é transformada em porcentagem.

# Mecanismo de Auto-Atenção

Exemplo do *softmax* usando a similaridade entre cada um dos *tokens* com o *token* `Transformers`. O mesmo processo ocorre para os outros *tokens*.

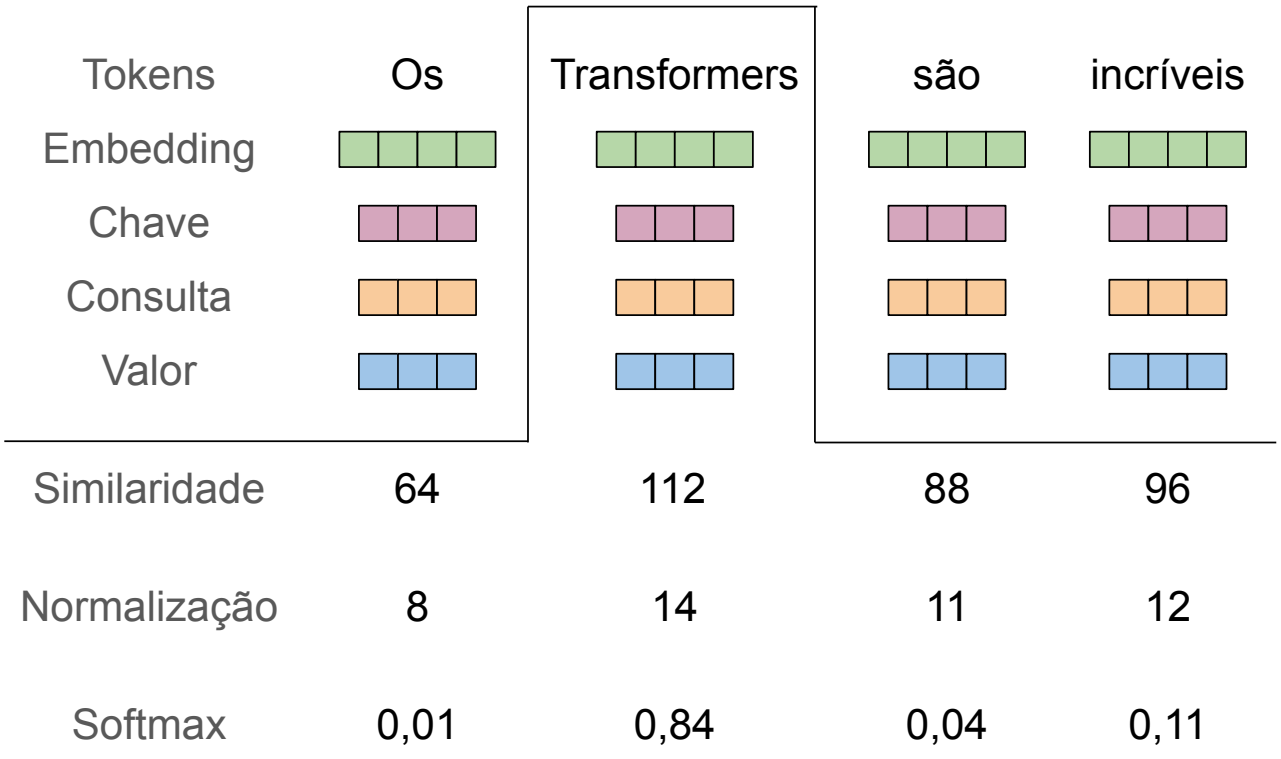



**Figura 7:** *Softmax* da similaridade dos *tokens* com `Transformers`.

# Mecanismo de Auto-Atenção

Por fim, o vetor de valor de cada *token* é ponderado pelo valor obtido no passo anterior.

# Mecanismo de Auto-Atenção

Exemplo da ponderação dos *tokens* com o *token* `Transformers`. O mesmo processo ocorre para os outros *tokens*.

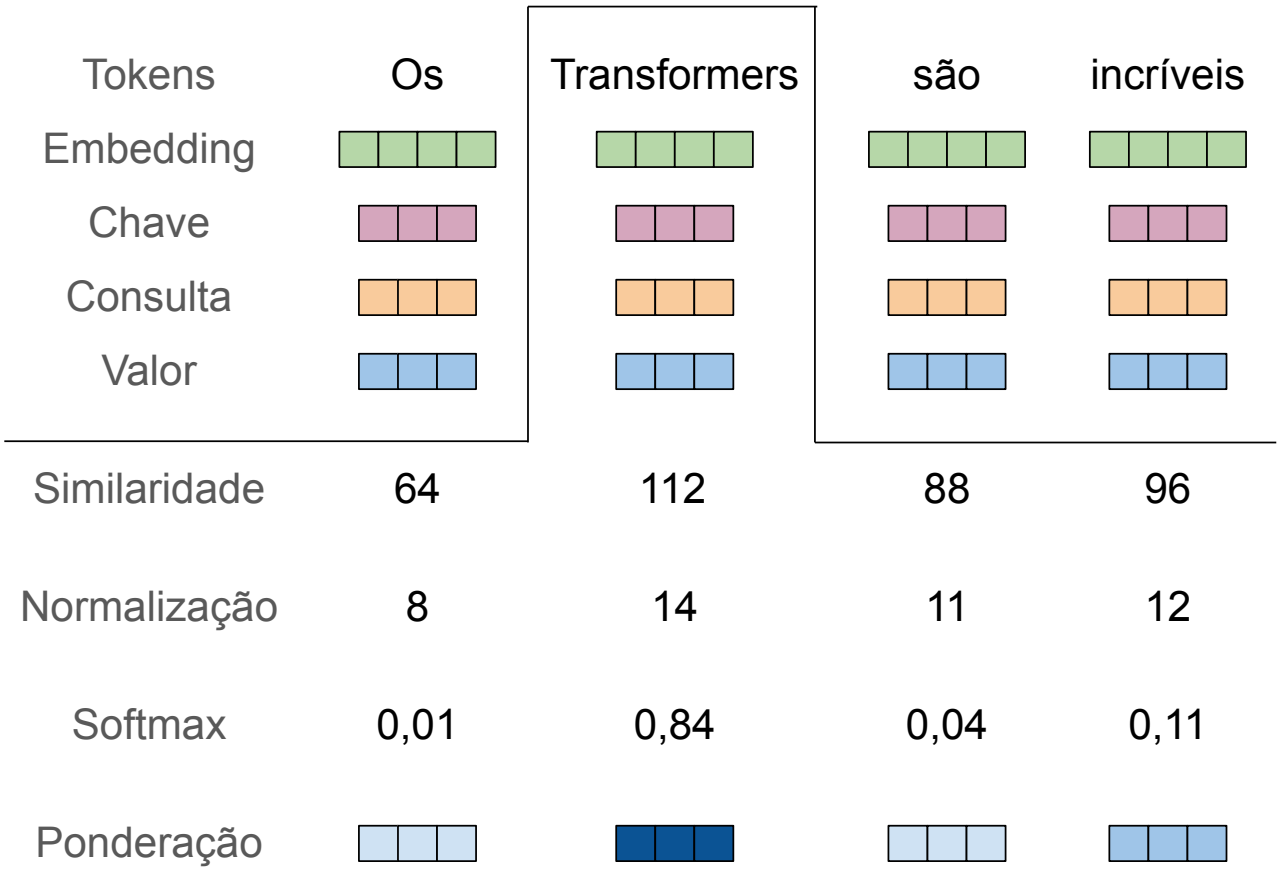



**Figura 8:** Ponderação da similaridade dos *tokens* com `Transformers`.

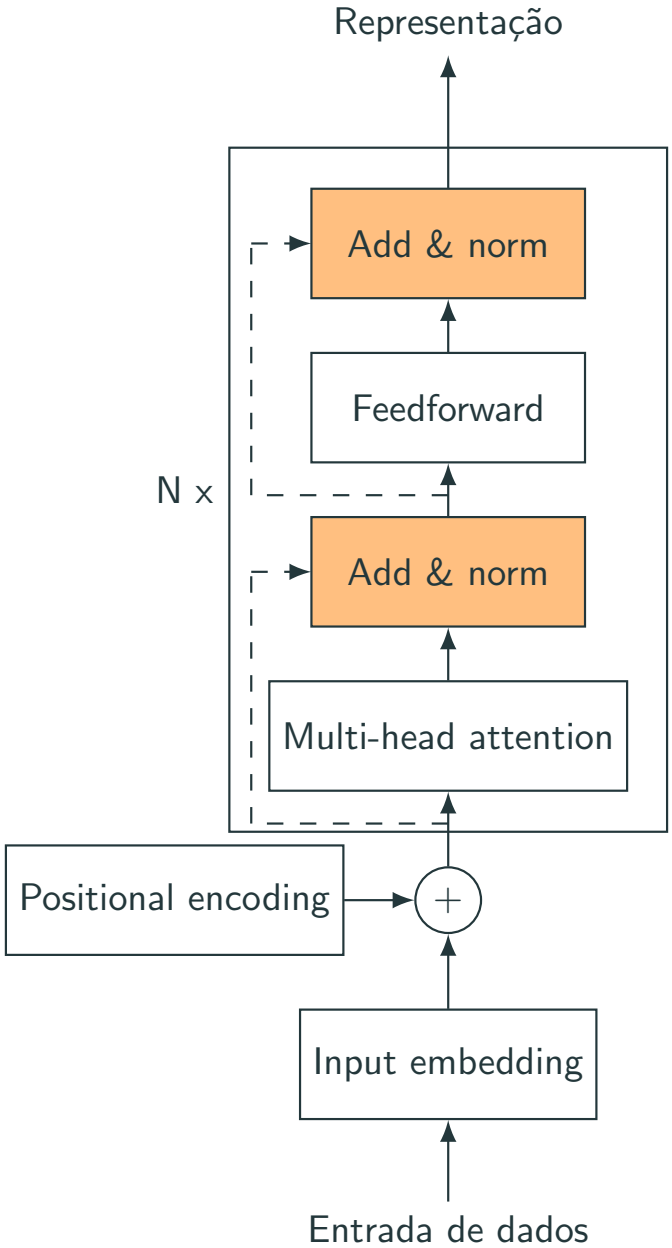
# Mecanismo de Auto-Atenção

- Ao final da ponderação, os vetores resultantes da análise de todos os *tokens* com o *token* analisado são somados.
- Dessa forma, teremos uma nova representação de um *token* específico.

# Multi-head Attention

- O processo descrito anteriormente gerará uma cabeça de atenção (*single-head attention*).
- Os modelos Transformers utilizam diversas cabeças de atenção (*multi-head attention*) ao mesmo tempo.
- Ao final, todas as cabeças de atenção são concatenadas e ponderadas por um peso aprendido durante o treinamento para gerar a representação final dessa camada.

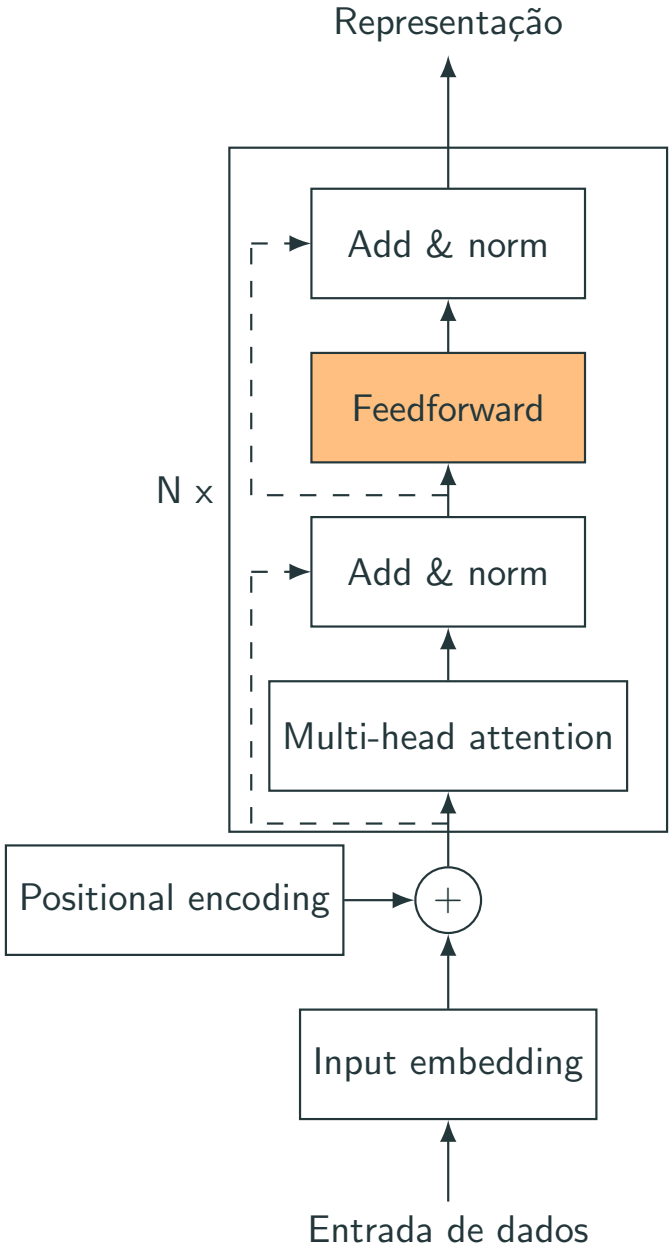
# Arquitetura Encoder



# Add & Norm

- Camada responsável por conectar a entrada e saída da camada anterior.
- Realiza a normalização:
  - Prevenção de valores discrepantes.
  - Facilitação do aprendizado.
  - Promove treinamento mais rápido.

# Arquitetura Encoder



# Feedforward

- Rede neural do tipo *multilayer perceptron* totalmente conectada.
- Possui duas camadas com ativação ReLU.
- Quantidade de neurônios por camada é exatamente igual ao tamanho da matriz de entrada na camada de *multi-head attention*.
- Sendo assim, cada *token* tem a mesma quantidade de representações no formato de números reais.

# Configuração dos Modelos Transformers

- Os modelos Transformers são conhecidos pelas suas configurações.
- Normalmente nomeados por quantidade de camadas *encoder*, número de *attention heads* ou número de *embeddings* por *token*.
- Exemplos:
  - ESM2 T6 - Possui 6 camadas de *encoder*.
  - ESM2 T12 - Possui 12 camadas de *encoder*.
  - ESM2 T36 - Possui 36 camadas de *encoder*.

# Pré-treinamento

- Inicialmente, os modelos Transformers passam por um pré-treinamento auto-supervisionado.
- Etapa de entendimento da linguagem.
- Utilização de milhões ou bilhões de documentos.
- Custos de aprendizado alto:
  - Múltiplas GPUs em paralelo (algumas vezes centenas ou milhares).
  - Muitas horas de treinamento.
- Normalmente com aprendizado por mascaramento.

# Aprendizado por Mascaramento

Frase Original: Os Transformers são incríveis

Frase com Máscara: Os [MASK] são incríveis

- O que `[MASK]` pode ser?
  - 18% de chance de ser "Aviões".
  - 20% de chance de ser "Carros".
  - 62% de chance de ser "Transformers". ✓

# Ajuste Fino

- Após a etapa de pré-treinamento, o modelo já conhece o comportamento da linguagem.
- Na sequência, podemos ajustar (levemente) o conhecimento para a tarefa investigada.
- Processo mais simples e que requer menos custo computacional.

# Modelos Transformers para Proteínas

Principais modelos *encoder* para proteínas:

- ProtBERT (2021).
- ProtElectra (2021).
- ESM1 (2021).
- ProteinBERT (2022).
- ESM2 (2022).
- Ankh (2024).

