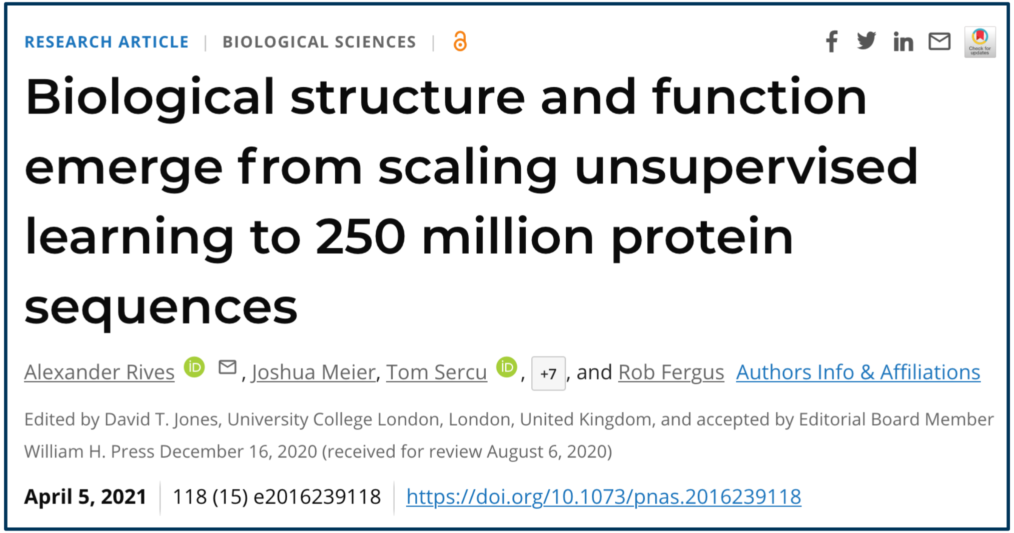
The ESM-1b Transformer Network

What is the ESM-1b Transformer Network?



RESEARCH ARTICLE | BIOLOGICAL SCIENCES |

Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences

Alexander Rives, Joshua Meier, Tom Sercu, +7, and Rob Fergus Authors Info & Affiliations

Edited by David T. Jones, University College London, London, United Kingdom, and accepted by Editorial Board Member William H. Press December 16, 2020 (received for review August 6, 2020)

April 5, 2021 | 118 (15) e2016239118 | https://doi.org/10.1073/pnas.2016239118

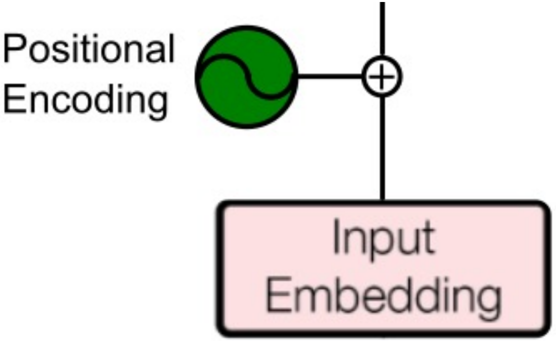
- Open-Source protein language model by the Facebook AI research team
- Architecture: Transformer Network Encoder
- Training for Masked Language Modeling (MLM)

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., ... & Fergus, R. (2021). *PNAS*, *118*(15), e2016239118.

ESM-1b Architecture – Input Embeddings

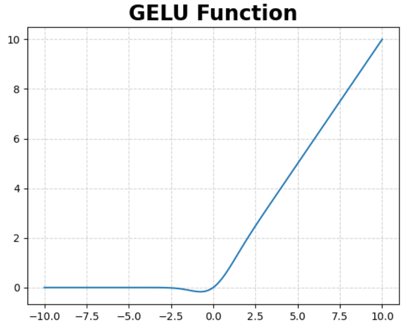
- Token embeddings are learned with a dimension of $d = 1280$
- Sinusoidal positional encodings were compared to learned positional encodings:
  - Learned positional embeddings led to better results
- How can we learn token (and positional) embeddings?

