Protein Language Models
Proteins: Building Blocks of Life

- Support diverse functions: present in every cell of our body (enzymes, antibodies, hormones)
- Constructed from chains of 20 amino acids
- 3D structure determines their function
Protein Modeling

Opportunities:
Engineering novel, functional proteins *vaccines, drugs, enzymes, (bio)nano-materials, etc.)
Understanding fundamental biological phenomena
Protein Folding: Key Biological Question

Solving Protein 3D Structure with crystallography used to take a whole PhD

Previous Approaches to computationally predict protein folding had completely failed:

- Physics Based Approaches are too slow
- ML Approaches did not seem to have enough data
AlphaFold2 success

T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

T1049 / 6y4f
93.3 GDT
(adhesin tip)

- Experimental result
- Computational prediction

Image credit: DeepMind
Sequence vs Structure

Protein sequence

Protein structure
Role of Syntax in NLP

Syntax was considered a core computational problem in NLP.

Modern NLP systems do not use explicit syntactic information, operating directly at the sentence level.
Learning Protein Representations from Sequences

Predicting Protein Properties/Functions

Modeling Interactions

Generating New Proteins
Plan for Today

Transformer

Protein Representations

Applications
Inductive Bias Via Neural Architecture

- **RNN (SMILES)**
  - Input: sequence
  - Flow: follow sequence

- **GNNs (2D Molecules)**
  - Input: graph
  - Flow: along fixed edges

- **Attention-based Models**
  - Input: set (unordered)
  - Flow: dynamic control
Knowledge Grounding of Protein LMs

Inductive Biases reflect our knowledge of protein physics and geometry

The absolute positions of residues in the sequence are de-emphasized

Instead residues that are close in the folded protein need to communicate
Second Look at RNN

sentence as a vector

Efforts and courage are not ...

Encoding

Decoding

sampled word = Olen nähnyt parem. luentoja <end>

\[
p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4 \rightarrow p_5 \rightarrow \text{sentence as a vector}
\]
Second Look at RNN

- Handles input of any length
- Model Size is independent of input length
- Shared Weights
- (Theoretically) Flexibly Encodes History

- Recurrent computation is slow
- Difficult to access information many steps back

**Encoding**  
**Decoding**
More Colorful Way to Describe RNNs

You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!

Prof. Ray Mooney (UT Austin)
RNN Alternative: Attention

Key idea: enable model to focus on relevant information

Keep more intermediate context but provide refined access controls

Proxy to word alignment in machine translation where it was developed

Animation Source: J. Alammar
Decoder with Attention

Attention at time step 4

Graphics Source: J. Alammar
Attention Score: Dot Product of Query&Keys
Attention Distribution vis à vis Softmax

Keys

Je  suis  étudiant  I

Dot Product

Query 1
Attention Output: weighted average of hidden states

Dot Product

Keys

Je | suis | étudiant

Query 1
suis

Je étudiant

Keys

Attention Output: weighted average of hidden states

Dot Product

1. Concatenate attention output with hidden state

2. Run through softmax to predict next word
Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL WITH ATTENTION
Je suis étudiant
Attention Helps

Attention allows decoder to look directly at the source

Provides shortcut to faraway states (helps with vanishing gradients)

Provides a form of interpretability (i.e., computes alignment)
Transformers: Attention on Steroids

Attention in both encoder and decoder

Key advantage: speed due to parallelization

New Terms: Self Attention, Multi-Headed Attention, Positional Encoding
Transformers: Encoder

Note parallel processing: no dependencies in FF!!!
Computing Attention
Query/Value/Key

Why Values?

Three roles a word embedding play:

1. Encode their own context
2. Serve as context for computing other embeddings
3. Transfer Information for the next layer

Values provide extra encoding flexibility
Computing Attention: One Implementation

\[ A(q, K, V) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{k_j}} v_i \]

query keys values

\[ x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \]

parameters we learn

\[ W^q \quad Q \]

\[ W^k \quad K \]

\[ W^v \quad V \]
Computing Attention: More Elaborate Implementation

\[
X \times W^Q = Q
\]

\[
X \times W^K = K
\]

\[
X \times W^V = V
\]
The animal didn't cross the street because it was too tire.
Multi-Headed Attention

Idea: Project input embeddings into a different representation spaces

Advantage: richer representation via more flexible blending

Graphics Source: J. Alammar
Multi-Headed Attention: Combining heads

1) This is our input sentence*  
2) We embed each word*  
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W\textsuperscript{o} to produce the output of the layer

* In all encoders other than #0, we don’t need embedding. We start directly with the output of the encoder right below this one