# Modelos Transformers para Proteínas

Outros modelos para proteínas:

- ProtT5 (*encoder-decoder*, 2021).
- ProtXLNet (*encoder-decoder*, 2021).
- ProtGPT2 (*decoder*, 2022).
- ProGen2 (*decoder*, 2022).

# Alinhamento Local

# Alinhamento Local

- Alinhamento entre subsequências que possui a maior similaridade entre si.
- Avaliação de alinhamento com pontuações:
  - Normalmente é utilizada a matriz BLOSUM62.
- Aplicação de ferramentas que utilizam heurísticas:
  - BLASTp.
  - DIAMOND.

# BLASTp

- Versão 2.0 do BLAST que consulta proteínas em um banco de dados de proteínas.
- Bitscore:
  - Pontuação normalizada para o alinhamento.
  - Quanto maior, melhor.
- E-Value:
  - Limiar para a probabilidade de encontrar alinhamentos semelhantes ao acaso.
  - Leva em conta o alinhamento avaliado e o tamanho da base.
  - Quanto menor, mais raro.

# DIAMOND

- Ferramenta proposta por Buchfink, Reuter e Drost (2021).
- Alternativa mais eficiente (em tempo) do que o BLASTp.
- Tende a ser entre 100 e 10.000 vezes mais rápido que o BLASTp.

# DIAMOND

Como o DIAMOND consegue ser mais rápido que o BLASTp?

- Indexação dupla:
  - Tanto a base de dados quanto a consulta sofre o processo de indexação.
- Uso otimizado de memória e cache.
- Filtros baseados em heurísticas mais avançadas focando em regiões promissoras.

# Classificação de Estruturas Secundárias

# Estruturas Secundárias

- Estruturas secundárias são estruturas tridimensionais que cada um dos aminoácidos forma.
- Ocorrem devido às interações físicas e químicas entre os aminoácidos da proteína.
- As estruturas formadas são energeticamente eficientes.

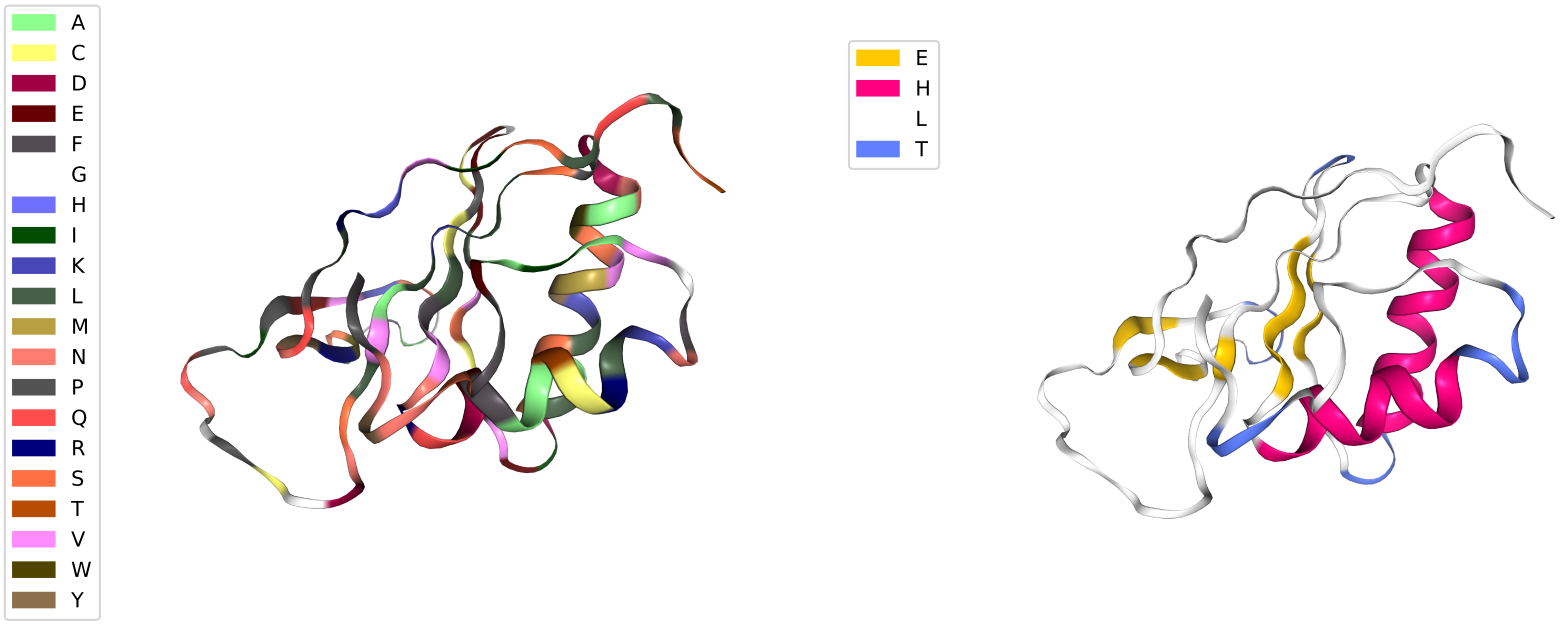
# Classificação de Estruturas Secundárias - Q3

- A classificação mais simples de estruturas secundárias é chamada de Q3.
- Cada aminoácido pode fazer parte de uma de três possíveis classes:
  - Hélice ("H").
  - Folha ("E").
  - Espiral ("C").

# Classificação de Estruturas Secundárias - Q8

- A classificação mais detalhada de estruturas secundárias é chamada de Q8.
- Nessa classificação, as esturuturas são uma subclassificação da Q3.
- Cada aminoácido pode fazer parte de uma das oito possíveis classes:
  - Hélice alfa ("H").
  - 3-hélice ("G").
  - Resíduo em folha beta isolada ("B").
  - Folha estendida ("E").
  - 5-hélice ("I").
  - Torção de ligação de hidrogênio ("T").
  - Torção ("S").
  - Espiral ("L").

# Classificação de Estruturas Secundárias - Q8




**(a)** Aminoácidos na forma tridimensional.

**(b)** Estruturas secundárias.

**Figura 9:** Sequência de aminoácidos e estruturas secundárias da proteína PDB ID: 6BI6.

# Mapeamento Q8 - Q3

- O mapeamento Q8 para Q3 não é padronizado.
- Dificuldade por não possuir uma clara fronteira biológica entre as estruturas.

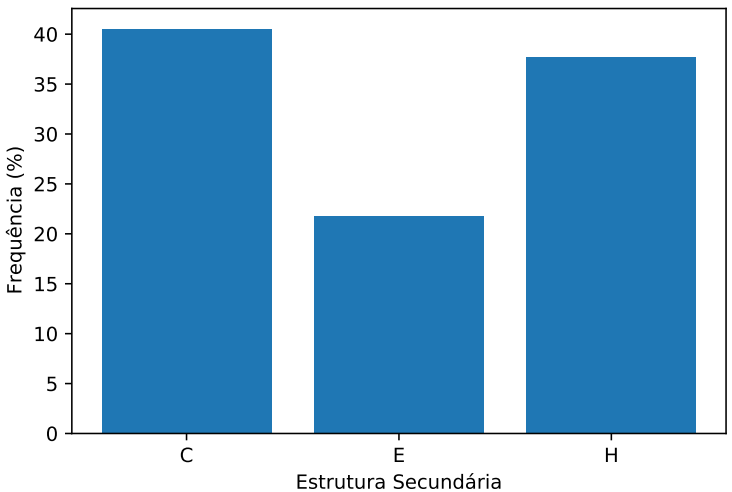
| | Classes Q8 | | | | |
|---|---|---|---|---|---|
| **Classes Q3** | **Método 1** | **Método 2** | **Método 3** | **Método 4** | **Método 5** |
| C | I, L, S, T | B, G, I, L, S, T | L, S, T | B, I, L, S, T | B, L, S, T |
| E | B, E | E | B, E | E | E |
| H | G, H | H | G, H, I | G, H | G, H, I |

**Tabela 2:** Métodos para agrupamento de classes.

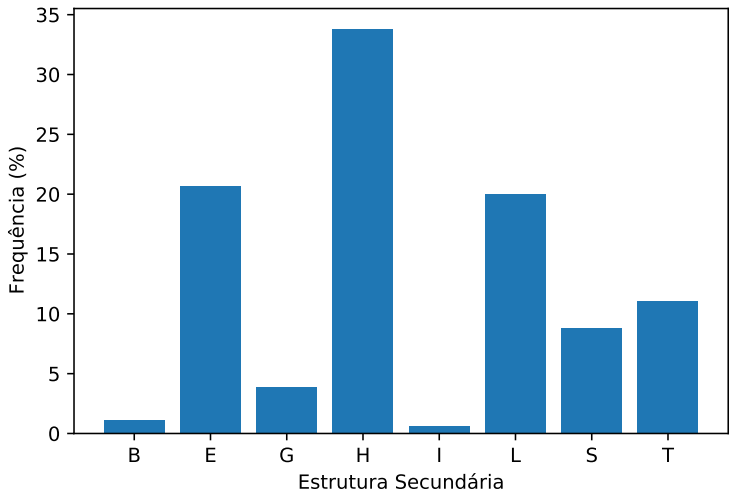
- O método 1 é o mais aplicado na literatura.

# Frequência das Classes

- A frequência das classes é desbalanceada.
- Para Q3, as classes espiral (“C”) e hélice (“H”) são mais frequentes.
- Para Q8, as classes hélice alfa (“H”), folha estendida (“E”) e espiral (“L”) são as mais frequentes.




**(a)** Frequência de cada classe Q3.




**(b)** Frequência de cada classe Q8.

**Figura 10:** Distribuição das classes na base PDB 2018.

# Bases de Dados

- As bases mais utilizadas na literatura são:
  - CB6133: base com 6.133 proteínas com até 700 aminoácidos.
  - CB513: base com 513 proteínas com até 700 aminoácidos.
  - CASP: desafio bianual de classificação de estruturas.
- Base proposta na dissertação do mestrado:
  - PDB 2018: base com 7.979 proteínas com até 700 aminoácidos.

# Avaliação

- Como avaliar as predições dos modelos?

$$\text{Acurácia Q3} = \frac{\sum_{i \in \{\text{C, E, H}\}} \text{predições corretas em } i}{\sum_{i \in \{\text{C, E, H}\}} \text{dados da classe } i} \tag{1}$$

$$\text{Acurácia Q8} = \frac{\sum_{i \in \{\text{B, E, G, H, I, L, S, T}\}} \text{predições corretas em } i}{\sum_{i \in \{\text{B, E, G, H, I, L, S, T}\}} \text{dados da classe } i} \tag{2}$$

# História dos Métodos

Os trabalhos de classificação de estruturas secundárias usando métodos computacionais se dividem em três fases.

# Primeira Fase

- Abordagens desenvolvidas até 1990.
- Primeiro trabalho sobre predição de estruturas secundárias: Chou e Fasman (1974).
- Métodos de regras quantitativas e qualitativas (definidas empiricamente) e medidas estatísticas.
- Baixo poder computacional e bases de dados pequenas.

# Segunda Fase

- Abordagens desenvolvidas entre 1990 a 2010.
- Aumento do poder computacional e criação de grandes bases de proteínas.
- Primeiros trabalhos usando redes neurais: Holley e Karplus (1989), Kneller *et al.* (1990) e Jones (1999).
- Limitação de algoritmos de aprendizado de máquina com utilização de janelas deslizantes.
- Desenvolvimento do BLAST nesse período.
- Aumento de Acurácia Q3 e utilização da Acurácia Q8.

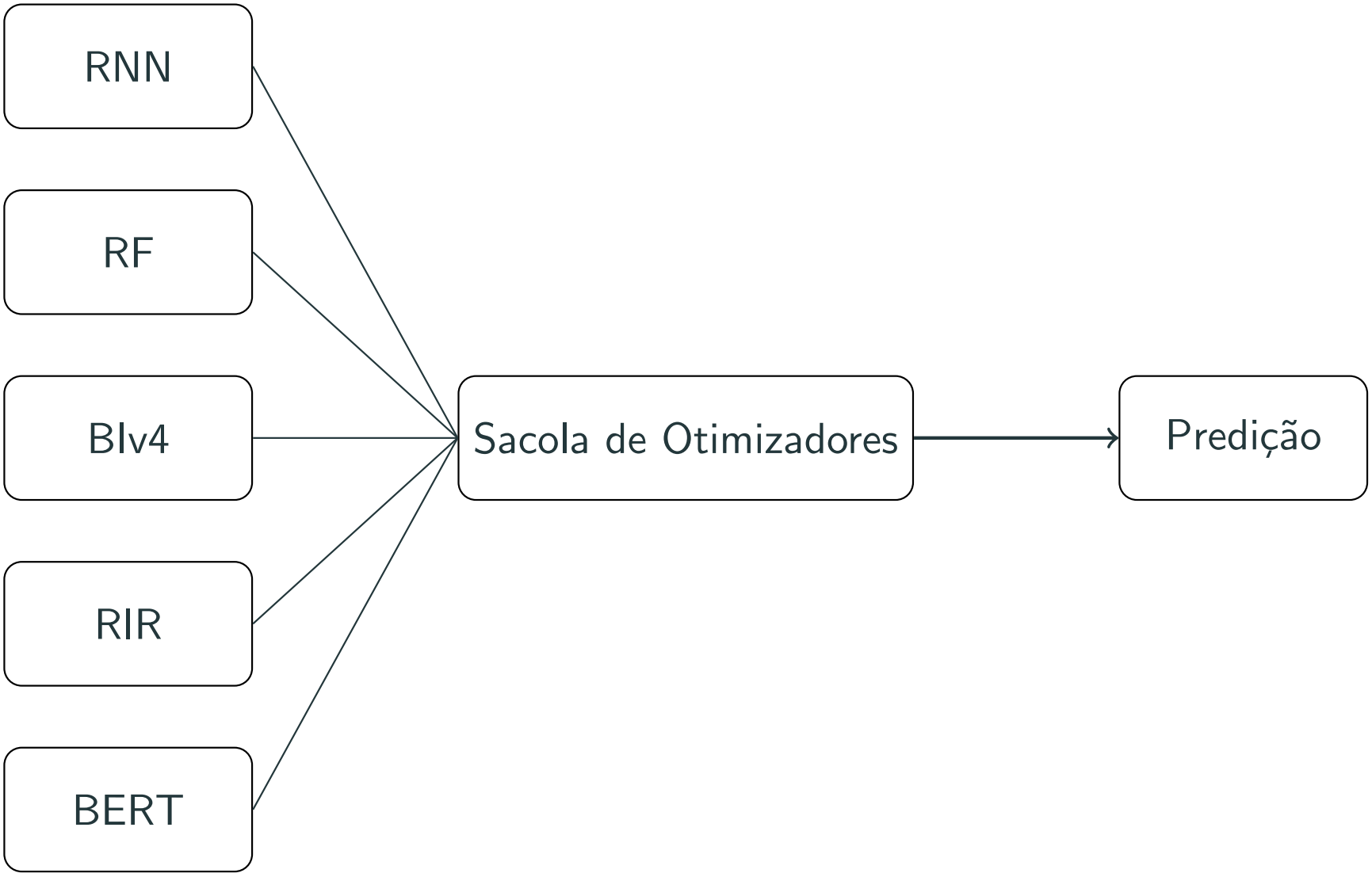
# Terceira Fase

- Abordagens desenvolvidas após 2010.
- Uso de redes neurais:
  - Convolução.
  - Recorrência.
  - Transformers.
- Fusão de classificações feitas com aprendizado de máquina e alinhamento local.