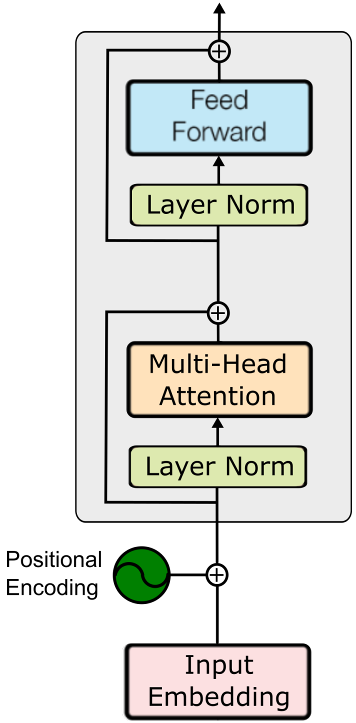
$$\begin{pmatrix} w_{1,1} & \cdots & w_{1,1280} \\ \vdots & \ddots & \vdots \\ w_{23,1} & \cdots & w_{23,1280} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,1280} \end{pmatrix}$$



ESM-1b Architecture – Encoder

- Each encoder layer
  - Dimension of representations $d = 1280$
  - Number of attention heads $h = 20$
  - Maximum sequence length: 1024
  - Hidden dimension of FFN: $5120\ (4 \cdot d)$
- 33 encoder layers: $652.4\ M$ learnable parameters
- Layer Normalization before Attention and FFN block
- Activation of Feed Forward Neural Network:
  GELU instead of ReLU





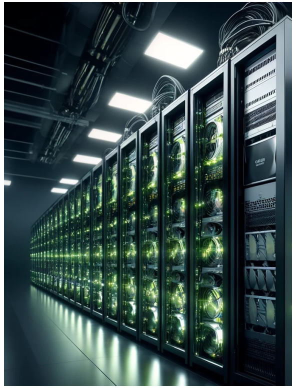
GELU: medium.com/@gauravnair :“The Spark Your Neural Network Needs“; Modified figure from: Vaswani, A., et al. (2017). Attention is all you need.

ESM-1b Training

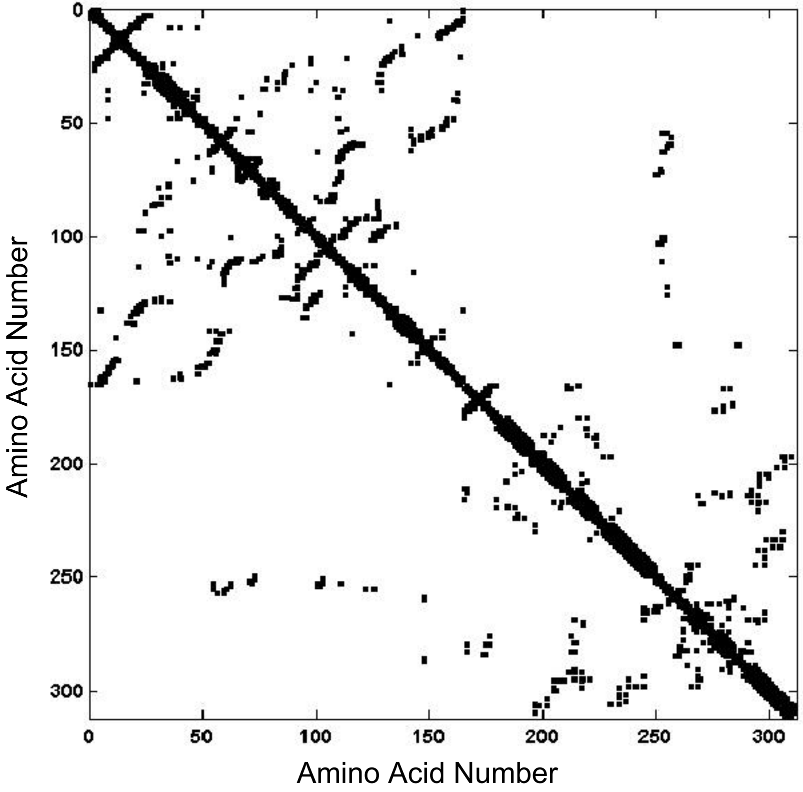
- Training task: Masked Language Modeling (MLM)

