Protein representations

Encoding protein sequences for machine learning

Why Train a Machine Learning Model?

✅ Reduce computational cost

✅ Quickly test hypotheses

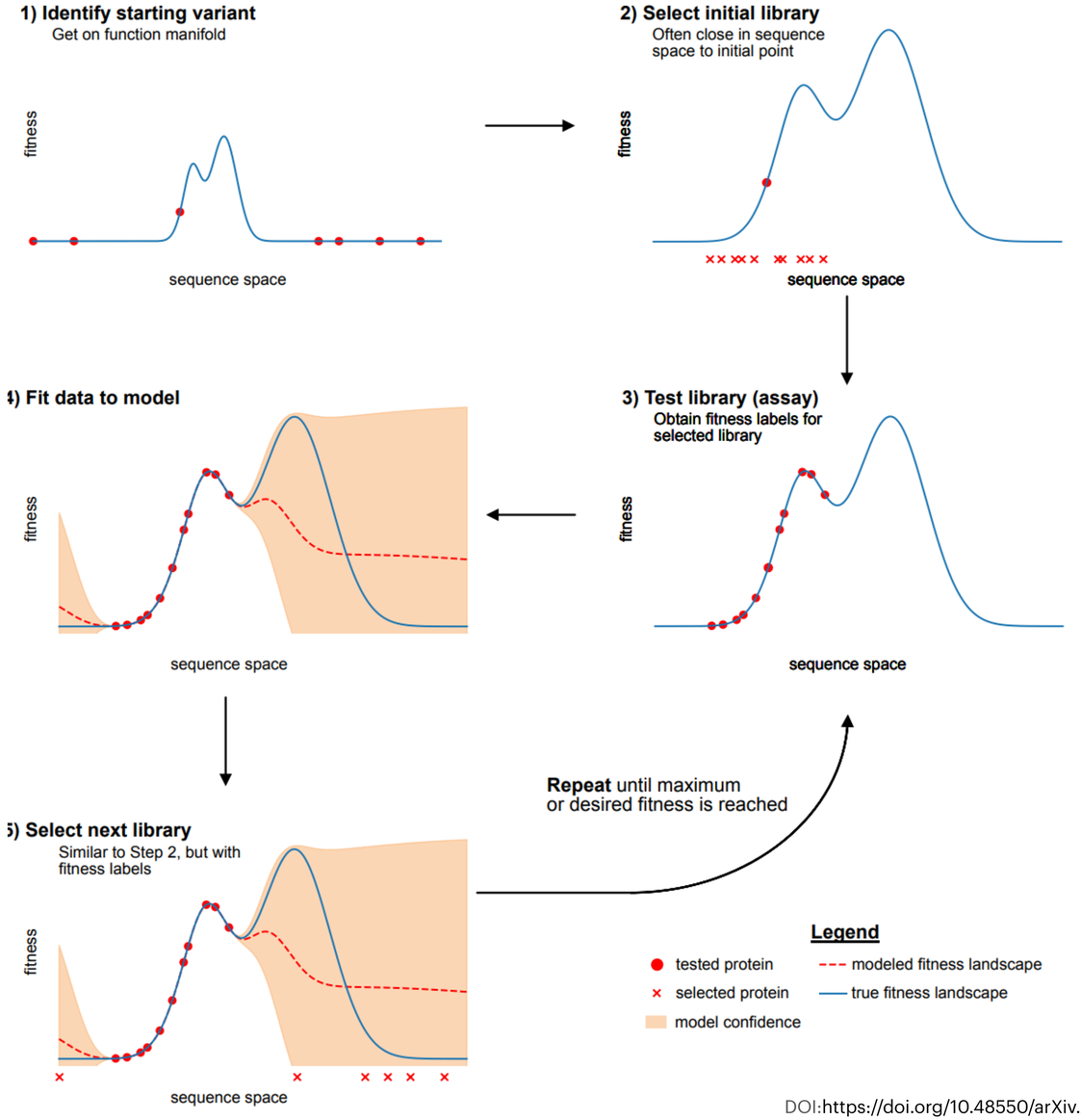
✅ Explore mutation space that is physically hard to cover

🧠We don’t expect the model to replace MD entirely in this case, but it can prioritise promising candidates for detailed simulation (e.g., MD in explicit solvent).

Data-Driven Estimation of Binding Energy

Aim: to train a model that learns the following function

$$f(sequence) \rightarrow binding\ energy$$



🧠 This model allows **fast screening of hundreds** of mutations, helping us identify most promising ones for enhanced simulation.

DOI:https://doi.org/10.48550/arXiv.2305.16634

What is Encoding?

Encoding - translating biological sequences into machine-readable vectors.

Encoding defines how well the model can “understand” sequence differences.

Requirements for Good Encoding:

- **Distinguishability** → encoded elements must be clearly separable (e.g., different amino acids must look different for the model)
- **Preservability** → biologically or chemically related amino acids should remain similar in encoded form.