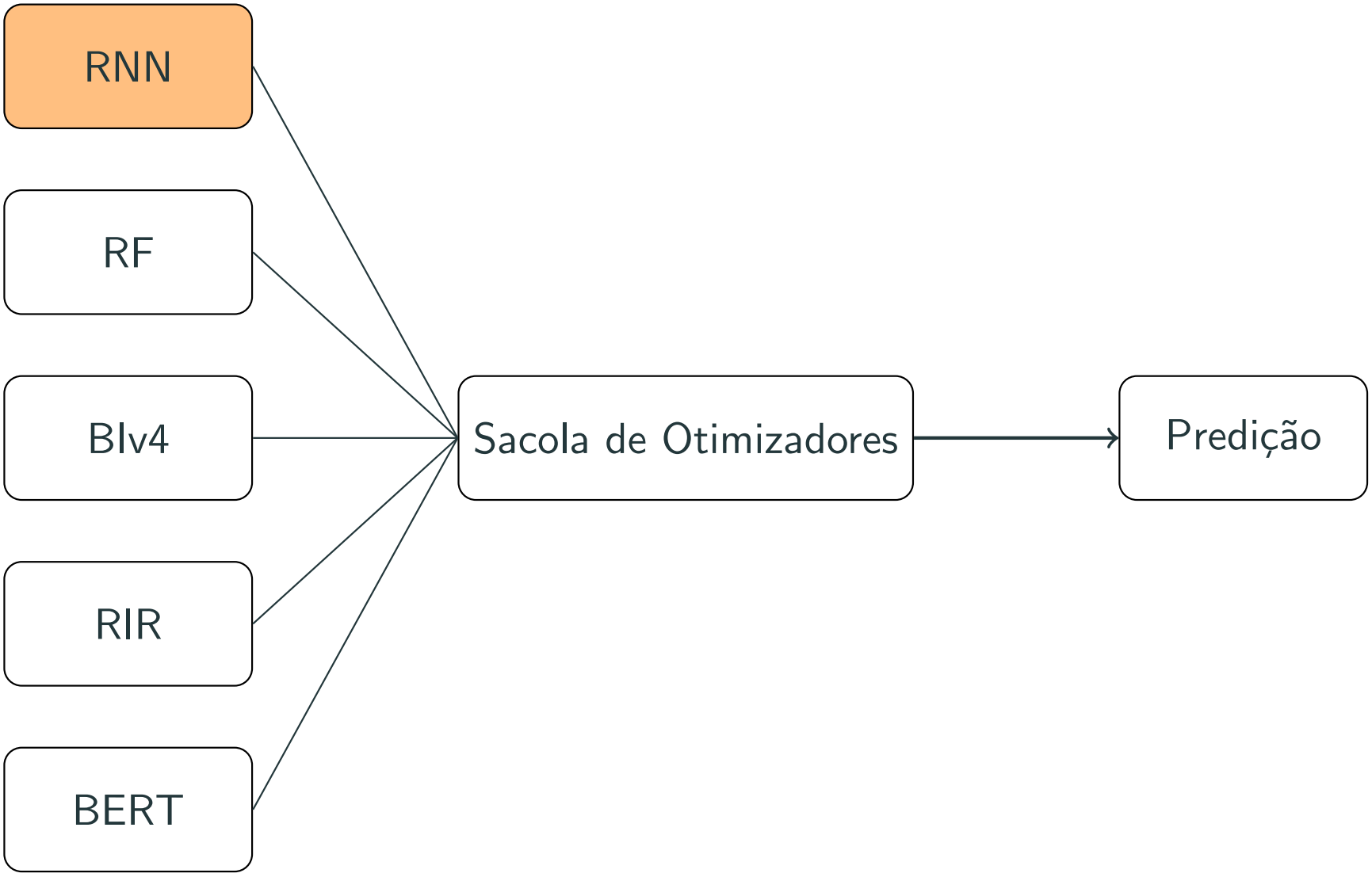
# Método Proposto no Mestrado

- Duas abordagens:
  - Métodos livres de modelo (aprendizado de máquina).
  - Métodos baseados em modelo (alinhamento local).
- Agregação das predições das duas abordagens.
- O texto da dissertação pode ser acessado pelo link: `https://bit.ly/masterGBO`.

# Métodos Livres de Modelo



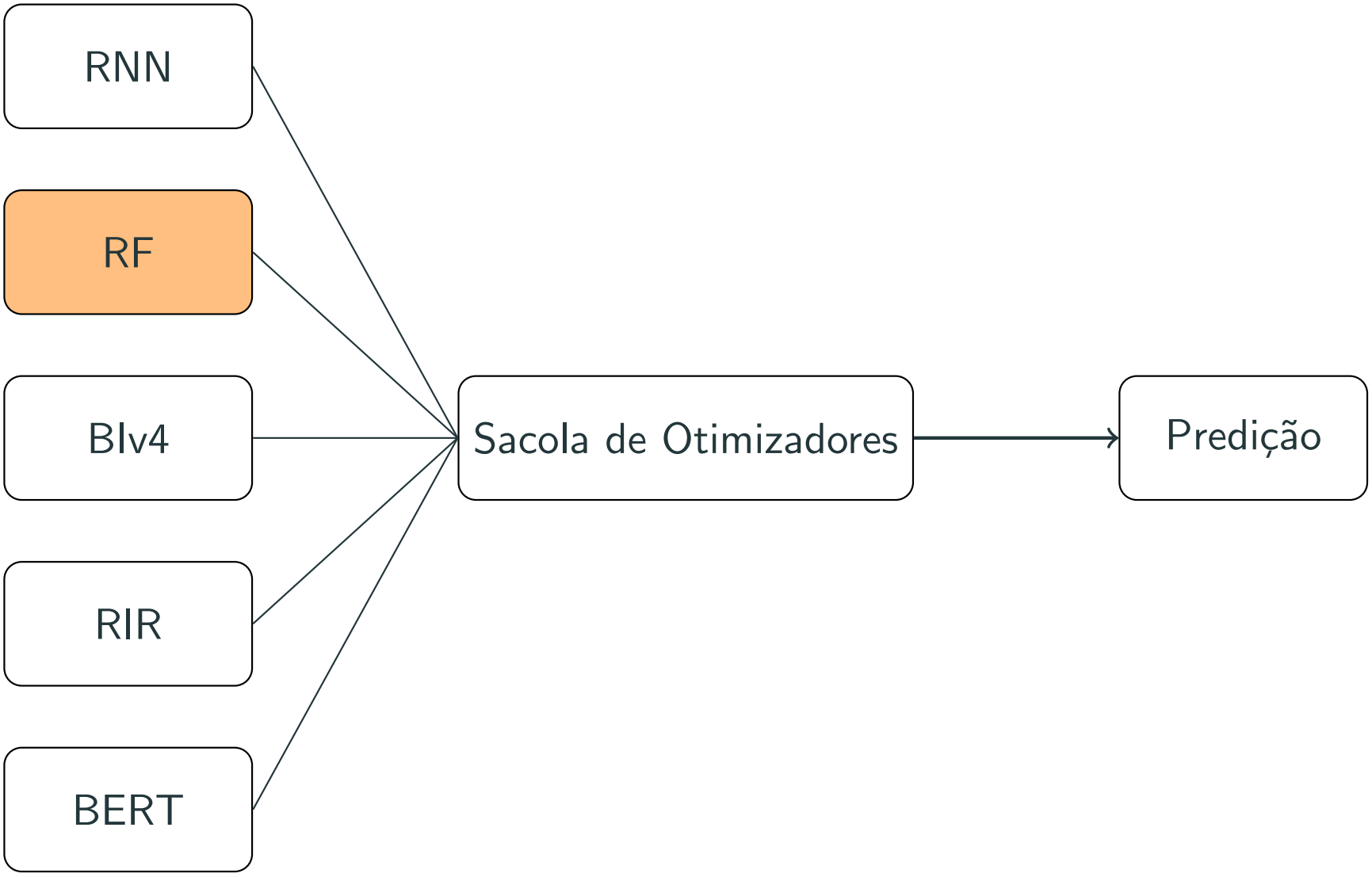
# Métodos Livres de Modelo



# Rede Neural Recorrente

- Redes neurais bidirecionais recorrentes.
- Módulos GRU de memória.
- Embedding para a sequência de aminoácidos.
- Diversas configurações, variando de 2 a 6 camadas bidirecionais recorrentes.
- Agregação das predições dos diversos modelos.

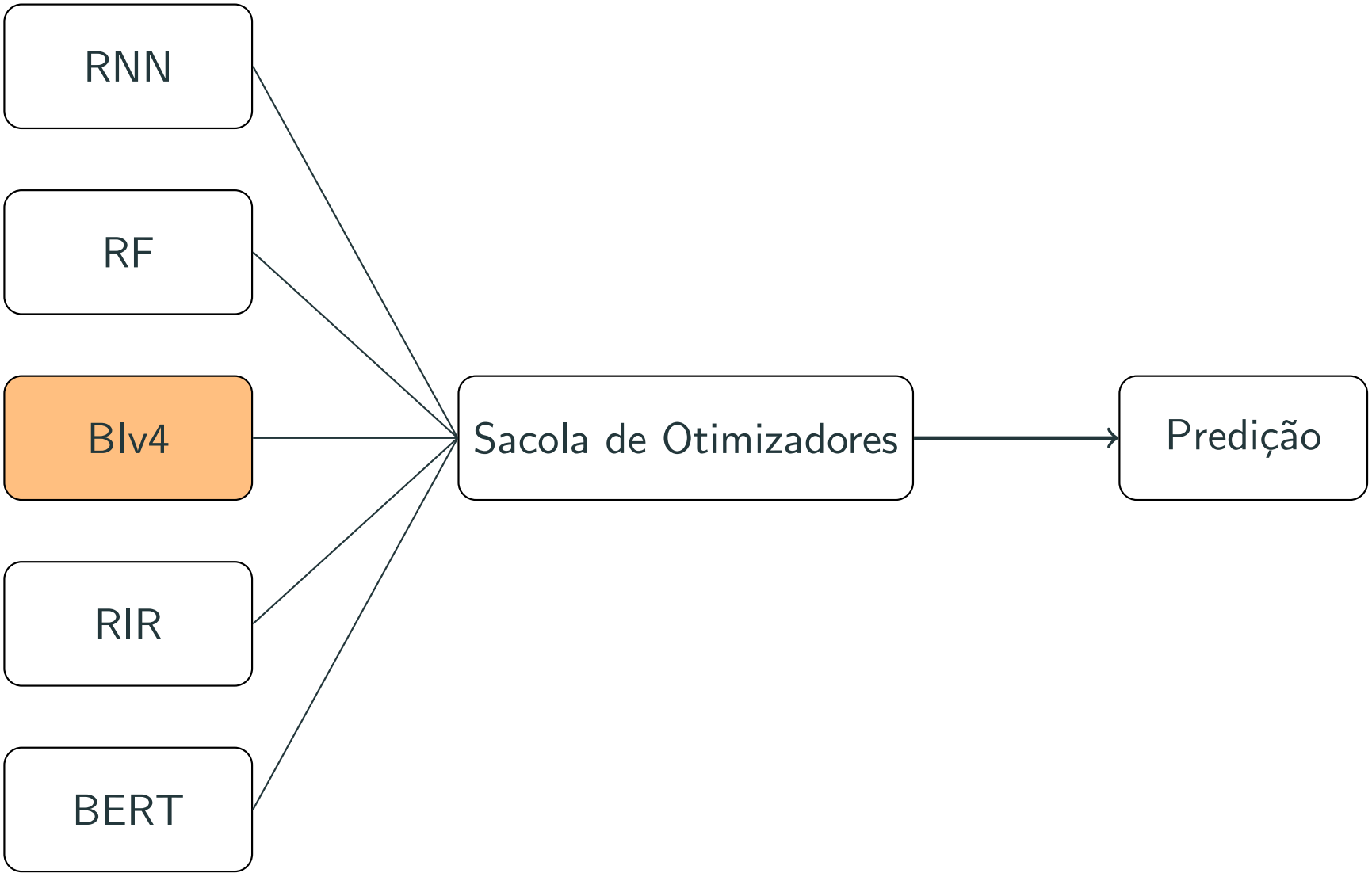
# Métodos Livres de Modelo



# Florestas Aleatórias

- Florestas Aleatórias (*Random Forest*) analisando janelas de aminoácidos.
- Janelas de 9 a 17 aminoácidos, classificando o aminoácido central.
- Agregação das predições por cada janela.

# Métodos Livres de Modelo



# Blocos InceptionV4

- Rede baseada na arquitetura InceptionV4, alterando os filtros para uma dimensão.
- Redes com 3 a 7 blocos empilhados.
- Agregação das predições por cada rede.