Graphics Source: J. Alammar
Single vs Multi-Headed Attention

Source: J. Alammar
Positional Encoding

Encoder needs to preserve relative word positioning

Idea: enrich embeddings with positional information so that the distances are meaningfully represented in the projected space
Positional Encoding

\[ PE_{pos, 2i} = \sin \left( \frac{pos}{10000 \cdot \frac{2i}{d}} \right) \]

POS: position within a sentence

I: coordinate within embedding

d: embedding dimension

A positional vector for the jth position in the sequence.
Decoding
Decoding

Decoding time step: 1 2 3 4 5 6

OUTPUT

ENCODERS

K_{encdec}  V_{encdec}

LINEAR + SOFTMAX

DECODERS

EMBEDDING WITH TIME SIGNAL

EMBEDDINGS

INPUT: je  suis  étudiant

PREVIOUS OUTPUTS
Where Initial Embeddings Come From?

One option: one-hot embeddings (sparse vectors of vocabulary size)
Desiderata for Embeddings

dogs

0
0
0
0

1
0

\|V\|

\longrightarrow

0.19
0.73
-0.10
1.15
-0.09
0.92
0.46

-0.32

\|V\|

d << \|V\|

dogs

0.93

samoyed

0
0
0
0

1
0

\|V\|

\longrightarrow

-0.30
0.18
0.75
-0.06
1.09
-0.12
0.89
0.53

cosine similarity = 0.93

x

y
Pretraining
Unsupervised Learning of Word Embeddings

Key Idea: Create artificial training tasks by masking words

i went to the [mask] to buy a [mask] of milk
Size of Foundational Models

![Graph showing the trend of sizes of state-of-the-art NLP models over time](https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/)

**Figure 1. Trend of sizes of state-of-the-art NLP models over time**

- ELMo (94M)
- BERT-Large (340M)
- GPT-2 (1.5B)
- T5 (11B)
- Turing-NLG (17.2B)
- Megatron-LM (8.3B)
- GPT-3 (175B)
- Megatron-Turing NLG (530B)
Protein Language Models

Pretraining Objective is based on Masking (Unsupervised):

Options: One position, multiple positions spans for a single protein, multiple positions for MSA

\[
\mathbb{E}_M \left[ \sum_{m \in M} - \log p(x_m | x_{/M}) \right] \quad \mathbb{E}_{z \sim z_m} \left[ \sum_{i=1}^{m} \sum_{j=1}^{l_i} - \log p(s_{z_i,j} | x_{\text{corrupt}}, s_{z_{<i}}, s_{z_i,<j}) \right]
\]

Size of Training Data: 360M+ sequences (1T symbols)

Number of Parameters: 100M+ to 100
Training Data for Protein Language Models

[Image of a pie chart showing the distribution of training data across different bacterial and eukaryotic species.]

https://www.biorxiv.org/content/10.1101/2023.07.05.547496v1.full.pdf
Model Retrieves Structural Classes of Proteins

A. Protein sequences → Language model → Contextual vector embeddings → Summary vector embeddings → t-SNE for visualization

B. Edit Distance (no structure supervision) → DLM-LSTM (no structure supervision) → MT-LSTM (with structure supervision)
Model Retrieves Structural Classes of Proteins

1: DNA-binding transcription repressor activity, RNA polymerase II-specific (GO:0001227, n=324)
2: transmembrane transporter activity (GO:0022857, n=107)
3: neuropeptide hormone activity (GO:0005184, n=30)
4: cytoplasmic translation (GO:0002181, n=87)
5: odorant binding (GO:0005549, n=118)
6: delayed rectifier potassium channel activity (GO:0005251, n=32)
7: protein glycosylation (GO:0006486, n=89)
8: immunoglobulin complex (GO:0019814, n=90)
9: collagen trimer (GO:0005581, n=60)

Common Evaluation Tasks

- **Contact Prediction**: predict whether two residues are in the pre-defined proximity
- **Fold Prediction**: assign sequence to one to 1,195 known folds
- **Secondary Structure Prediction**: classify sequence into Helix, Strand, Coil
- **Solubility**: binary
- **Stability**: whether protein remains folded
- **Localization**: distribution into 100 Unique Subcellular categories
- **TCR-pMHC Affinity**: interaction between T cell receptors and peptide-major hits compatibility complex
- ....
More Complex Models Do Better
Interpreting Attention

(a) Attention in head 12-4, which targets amino acid pairs that are close in physical space (see inset subsequence 117D-157I) but lie apart in the sequence. Example is a de novo designed TIM-barrel (5BVL) with characteristic symmetry.