A	L	?	A	…	A

  - Randomly select 15% of the input tokens that shall be predicted: Of these:
    - 80% are replaced with the mask token
    - 10% are placed with a random amino acid
    - 10% are not changed
- Training dataset: UniRef50 dataset with $\sim 30M$ protein sequences
- Training time: ~20 days of 64 NVIDIA V100 GPUs
  - Each epoch took 8.5 hours
  - 56 epochs in total



Applications – Contact Map Prediction



- Protein contact maps
  - Black dots indicate which amino acids are close to each other in the protein 3D structure
- Predicting Contact Maps using amino acid representations shows learned structural information
  - Binary prediction problem:
    - Let h_i and h_j be the representations of amino acids i and j
    - $f(h_i, hj) = sigmoid(< P \cdot h_i, Q \cdot h_j > \ +b) \in [0,1]$

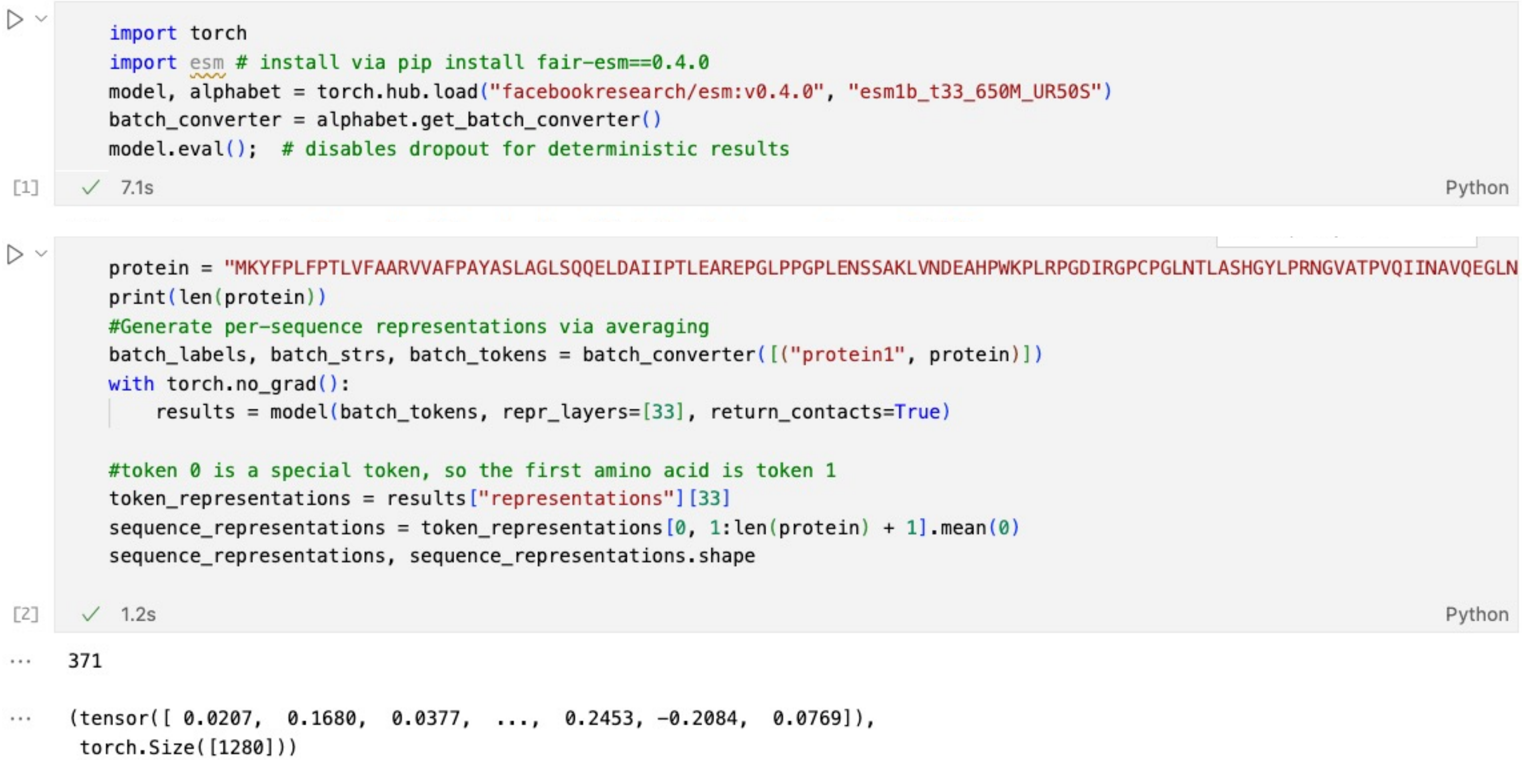
Image source: Ramanathan, A et al. (2008). *Using tensor analysis to characterize contact-map dynamics of proteins* (PhD thesis)