# Blocos InceptionV4

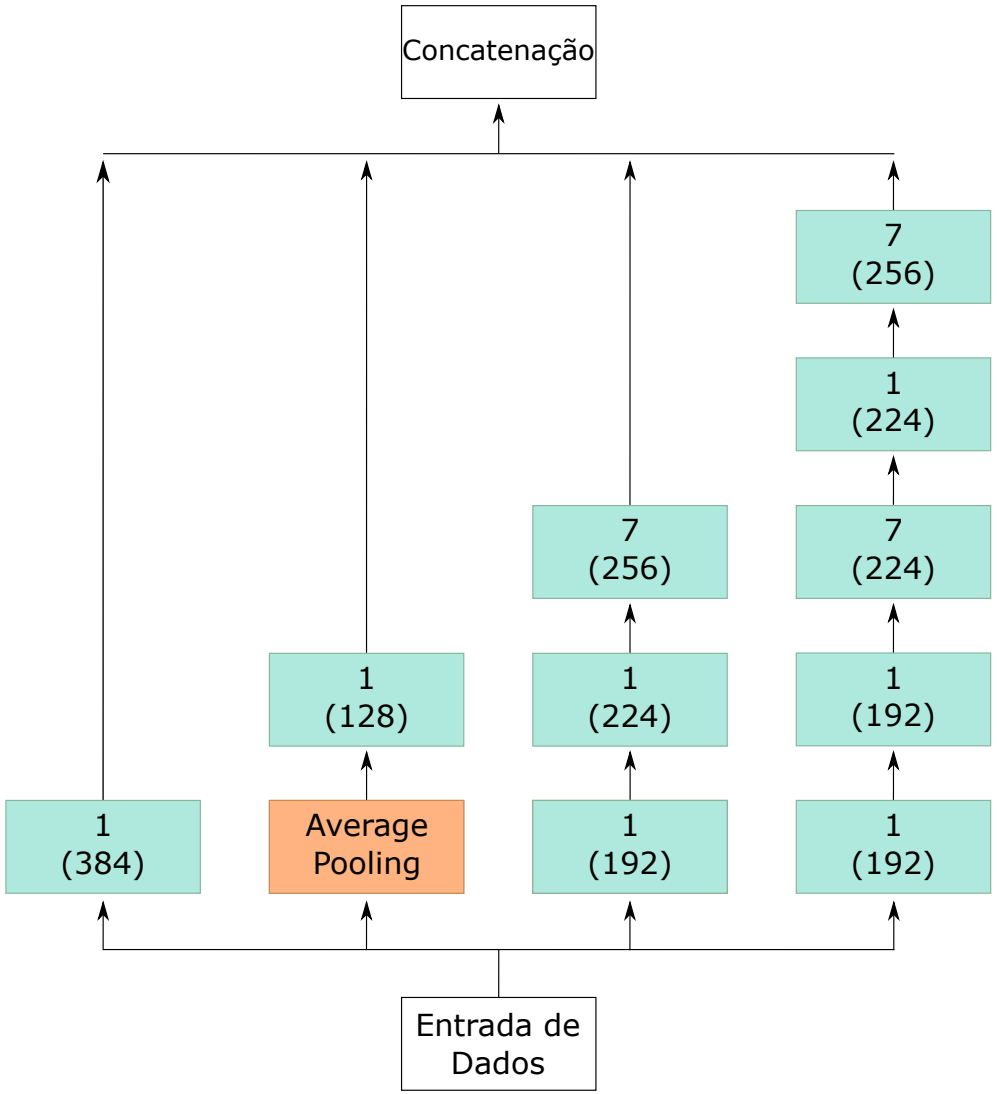
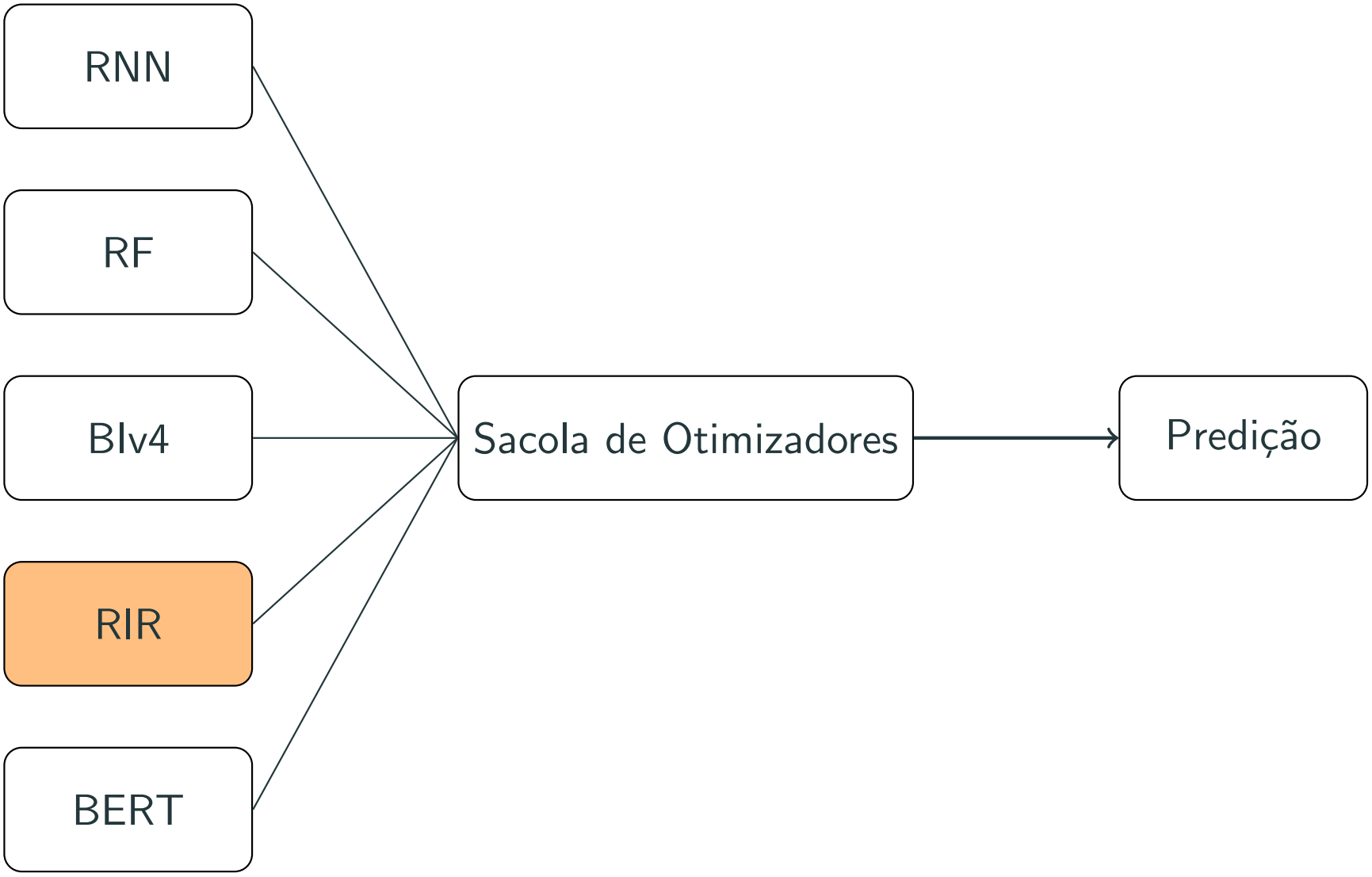



**Figura 11:** Bloco InceptionV4.

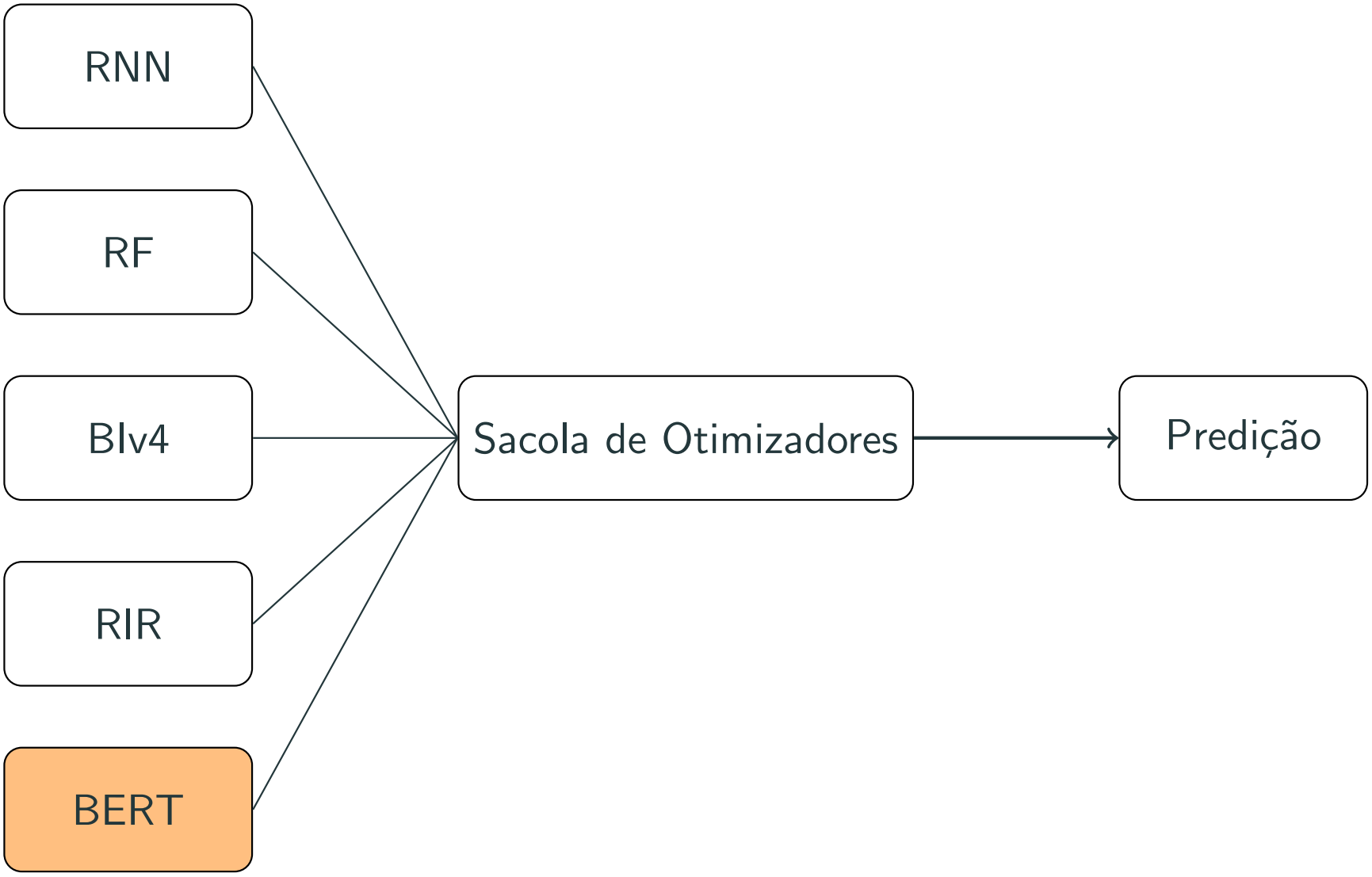
# Métodos Livres de Modelo



# Redes Inception Recorrentes

- Junção das redes InceptionV4 com as recorrentes.
- Após as convoluções, a informação passa por camadas de recorrência.
- Agregação das predições com variações de blocos InceptionV4 (3 a 7 blocos).

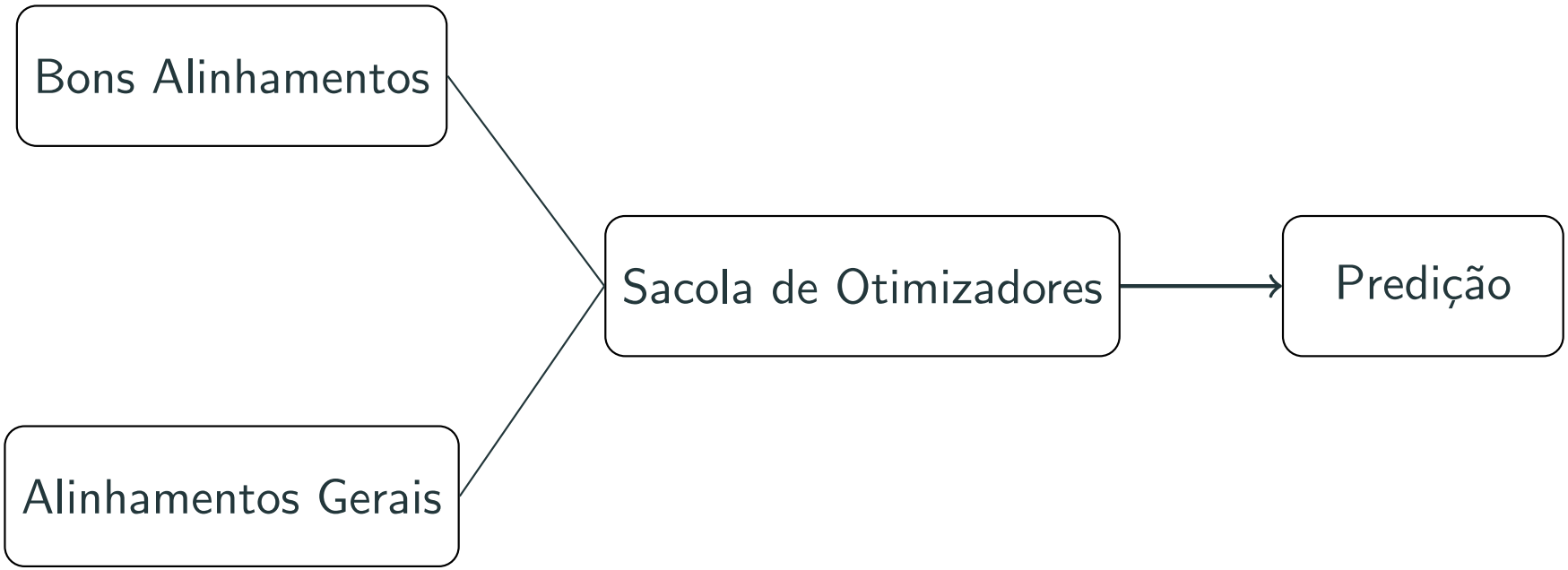
# Métodos Livres de Modelo



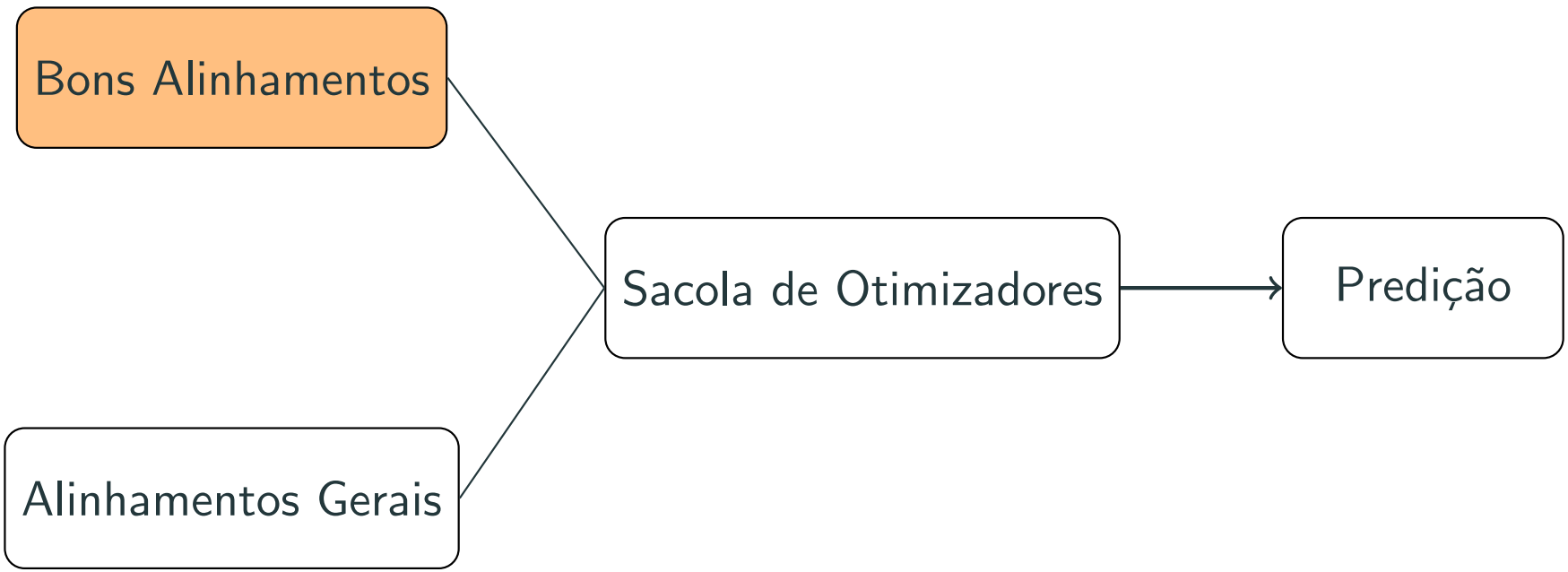
# BERT

- Utilização do BERT pré-treinado em inglês.
- Análise de janelas deslizantes para classificar o aminoácido central.