(b) Attention in head 7-1, which targets binding sites, a key functional component of proteins. Example is HIV-1 protease (7HVP). The primary location receiving attention is 27G, a binding site for protease inhibitor small-molecule drugs.
Interesting Case Studies: Exploring Metagenomics

ESM Metagenomic Atlas
An open atlas of 772 million predicted metagenomic protein structures

Explore →

Fold sequence
Read blog post
Read research paper
Interesting Case Studies: Covid Evolutions

Example 1: Controlling Molecular Localization through Protein IDRs

IDRs shown to contribute to transmembrane transport, transcription, and phase separation.
Localization tags have been well-established for some compartments (e.g., nucleus, mitochondrion).

Partitioning in condensates as well as some membrane-bound compartments (e.g., peroxisome) depend on more complex chemical patterns encoded in the sequence.
Learning to Predict Protein Localization

Fine-tuned ESM2

{ endosome, ER, golgi, stress granule }

{ nucleus }

{ mitochondrion, P-body }
**Protein Localization Classifier**

**Dataset**
- 47k Sequences: Membrane Bound
- 4k Sequences: Condensate Forming

**Membrane Bounding**

<table>
<thead>
<tr>
<th>Structure</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nuclear Membrane</td>
<td>14k</td>
</tr>
<tr>
<td>Nucleus</td>
<td>7k</td>
</tr>
<tr>
<td>Mitochondrion</td>
<td>1k</td>
</tr>
<tr>
<td>Lysosome</td>
<td>1k</td>
</tr>
</tbody>
</table>

**Condensates**

<table>
<thead>
<tr>
<th>Structure</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nuclear Speckle</td>
<td>2k</td>
</tr>
<tr>
<td>PML Body</td>
<td>1k</td>
</tr>
<tr>
<td>PSD</td>
<td>1k</td>
</tr>
<tr>
<td>Transcriptional</td>
<td>1k</td>
</tr>
</tbody>
</table>

**Classification Performance**

- AUC-ROC:
  - Nuc. Speckle: 1.0
  - Transcriptional Cond.: 1.0
  - PML Body: 1.0
  - ER: 1.0
  - Nucleus: 1.0
  - Membrane: 1.0
  - Golgi App.: 1.0
IDR Generation

- Decode using compartment classifier to generate IDRs for specific compartments

GFP Sequence:

MSKGEELFTG...ACGHTHGMDELYK

GFP:

IDR:

MSK...LYKVRFR...YFPHL
Example 2: Modeling Broadly Neutralizing Antibodies
How to Model Neutralization?

**VH**: QKQLVY...VIVSS  
**VL**: QSVLTQ...TGTKV

**ENV**: MRVKE...GFERALL

Probability of neutralization at 4 different IC50 potencies
How to Model Neutralization?

1. Encoding Amino Acid sequences into high-dimensional vector space

VH: QKQLVY...VIVSS
VL: QSVLTQ...TGTKV

ENV: MRVKE...GFERALL

Encoder 🔗 Encoder

Self-attentive Recurrent Neural Network

<table>
<thead>
<tr>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
</tr>
<tr>
<td>0.9</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>-0.2</td>
</tr>
<tr>
<td>1.6</td>
</tr>
<tr>
<td>-1.3</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.8</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>-0.2</td>
</tr>
<tr>
<td>1.6</td>
</tr>
<tr>
<td>-1.3</td>
</tr>
</tbody>
</table>
How to Model Neutralization?

2. Fuse the learned features from both inputs and predict the neutralization strength

\[ \text{CATNAP HIV Database:} \]
\[ \sim 300 \text{ Antibodies: VH + VL amino acid sequence} \]
\[ \sim 1000 \text{ HIV Viral strains: Env amino acid sequence} \]
\[ \sim 32K \text{ [antibody, virus] pairs with measured IC50}^* \]
Find broadly neutralizing antibodies (bnAbs)

Using our developed neutralization predictor for HIV, we can now identify broadly neutralizing antibodies by running the model against many HIV strains.
Can we design better antibodies?

- Our goal: modify the CDR sequence of an existing antibody (e.g. VRC01) to improve its neutralization breadth

VRC01 heavy chain

VRC01 light chain

Modify CDR

New antibody sequence

- Finding better antibody is a challenging search problem

- 10 mutations \( \Rightarrow 2^{10} \) possible sequences.
Model-Based Protein Engineering
Landscape Challenges

Starting data → $x$, $f_{θ}(x)$ → Proposal → Fitness oracle

Noisy Landscape Estimate

Low Fitness Barrier
Apply L1 graph Laplacian Regularization to ensure local consistency
Bi-Level Gibbs Sampling

BiGGS sampling

\[ f_\theta(x) \]

Current fitness

\[ \nabla f_\theta(x) \]

Compute gradients

Step 1: sample residue locations

D G N Y K T R A E V K F E

Step 2: sample mutations at locations

I G N L K F R A E K K F E

D. Iterative Extrapolation with Bi-level Gibbs sampling

E. Mutation trajectory

Samples multiple mutations supporting diversity and novelty