Applications – Function Prediction

- ESM-1b vectors (whole protein representations) have been successfully used for all kinds of protein function predictions:
  - Enzyme functions
  - Substrates for transport proteins
  - Protein-Protein interactions
  - Drugs inhibiting proteins
  - …
- To solve these prediction tasks, the ESM-1b vectors $\vec{e}$ (element-wise mean of all amino acid representations) are used as the input of new prediction models

$$f(\vec{e}) = y$$

How to use the model



```
import torch
import esm # install via pip install fair-esm==0.4.0
model, alphabet = torch.hub.load("facebookresearch/esm:v0.4.0", "esm1b_t33_650M_UR50S")
batch_converter = alphabet.get_batch_converter()
model.eval();  # disables dropout for deterministic results
```

```
protein = "MKYFPLFPTLVFAARVVAFPAYASLAGLSQQELDAIIPTLEAREPGLPPGPLENSSAKLVNDEAHPWKPLRPGDIRGPCPGLNTLASHGYLPRNGVATPVQIINAVQEGLN
print(len(protein))
#Generate per-sequence representations via averaging
batch_labels, batch_strs, batch_tokens = batch_converter([("protein1", protein)])
with torch.no_grad():
    results = model(batch_tokens, repr_layers=[33], return_contacts=True)

#token 0 is a special token, so the first amino acid is token 1
token_representations = results["representations"][33]
sequence_representations = token_representations[0, 1:len(protein) + 1].mean(0)
sequence_representations, sequence_representations.shape
```

```
371
```

```
(tensor([ 0.0207,  0.1680,  0.0377,  ...,  0.2453, -0.2084,  0.0769]),
 torch.Size([1280]))
```