# Métodos Baseados em Modelo



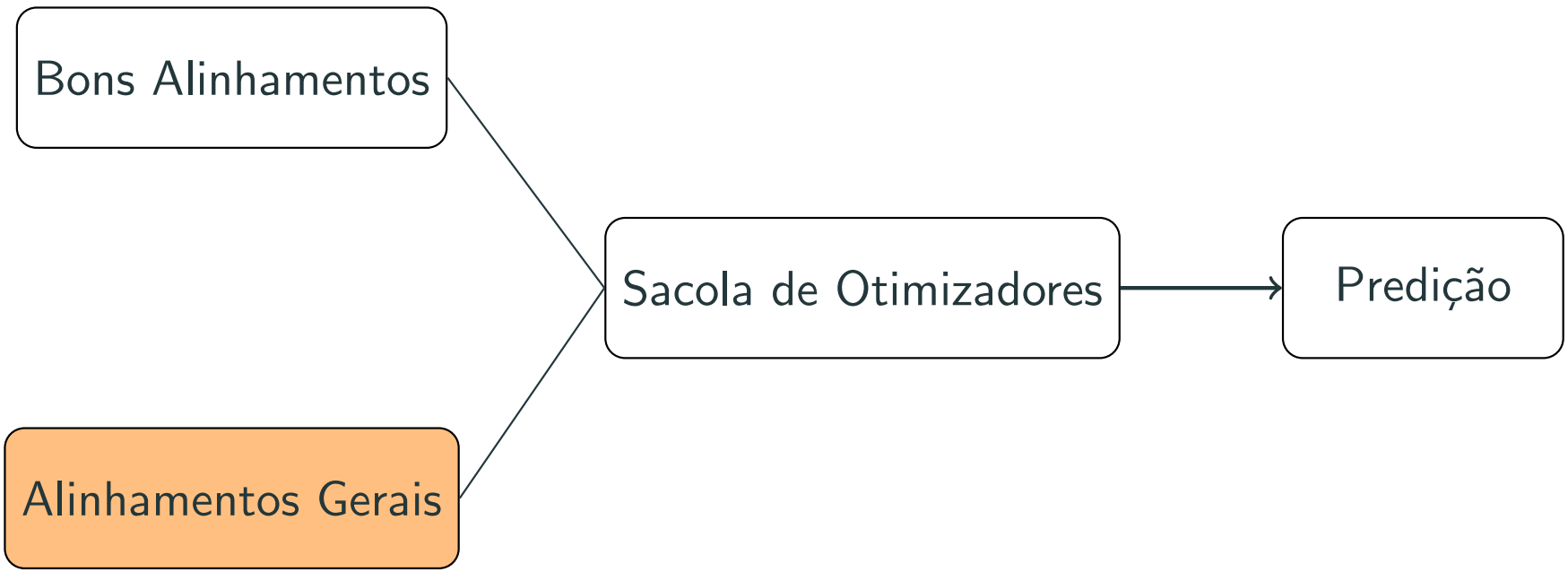
# Métodos Baseados em Modelo



# Bons Alinhamentos

- Aplicação do BLASTp.
- Seleção dos 10 melhores alinhamento com no máximo $10^{-5}$ de E-Value.
- Peso decrescente para cada alinhamento, de 10 a 1.

# Métodos Baseados em Modelo



# Alinhamentos Gerais

- Aplicação do BLASTp.
- Seleção dos alinhamento com no máximo 10 de E-Value.
- Peso decrescente para cada alinhamento.

# Agregação entre as Predições

- Desenvolvemos a Sacola de Otimizadores:
  - Algoritmo Genético.
  - Busca Cuco.
  - Otimização por Enxame de Partículas.
- Cada otimizador irá buscar os pesos para a predição de cada classe para cada classificador.
- O melhor resultado encontrado é utilizado.

# Algoritmo Genético

Utiliza duas buscas:

- Busca global.
- Buscal local.

# Busca Global

**Algoritmo 1:** Etapas da Busca Global

Gerar população inicial de 2.000 indivíduos com pesos entre 0 e 10;
Definir máximo de 100 iterações;
**para** *cada geração* **faça**
    **para** *cada indivíduo* **faça**
        Calcular função de custo (acurácia Q3 ou Q8);
    **fim**
    Selecionar os 100 melhores indivíduos como pais;
    Gerar 900 novos indivíduos por cruzamento;
    Formar nova geração com 1.000 indivíduos (pais + filhos);
    Gerar 1.000 novos indivíduos por mutação;
    Normalizar pesos dos indivíduos;
**fim**

# Busca Local

**Algoritmo 2:** Etapas da Busca Local

Selecionar os 100 melhores indivíduos da busca global;

Definir máximo de 100 iterações;

**para** *cada geração* **faça**

    Gerar 900 novos indivíduos por mutação;

    Normalizar pesos dos indivíduos;

**fim**

Selecionar o melhor indivíduo ao final;

# Busca Cuco

**Algoritmo 3:** Etapas da Busca Cuco

Iniciar população de 1.000 indivíduos com pesos entre 0 e 10;
Ordenar indivíduos pela função de custo (acurácia Q3 ou Q8);
Definir máximo de 100 iterações;
**para** *cada geração* **faça**
    Calcular o ovo de cada indivíduo usando voos de Lévy;
    Avaliar ovo pela função de custo;
    **para** *cada ovo* **faça**
        Selecionar indivíduo aleatório para comparação;
        Substituir o ovo do indivíduo aleatório com a cópia do melhor ovo;
    **fim**
    Ordenar indivíduos por acurácia decrescente (Q3 ou Q8);
    Reinicializar os 25% piores indivíduos com novos pesos aleatórios;
**fim**
Selecionar o melhor indivíduo ao final;

# Otimização por Enxame de Partículas

**Algoritmo 4:** Etapas da Otimização por Enxame de Partículas

```
Iniciar população de 1.000 indivíduos com pesos entre 0 e 10 e
 velocidade igual a 0;
Definir máximo de 100 iterações;
para cada geração faça
    para cada indivíduo faça
        Avaliar posição pela função de custo (acurácia Q3 ou Q8);
        se posição atual for melhor que a melhor posição anterior
         então
            Atualizar melhor posição individual;
        fim
        se posição atual for melhor que a melhor posição global então
            Atualizar melhor posição global;
        fim
    fim
    para cada indivíduo faça
        Atualizar velocidade;
        Atualizar posição;
    fim
fim
Selecionar o melhor indivíduo ao final;
```

# Fusão Final

- Avaliamos duas maneiras de fundir as predições:
  - Fusão Hierárquica: fusão da predição final dos métodos livres de modelo com a predição final dos métodos baseados em modelo.
  - Fusão Horizontal: fusão entre os sete classificadores (cinco vindo dos métodos livres de modelo e dois vindo dos métodos baseados em modelo).

# Resultados

| Método | CB6133 Q8 | CB513 Q8 | PDB 2018 Q8 | PDB 2018 Q3 |
|---|---|---|---|---|
| Fusão Horizontal | **82,83** | 88,79 | **89,66** | 93,72 |
| Fusão Hierárquica | 82,61 | 88,69 | 89,60 | **93,74** |
| Métodos Baseados em Modelo | 78,73 | **89,16** | 88,55 | 93,21 |
| Métodos Livres de Modelo | 62,69 | 57,81 | 74,76 | 83,03 |

**Tabela 3:** Acurácias Q3 e Q8 dos métodos de classificação de estruturas secundárias.

# Características Adicionais

- A partir da segunda fase de classificação de estruturas secundárias de proteínas, os métodos utilizam características adicionais.
- A principal característica adicional usada é a matriz de pontuação de posição específica.

# Matriz de Pontuação de Posição Específica

- Alinhamento de múltiplas sequências com alguma base de referência. Normalmente aplicada na base UniRef.
- A partir das proteínas homólogas, alguns passos são realizados:
  1. Calcular o número de cada aminoácido $j$ em uma posição específica $i$ de todas as $N$ proteínas homólogas;
  2. Gerar a pontuação de posição $i$ e aminoácido $j$, seguindo a equação:

$$\text{score}_{i,j} = \frac{F_{i,j}}{N} \tag{3}$$

  3. Calcular a frequência $P_j$ de cada aminoácido $j$;
  4. Gerar a matriz $\text{M}_{i,j}$ para o aminoácido $j$ na posição $i$ a partir do $\text{score}_{i,j}$ e pela taxa de frequência $P_j$:

$$\text{M}_{i,j} = \log(\frac{\text{score}_{i,j}}{P_j}) \tag{4}$$

# Transformers

- Ao final do mestrado, novos modelos baseados em Transformers para proteínas foram desenvolvidos pela comunidade.
- Na época, o Transformer utilizado como parte da abordagem era pré-treinado em inglês.
- Então, qual é o resultado com um Transformer pré-treinado em dados de proteínas comparado com a pesquisa do mestrado?

# Exercício Prático

- Vamos verificar como usar um Transformer pré-treinado em dados de proteínas para a classificação Q8 da base CB6166.
- Diferentemente do trabalho do mestrado, que fazia classificação do aminoácido central, iremos utilizar o paradigma de classificação de *tokens* da sequência. Isso permite que um aminoácido não veja apenas uma janela, mas sim toda a sequência de aminoácidos.
- O notebook pode ser acessado pelo link: `https://bit.ly/M0640protein1`.

# Classificação de Estruturas Terciárias

# Classificação de Estruturas Terciárias

- A análise de estrutura terciária indica a estrutura tridimensional global da proteína.
- Extensão dos resultados para estruturas secundárias.

# Avaliação

Como avaliar as predições dos modelos?

- RSMD: a métrica RSMD (*Root Mean Square Deviation*) mede as diferenças das posições atômicas do que foi predito com o que foi medido.
- TM-score: medida que avalia a similaridade global, considerando o que foi predito e o que foi verificado manualmente.

# História dos Métodos

- A principal competição para classificação de estruturas de proteínas é o desafio bianual CASP.
- Desafio bianual iniciado em 1994.
- CASP12 (2016):
  - Métodos baseados em homologia e similaridade de estruturas.
  - Progressos pequenos para métodos que não utilizavam modelos (proteínas conhecidas).

# AlphaFold

- Desenvolvido pela empresa DeepMind.
- Introdução do AlphaFold no CASP13, em 2018.
- Apresentado no artigo de Senior *et al.* (2020).
- Mostrou os melhores resultados da competição em 25 de 43 proteínas alvo.
- Ganhou a competição, com a medida GDT (*Global Distance Test*) de 58.9, contra 52.5 e 52.4 do segundo e terceiro lugar, respectivamente.

# AlphaFold

- Como entrada de dados, usa a sequência de aminoácidos e o alinhamento múltiplo de sequências (MSA).
  - O MSA é utilizado para capturar a coevolução de aminoácidos.
- A parte central do modelo envolve redes neurais convolucionais profundas.
- Como saída, prediz as distâncias entre os aminoácidos.
- A principal inovação é o uso de redes neurais profundas para processar dados evolutivos e prever interações de resíduos.

# AlphaFold

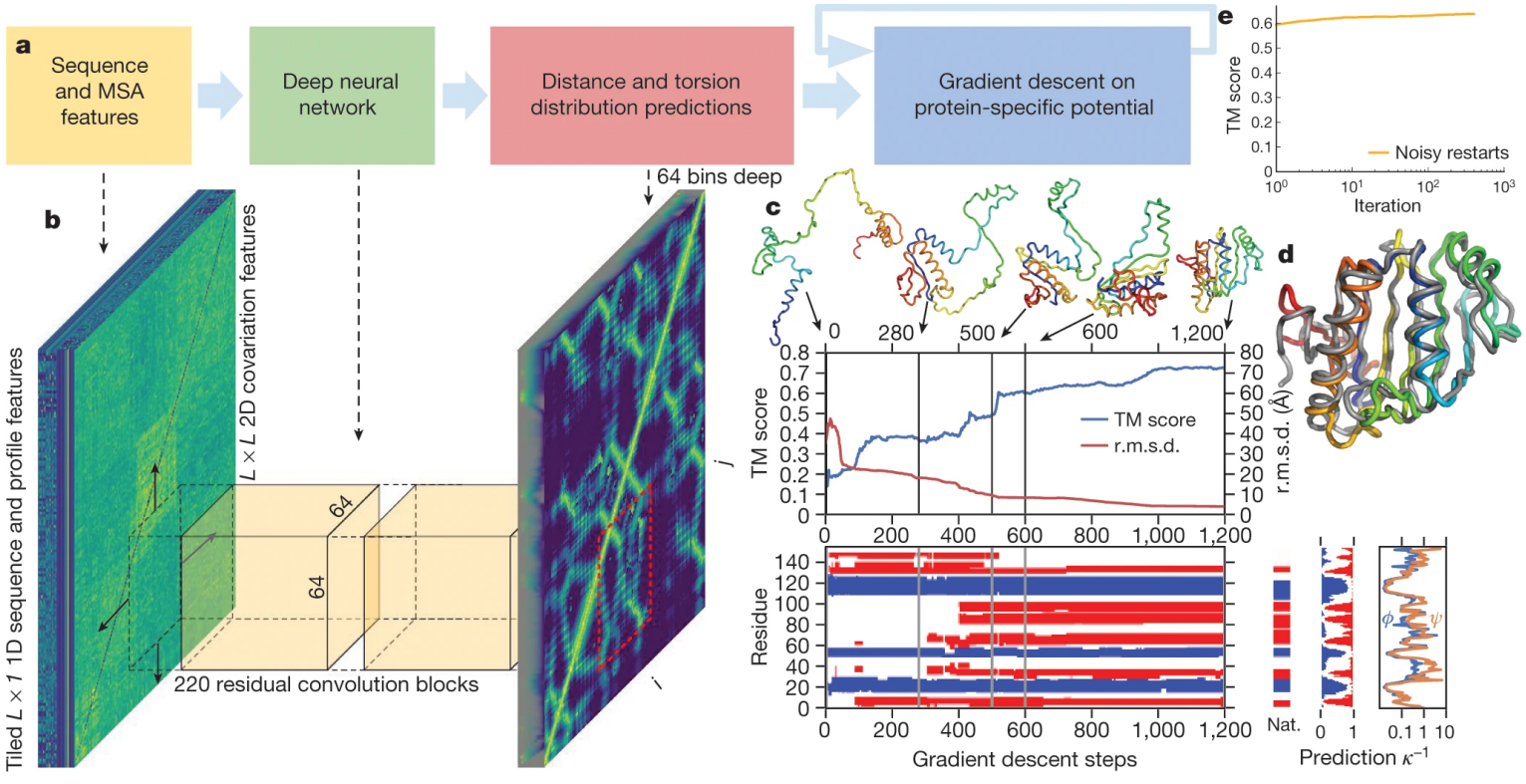
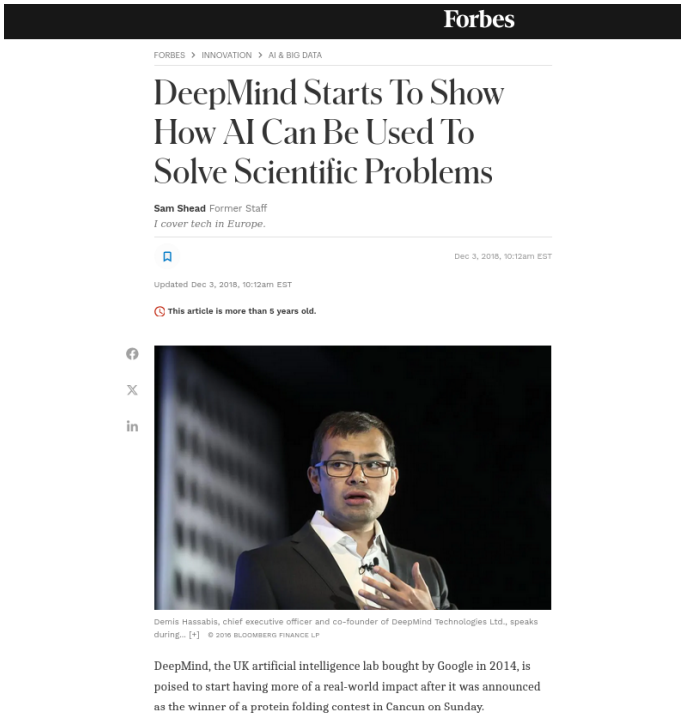



**Figura 12:** Arquitetura do AlphaFold.

# AlphaFold – Repercussão

Com o resultado alcançado pelo AlphaFold, publicações em diversas revistas, como Science, The Guardian, Forbes e The New York Times, indicaram um novo marco na biologia computacional e inteligência artificial.



Forbes

FORBES › INNOVATION › AI & BIG DATA

DeepMind Starts To Show How AI Can Be Used To Solve Scientific Problems

**Sam Shead** Former Staff
*I cover tech in Europe.*

Dec 3, 2018, 10:12am EST

Updated Dec 3, 2018, 10:12am EST

**This article is more than 5 years old.**

Demis Hassabis, chief executive officer and co-founder of DeepMind Technologies Ltd., speaks during... [+] © 2016 BLOOMBERG FINANCE LP

DeepMind, the UK artificial intelligence lab bought by Google in 2014, is poised to start having more of a real-world impact after it was announced as the winner of a protein folding contest in Cancun on Sunday.

**Figura 13:** Notícia da Forbes sobre o AlphaFold.

# AlphaFold – Limitações

Algumas limitações estão presentes no AlphaFold:

- O modelo depende de MSA para ter resultados satisfatórios.
- Dificuldade em analisar proteínas grandes.
- A versão do AlphaFold disponível no Github consegue apenas predizer estruturas das proteínas da base CASP13.

# AlphaFold2

- Introdução do AlphaFold2 no CASP14, em 2020.
- Apresentado no artigo de Jumper *et al.* (2021).
- Mostrou os melhores resultados da competição em 88 de 97 proteínas alvo.
- Ganhou a competição, com a medida GDT (*Global Distance Test*) de 92.4.

# AlphaFold2

- Assim como o AlphaFold, usa MSA e sequência de aminoácidos como entrada de dados.
- Substituição de convoluções por modulos atencionais (Evoformer).
- Transformação em um problema de inferência de grafos em três dimensões.
    - Relacionamento de aminoácidos no espaço.
- Pré-treinamento com dados de estruturas de proteínas.
    - Utilização de 128 TPUs em pré-treinamento por uma semana.
- Mascaramento ou mutação de aminoácidos do MSA durante o pré-treinamento.

# AlphaFold2

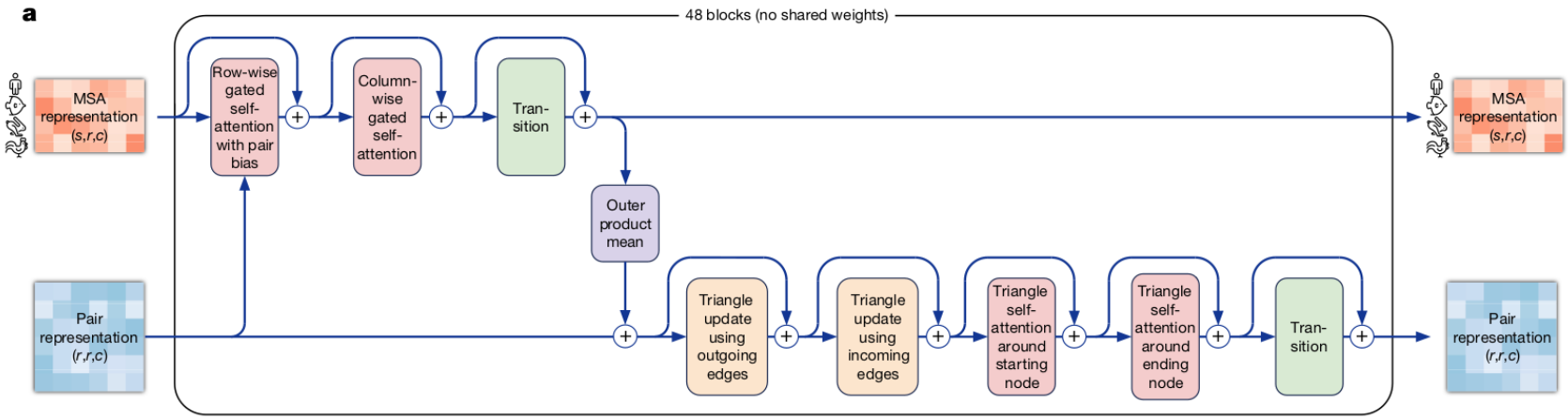



**Figura 14:** Arquitetura do Evoformer.

# AlphaFold2 – Base de Dados

- Diferentemente do AlphaFold, o AlphaFold2 consegue predizer estruturas tridimensionais de qualquer proteínas.
- Os resultados desse método são muito próximos da análise em laboratório feita por biólogos.
- Com isso, os autores criaram a base de dados AlphaFold, que possui as estruturas de quase todas as proteínas do UniProtKB na versão Swiss-Prot (proteínas sequenciadas manualmente).
- Link para a base: `alphafold.ebi.ac.uk`.

# AlphaFold2 – Repercussão

- A repercussão do AlphaFold2 foi ainda maior que o AlphaFold.
- Atualmente, o artigo do AlphaFold2 possui cerca de 28.000 citações.
- Em outubro de 2024, o primeiro autor (John M. Jumper) e o último autor (Demis Hassabis) do artigo receberam o Prêmio Nobel de Química, pelo trabalho de predição de estruturas de proteínas.

# AlphaFold2 – Limitações

- Mesmo resolvendo o problema da generalização do AlphaFold, o AlphaFold2 ainda é dependente do MSA.
- Proteínas com poucas sequências parecidas possuem resultados menos confiáveis.

# ESMFold

- Em 2022, um novo método é proposto, chamado ESMFold, desenvolvido pela empresa MetaAI.
- Apresentado no artigo de Lin *et al.* (2023).
- Arquitetura baseada em *encoder* pré-treinado com proteínas.
- Utilização somente da sequência de aminoácidos, sem precisar do MSA.
- Como saída de dados, indica a posição de cada aminoácidos no espaço tridimensional.
- Pré-treinamento em proteínas depositadas no PDB.

# ESMFold

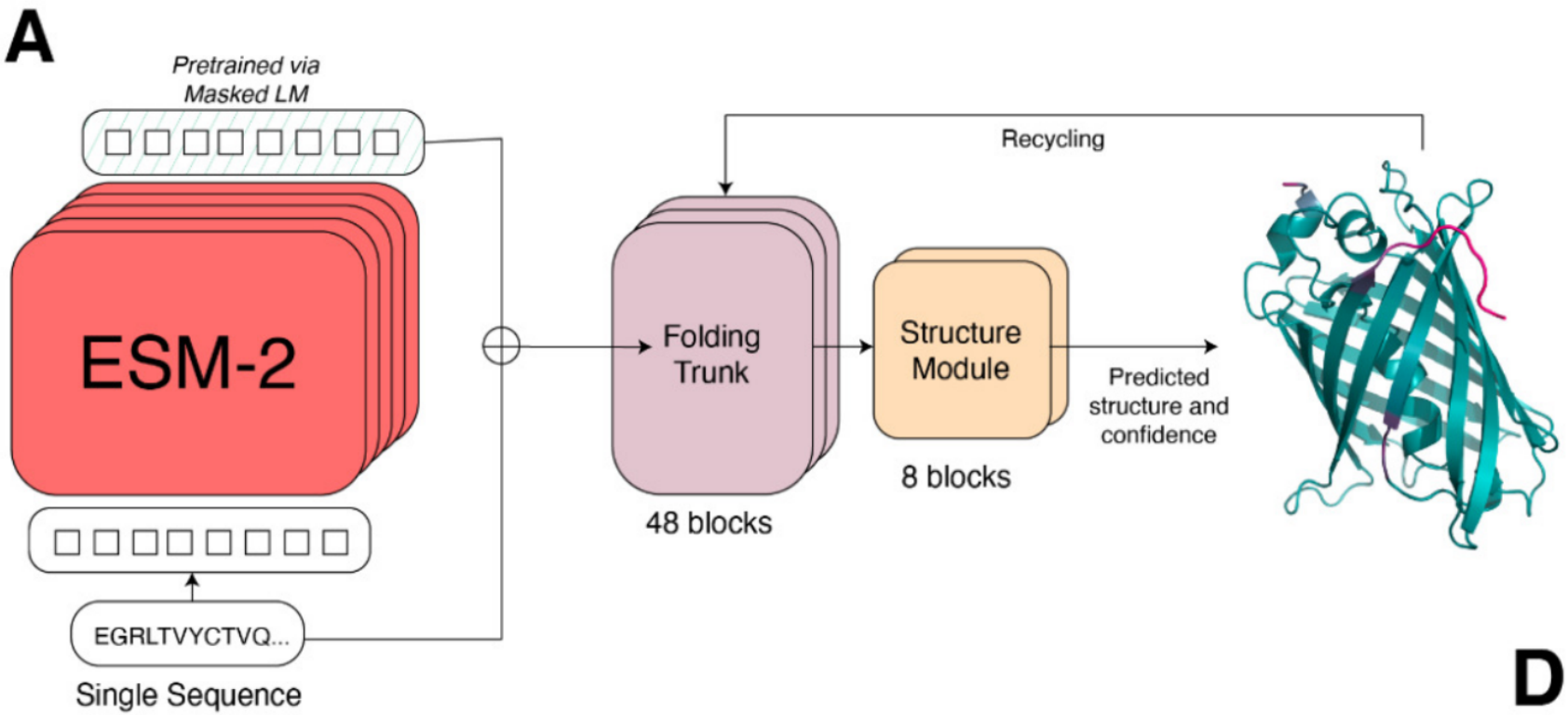



**Figura 15:** Arquitetura do ESMFold.

# ESMFold – Base de Dados

- Assim como o AlphaFold2, os autores do ESMFold disponibilizaram uma base de dados com estruturas tridimensionais preditas.
- Atualmente, a base conta com 772 milhões de estruturas avaliadas pelo ESMFold.
- Link para a base: `esmatlas.com`.

# AlphaFold2 x ESMFold

- AlphaFold2:
  - Considera as relações evolutivas das proteínas usando MSA.
  - Eficaz na modelagem de estruturas complexas.
- ESMFold:
  - Cerca de 6 vezes mais eficiente do que o AlphaFold2.
  - Resultados mais promissores quando a proteína possui poucos ou nenhum dado evolutivo.

# AlphaFold3

- Em 2024, a DeepMind apresenta o AlphaFold3.
- Nessa nova versão, o modelo é capaz de predizer as estruturas de grandes biomoléculas, como RNA, DNA e proteínas.
- Além do Evoformer, o modelo utiliza redes difusoras, encontradas em métodos de geração de imagem, para melhorar os resultados.

# Classificação de Funções

# Funções de Proteínas

- Objetivo: dada uma proteína, determinar que funções e em quais locais elas são realizadas.
- Anotações por ontologias:
  - FunCat.
  - Enzyme Classification.
  - Gene Ontology.
- A principal categorização é por Gene Ontology.

# Gene Ontology

O Gene Ontology (também chamado de Ontologia Genética) é dividido em três aspectos:

- Biological Process (BP, ou Processo Biológico): processo biológico em que a proteína está inserida, como tradução de sinal.
- Cellular Component (CC, ou Componente Celular): local em que a proteína exerce uma atividade, como mitocôndria.
- Molecular Function (MF, ou Função Molecular): atividade desempenhada em nível molecular, como transporte celular.

# Gene Ontology

- As três ontologias do Gene Ontology são estruturadas em um grafo acíclico direcionado.
- Termos mais próximo do nó raiz são mais genéricos (super-classes) enquanto termos mais profundos são mais específicos (sub-classes).
- Se uma proteína tem um termo, ela tem todos os ancestrais até o nó raiz.
- Uma proteína pode realizar mais do que uma função.
- Problema de classificação multirrótulo.