Code copied from https://github.com/facebookresearch/esm

Other Protein Language Models

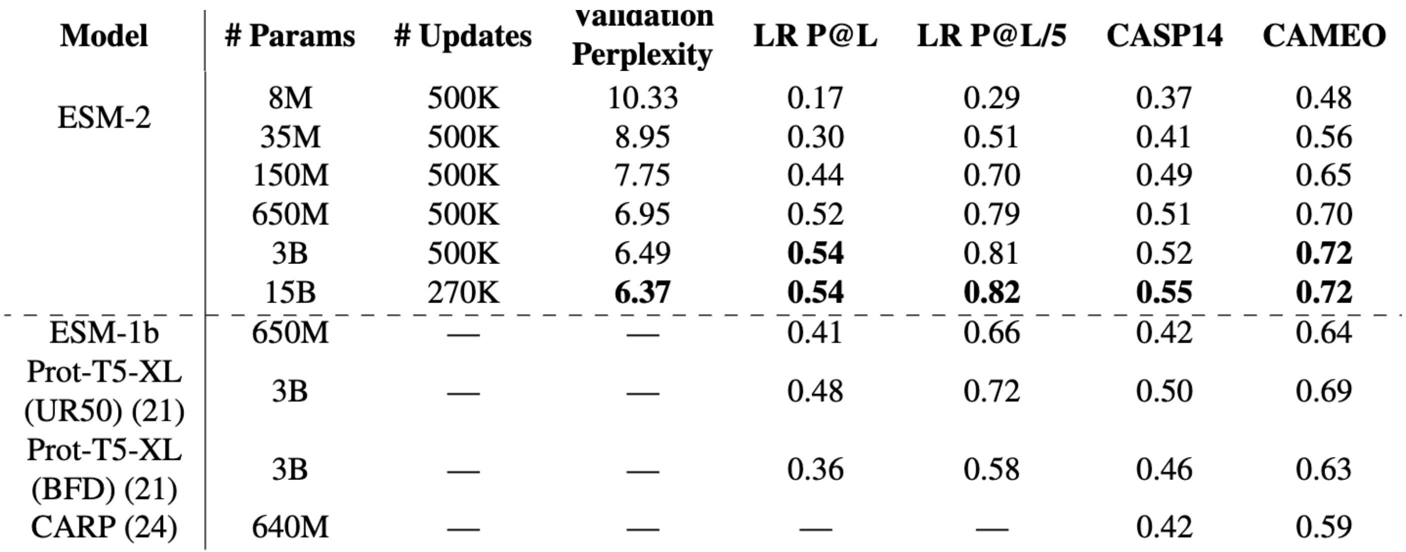
- ESM-2 (successor of ESM-1b)
  - Family of protein language models with different sizes: 8M to 15B
  - ESM-2 are Transformer Network encoder trained for masked language modeling (such as ESM-1b)
  - Improvements:
    - Data: ~65M unique protein sequences
    - Architecture: Different positional encodings (RoPE)
    - Training: Removed dropout
    - Computational resources

ESM-2 paper: DOI: 10.1126/science.ade2574

Other Protein Language Models (2)

- ProtT5 (ProtT5-XL-U50)
  - Training task:
    - Masked language modeling with masking probability of 15%
  - Training data:
    - UniRef50: ~45M unique protein sequences
  - Model architecture:
    - T5 Transformer Network consisting of encoder and decoder (for training)
    - Only uses the encoder for generating protein representations
    - 3B learnable parameters

Comparison between protein language models



Model	# Params	# Updates	Validation Perplexity	LR P@L	LR P@L/5	CASP14	CAMEO
ESM-2	8M	500K	10.33	0.17	0.29	0.37	0.48
	35M	500K	8.95	0.30	0.51	0.41	0.56
	150M	500K	7.75	0.44	0.70	0.49	0.65
	650M	500K	6.95	0.52	0.79	0.51	0.70
	3B	500K	6.49	**0.54**	0.81	0.52	**0.72**
	15B	270K	**6.37**	**0.54**	**0.82**	**0.55**	**0.72**
ESM-1b	650M	—	—	0.41	0.66	0.42	0.64
Prot-T5-XL (UR50) (21)	3B	—	—	0.48	0.72	0.50	0.69
Prot-T5-XL (BFD) (21)	3B	—	—	0.36	0.58	0.46	0.63
CARP (24)	640M	—	—	—	—	0.42	0.59

Modified table from: https://doi.org/10.1101/2022.07.20.500902