# Gene Ontology

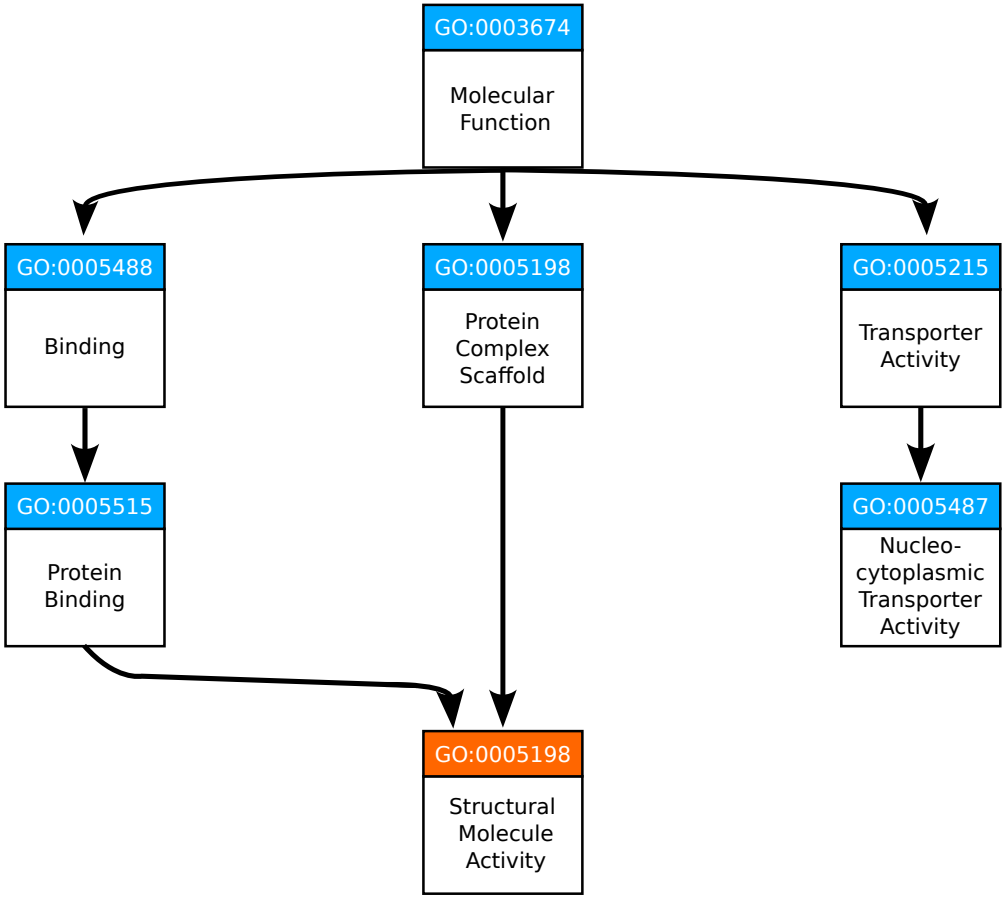



**Figura 16:** Uma proteína possui o termo GO:0005198.

# Gene Ontology

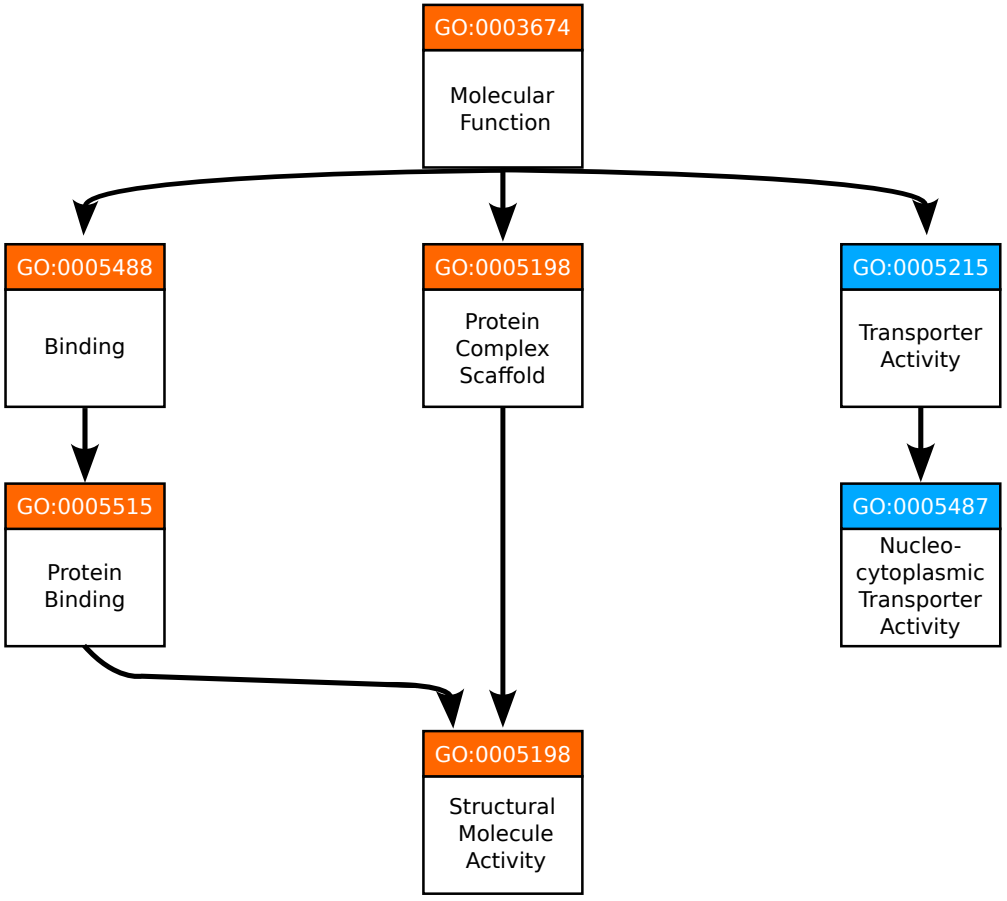



**Figura 17:** Essa proteína possui todos os termos até o nó raiz.

# Bases de Dados

- As principais bases utilizadas na literatura são desenvolvidas pelo desafio CAFA.
- No dasafio CAFA (*Critical Assessment of protein Function Anotation*), os organizadores disponibilizam proteínas sequenciadas que não possuem funções anotadas.
- Durante alguns meses, essas proteínas ficam disponíveis para a predição.
- Ao final do tempo estipulado, as predições terminam e os organizadores aguardam por alguns meses para que estas proteínas tenham os termos anotados.
- Após esse período, as proteínas que receberam verificações laboratoriais são utilizadas para avaliar os métodos propostos.

# Desafio CAFA

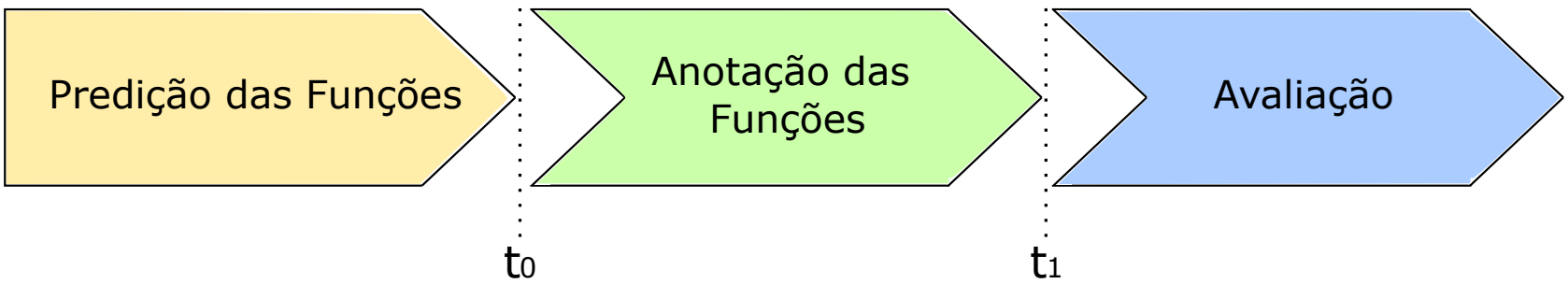



**Figura 18:** Método utilizado no desafio CAFA.

# Avaliação

- A principal avaliação aplicada para a classificação de funções de proteínas é o $F_{\text{max}}$.
- Essa medida avalia o valor de precisão (pr) e revocação (rc) em diversos limiares $\tau$, entre 0 e 1.

$$\text{pr}(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{|P_i(\tau) \cap T_i|}{|P_i(\tau)|} \tag{5}$$

$$\text{rc}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \frac{|P_i(\tau) \cap T_i|}{|T_i|} \tag{6}$$

$$F_{\text{max}} = \max_{\tau} \left\{ \frac{2 \times \text{pr}(\tau) \times \text{rc}(\tau)}{\text{pr}(\tau) + \text{rc}(\tau)} \right\} \tag{7}$$

# Avaliação

Outras medidas podem ser utilizadas para complementar as análises dos resultados:

- AuPRC: área sob a curva precisão e revocação gerada para todos os limiares do cálculo do $F_{\text{max}}$.
- IAuPRC: AuPRC interpolado, transformando em um medida mais justa para não penalizar métodos bons (com alta precisão e alta revocação).
- $S_{\text{min}}$: medida de distância semântica entre o que foi predito corretamente com os falsos positivos e falsos negativos, ponderado pela frequência de cada termo.

# História dos Métodos

- O Gene Ontology foi construído em 1998.
- Antes de 2010, métodos de aprendizado de máquina clássicos eram usados:
  - SVM (Support Vector Machine): Nenadić *et al.* (2003) e Cai *et al.* (2003).
  - Regressão logística: Ni *et al.* (2009).
  - Agregação de classificadores: Jeong *et al.* (2011).
- Alguns métodos utilizavam apenas alinhamento local.

# História dos Métodos

- Com o avanço de aprendizado profundo, novos métodos de aprendizado de máquina surgiram:
  - Redes convolucionais: Kulmanov *et al.* (2018) e Kulmanov e Hoehndorf (2019).
  - Redes recorrentes: Islam e Hasan (2018).
- Dessa forma, a aplicação desses métodos superaram as abordagens anteriores.

# História dos Métodos

- Com o desenvolvimento dos Transformers e a adaptação das arquiteturas para dados de proteínas, novas abordagens surgiram:
  - TALE (2021): Transformer com a parte *encoder* sem pré-treinamento.
  - ATGO (2022): Método utilizando o ESM-1b, arquitetura pré-treinada em dados de proteínas.
  - PU-GO (2024): Método utilizando o ESM2, uma das arquitetura mais recentes pré-treinadas em dados de proteínas.
  - PROTGOAT (2024): Método que utiliza cinco arquiteturas Transformers pré-treinadas em proteínas.
- Esses métodos também apresentam uma versão que agrega predições dos modelos de aprendizado de máquina com alinhamento local de proteínas, aumentando a eficácia final das abordagens.

# Método Proposto no Doutorado

- Duas abordagens:
  - Aprendizado de máquina.
  - Agregação de aprendizado de máquina com alinhamento local.

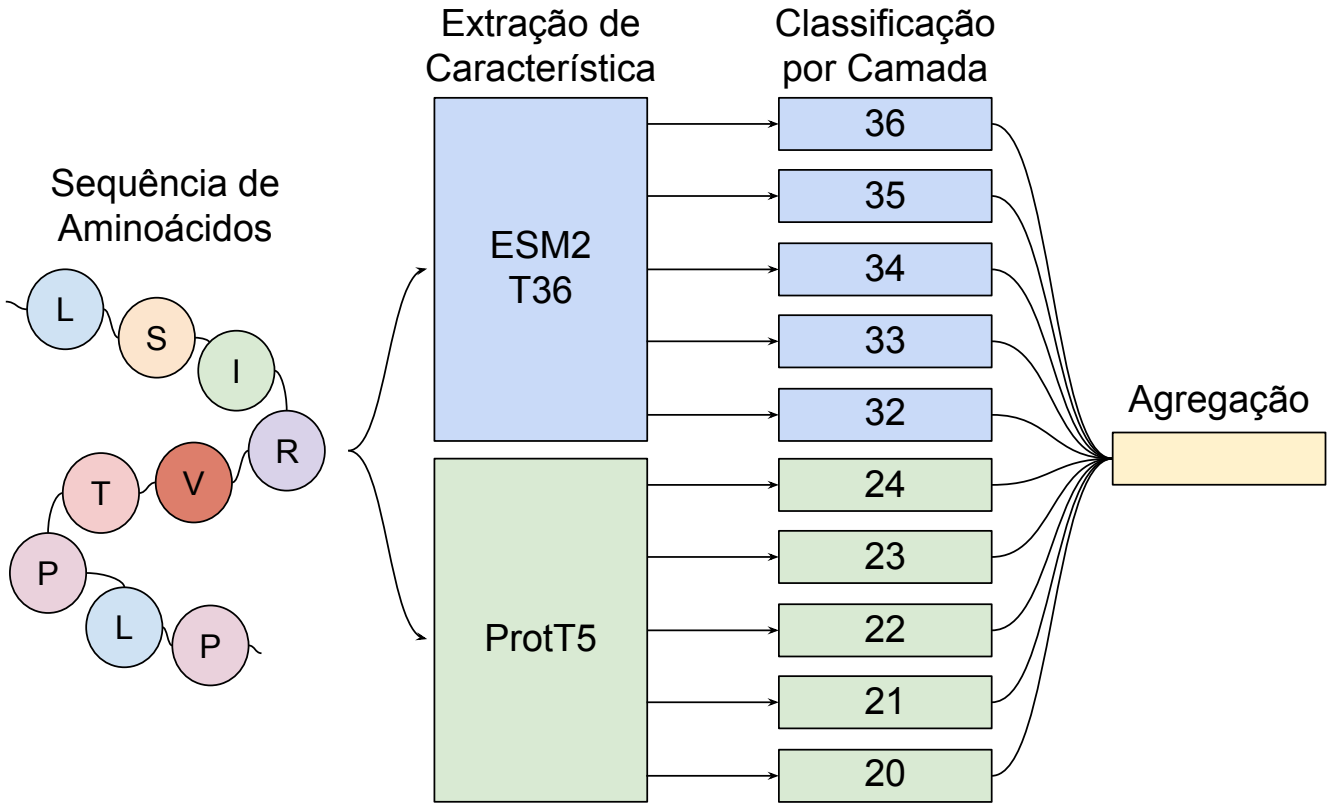



**Figura 19:** Método baseado em aprendizado de máquina.

# Extração de Características

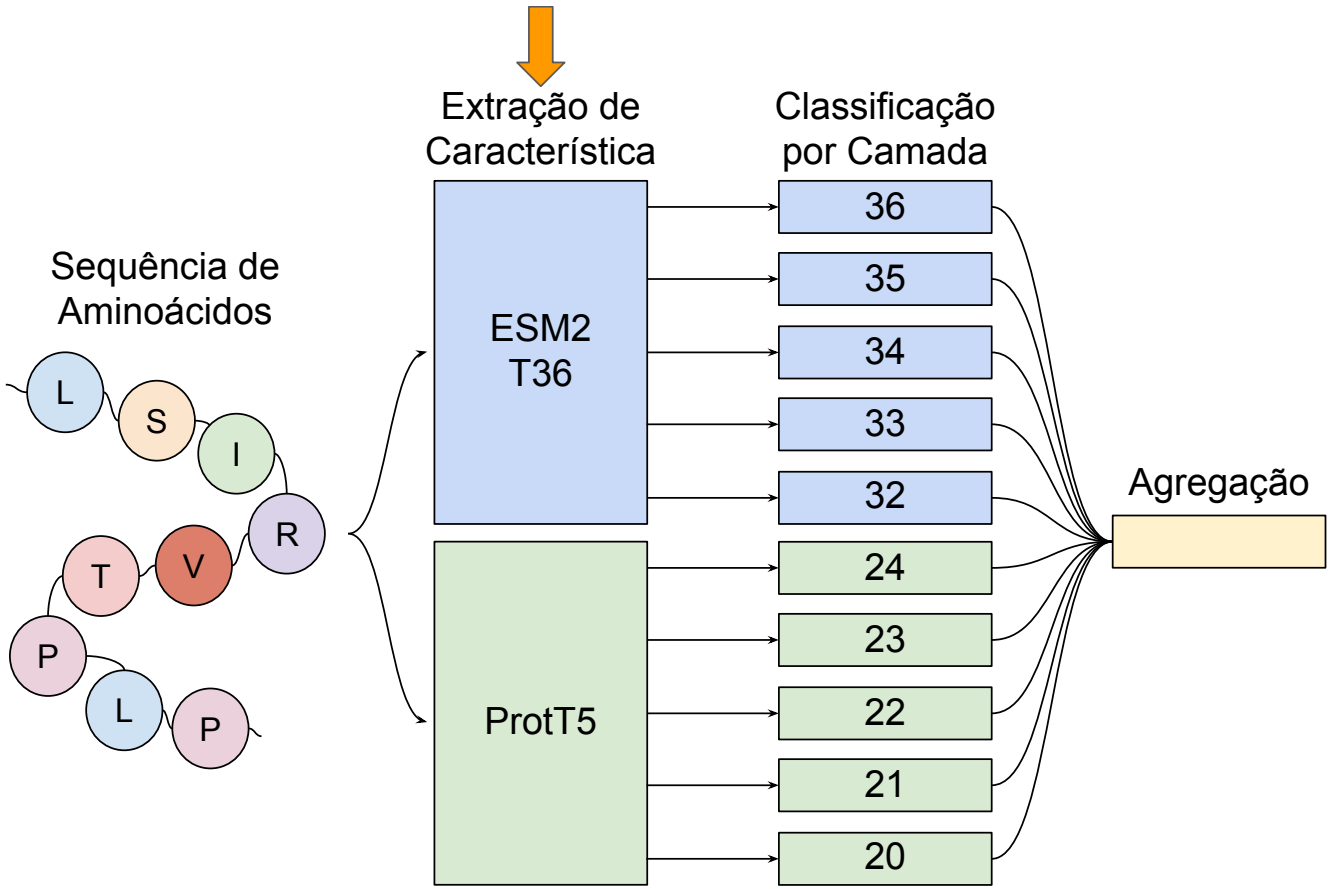



**Figura 20:** Método baseado em aprendizado de máquina.

# Extração de Características

- Utilização de dois modelos Transformers:
  - ESM2 T36.
  - Parte *encoder* do ProtT5.
- Obtenção das características das últimas cinco camadas:
  - Média das representações de cada aminoácido.

# Extração de Características

Como lidar com sequências grandes?

- Quebra da sequência em subsequências de 1.022 aminoácidos.
- Extração das características de cada subsequência.
- Média das características considerando as características de cada subsequência.

# Classificação por Camada

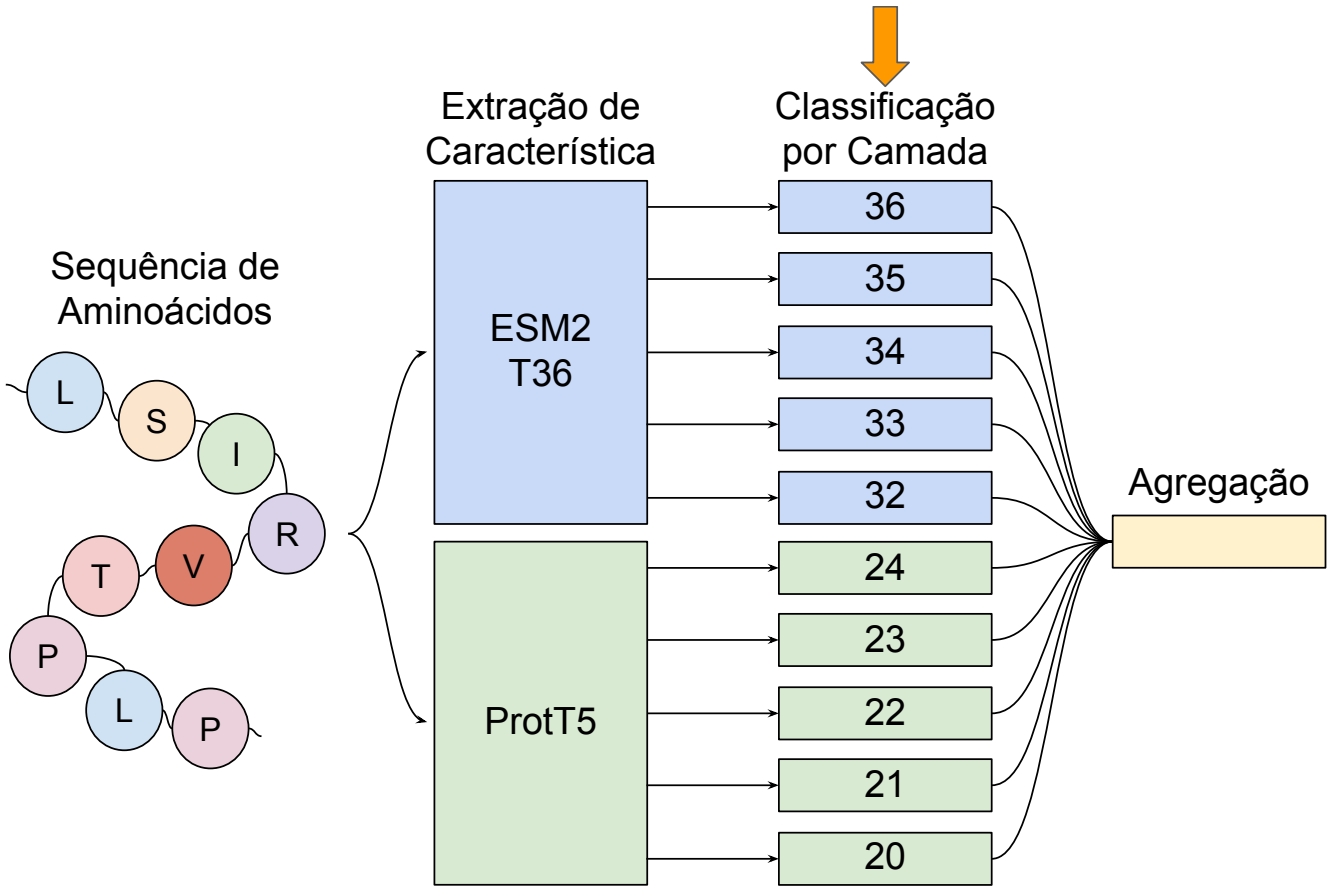



**Figura 21:** Método baseado em aprendizado de máquina.

# Classificação por Camada

- Treinamento de uma rede neural simples para classificação baseada na informação de cada camada.
- Ao final, para cada proteína, temos a classificação considerando dez classificadores, com informações complementares entre eles.

# Agregação

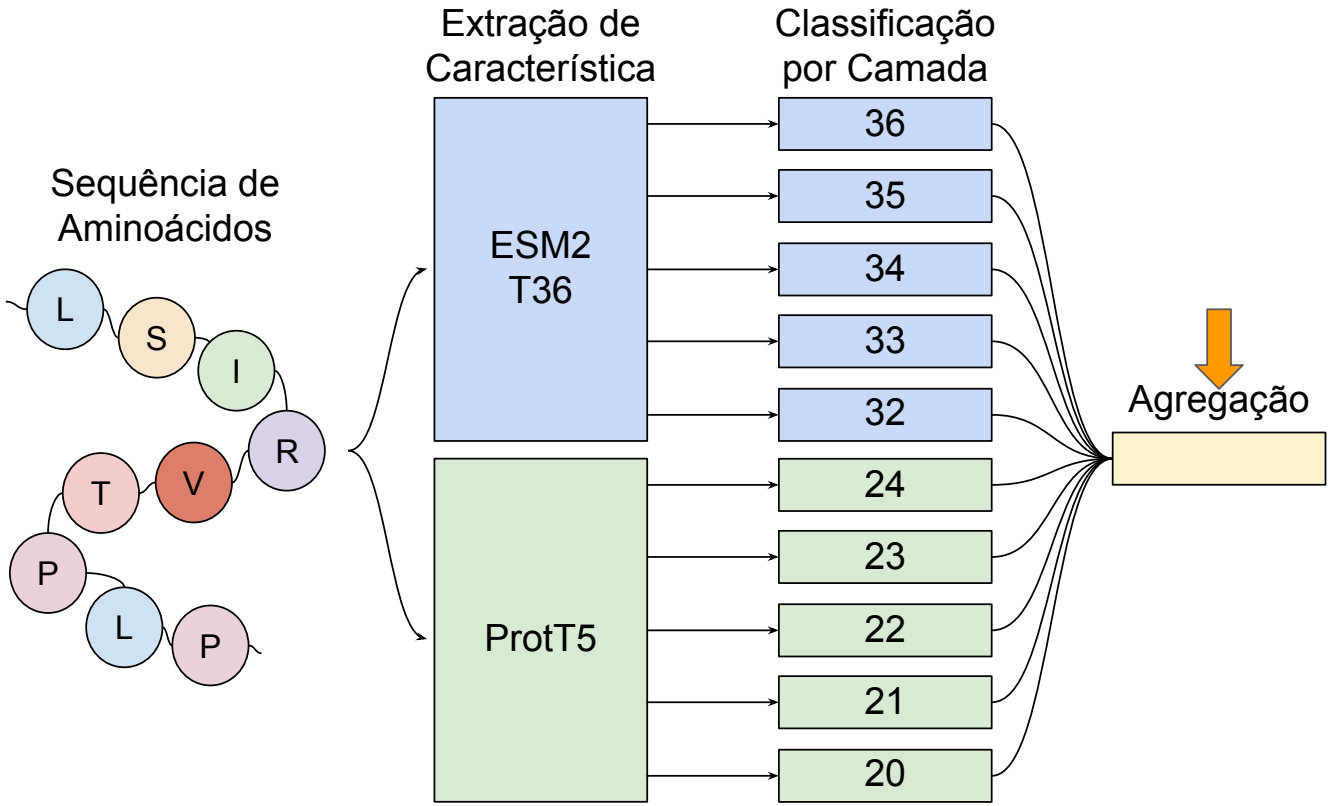



**Figura 22:** Método baseado em aprendizado de máquina.

# Agregação

- Por fim, é necessário agregar as predições de uma mesma proteína.
- Para isso, treinamos um meta-classificador baseado nas predições de validação.
- O meta-classificador possui pesos que somados são iguais a 1, para um termo específico.

# Agregação

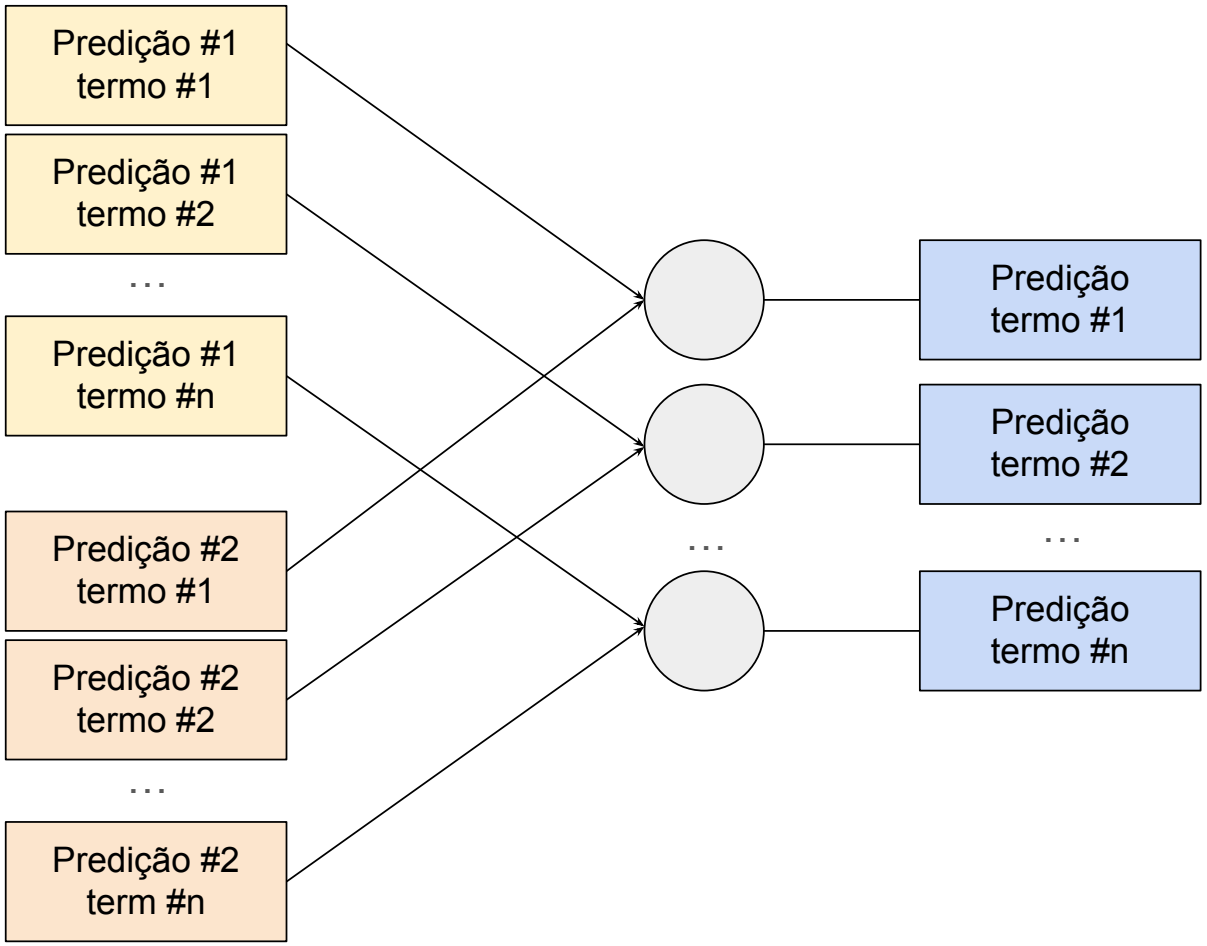



**Figura 23:** Agregação dos classificadores.

# Alinhamento Local

- Para utilizar alinhamento local, consideramos que proteínas com sequências similares possuem funções iguais.
- Execução de alinhamento local com a ferramenta DIAMOND.
- Ponderação dos alinhamentos pelo *bitscore* dos termos conhecidos.

# Alinhamento Local

$$S(p,t) = \frac{\sum_{s \in E} I(t \in T_s) \times \text{bitscore}(p,s)}{\sum_{s \in E} \text{bitscore}(p,s)} \tag{8}$$

- $S(p,t)$: predição de uma proteína $p$ e um termo $t$, entre 0 e 1.
- $s$: proteína qualquer do conjunto $E$ de proteínas que foram alinhadas com $p$.
- bitscore$(p,s)$: o valor de bitscore do alinhamento entre $p$ e $s$.
- $T_s$: conjunto de funções de $s$.
- $I()$: função que retorna 1, caso verdade, ou 0, caso contrário.

# Aprendizado de Máquina e Alinhamento Local

- Com as predições dos dois métodos, realizamos a agregação entre eles.
- O alinhamento local pode não gerar predição para algumas proteínas, caso não obtenha alinhamentos.
- Adaptação do método de agregação para lidar com esse fato:
  - Caso o alinhamento local tenha feito predições, possui o mesmo comportamento da agregação utilizada na metodologia do aprendizado de máquina.
  - Caso contrário, o peso da predição do aprendizado de máquina é 1 e o do alinhamento local é 0.

# Resultados

| Método | CC | FM | PB |
|---|---|---|---|
| Agregação | **0,619** | **0,777** | **0,809** |
| Aprendizado de Máquina | 0,582 | 0,769 | 0,802 |
| Alinhamento Local | 0,588 | 0,699 | 0,742 |

**Tabela 4:** Resultados na classificação de funções utilizando a medida $F_{\text{max}}$, para cada ontologia.

# Servidor Web

- Como parte da metodologia, desenvolvemos um servidor *web* que permite que usuários testem a abordagem.
- Para cada proteína, o usuário obterá os termos de cada ontologia, a confiança e o grafo desenhado.
- O servidor pode ser acessado pelo link: `https://supermago.ic.unicamp.br`.

# Características Adicionais

Além das características da sequência de aminoácidos, outras informações podem ser usadas:

- Estruturas tridimensionais.
- Interações de proteínas.
- Famílias e informações biológicas.
- Textos científicos.

# Estruturas Tridimensionais

- Alinhamento de estruturas de proteínas.
- Proteínas com estruturas similares possivelmente possuem ter funções similares.
- Estrutura é mais estável (conservada) que sequência de aminoácidos.
- Utilização de estruturas tridimensionais anotadas em laboratório ou obtidas por aprendizado de máquina (AlphaFold ou ESMFold).
- Informação não disponível para todas as proteínas.
- Ferramentas:
  - DALI.
  - Foldseek.

# Foldseek

- Desenvolvido por Kempen *et al.* (2024).
- Mapeamento das estruturas para sequências de caracteres.
- Aplicação de algoritmos de alinhamento local.
- 10.000 a 100.000 vezes mais rápido que os algoritmos padrões, como o DALI.
- Links para o algoritmo:
  - Artigo: `nature.com/articles/s41587-023-01773-0`.
  - Tutorial: `youtu.be/k5Rbi22TtOA`.

# Interações de Proteínas

- Análise de quais outras proteínas cada proteína se relaciona.
- Utilização dos dados da base STRING.
- Informação não disponível para todas as proteínas.
- Estudo de grafos de interação:
  - Ponderação pelas funções das proteínas próximas.
  - Random walks.
  - Node2vec.
  - Redes convolucionais em grafos.

# Famílias e Informações Biológicas

- Análise de informações biológicas e famílias.
- Proteínas com informações e famílias parecidas possuem funções parecidas.
- Dificuldade de existirem informações para todas as proteínas.
- Algumas características que podem ser utilizadas:
  - Famílias, com extração via base InterPro.
  - Informações biofísicas.
  - Domínio.

# Textos Científicos

- Busca de proteínas que possuam descrições parecidas com proteínas com funções conhecidas.
- Mineração de textos biológicos (artigos) que descrevem as funções das proteínas.
- Dificuldade para encontrar textos para todas as proteínas.

# Exercício Prático

- Vamos verificar como avaliar o DIAMOND para predição de funções de proteínas.
- Para fazer a predição, iremos ponderar os alinhamentos pelo bitscore.
- O notebook pode ser acessado pelo link: `https://bit.ly/M0640protein2`.