Overview of Encoding Methods

Fixed Encodings

- **One-hot encoding:**
  - → Simplest method, no biological knowledge
  - → Each amino acid = a vector with one “1”, rest “0”
- **VHSE8:**
  - → Encodes amino acids by 8 physicochemical properties
- **BLOSUM:**
  - → Based on evolutionary information
  - → Reflects biochemical similarity

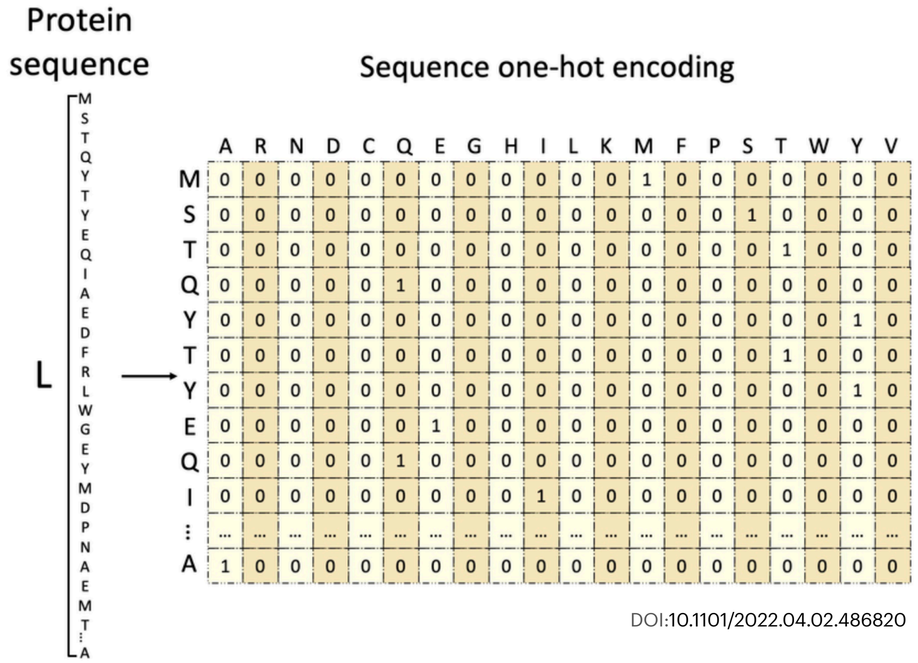
Learned Embeddings

- **ProtVec:**
  - → Trained on amino acid triplets
  - → Learns context-aware features, similar to word2vec in NLP
- **ESM:**
  - → Large transformer models trained on millions of sequences
  - → Embedded structural, statistical, and evolutionary info
  - → Very rich, but computational intensive

One-hot Encoding

💡 **Idea:** each amino acid is represented by a binary vector of length 20 (canonical amino acids), with a single 1 indicating the amino acid identity.

🔢 **Sequence representation:** a sequence of length $L \rightarrow$ matrix of size $L \times 20$.



DOI:10.1101/2022.04.02.486820

One-hot Encoding

✅ Advantages

- Simple and intuitive
- Easy and fast to implement
- No prior biological knowledge is required

❌ Limitations

- Sparse representation
- High dimensionality → inefficient memory usage
- No biological meaning
- Inefficient for long sequences

VHSE8 Encoding

Vector of Hydrophobic, Steric, and Electronic properties

💡**Idea:** each amino acid is represented as an 8-dim vector, derived from PCA on its physicochemical properties:

- 18 hydrophobic,
- 17 steric,
- 15 electronic features per amino acid.

After PCA:

- 2 PCs for hydrophobicity,
- 2 PCs for steric properties,
- 4 PCs for electronic properties.

VHSE8 Encoding

🔢 **Sequence representation:** a sequence of length L → matrix of size $L \times 8$.

AA	$VHSE_1$	$VHSE_2$	$VHSE_3$	$VHSE_4$	$VHSE_5$	$VHSE_6$	$VHSE_7$	$VHSE_8$
Ala A	0.15	−1.11	−1.35	−0.92	0.02	−0.91	0.36	−0.48
Arg R	−1.47	1.45	1.24	1.27	1.55	1.47	1.30	0.83
Asn N	−0.99	0.00	−0.37	0.69	−0.55	0.85	0.73	−0.80
Asp D	−1.15	0.67	−0.41	−0.01	−2.68	1.31	0.03	0.56
Cys C	0.18	−1.67	−0.46	−0.21	0.00	1.20	−1.61	−0.19
Gln Q	−0.96	0.12	0.18	0.16	0.09	0.42	−0.20	−0.41
Glu E	−1.18	0.40	0.10	0.36	−2.16	−0.17	0.91	0.02
Gly G	−0.20	−1.53	−2.63	2.28	−0.53	−1.18	2.01	−1.34
His H	−0.43	−0.25	0.37	0.19	0.51	1.28	0.93	0.65
Ile I	1.27	−0.14	0.30	−1.80	0.30	−1.61	−0.16	−0.13
Leu L	1.36	0.07	0.26	−0.80	0.22	−1.37	0.08	−0.62
Lys K	−1.17	0.70	0.70	0.80	1.64	0.67	1.63	0.13
Met M	1.01	−0.53	0.43	0.00	0.23	0.10	−0.86	−0.68
Phe F	1.52	0.61	0.96	−0.16	0.25	0.28	−1.33	−0.20
Pro P	0.22	−0.17	−0.50	0.05	−0.01	−1.34	−0.19	3.56
Ser S	−0.67	−0.86	−1.07	−0.41	−0.32	0.27	−0.64	0.11
Thr T	−0.34	−0.51	−0.55	−1.06	0.01	−0.01	−0.79	0.39
Trp W	1.50	2.06	1.79	0.75	0.75	−0.13	−1.06	−0.85
Tyr Y	0.61	1.60	1.17	0.73	0.53	0.25	−0.96	−0.52
Val V	0.76	−0.92	0.17	−1.91	0.22	−1.40	−0.24	−0.03

doi:10.1371/journal.pone.0074506.t001

VHSE8 Encoding

✅ Advantages

- Captures physicochemical similarity between amino acids
- More compact than one-hot encoding

❌ Limitations

- Fixed vectors: cannot adapt to specific context
- Limited scope: ignores evolutionary and structural context

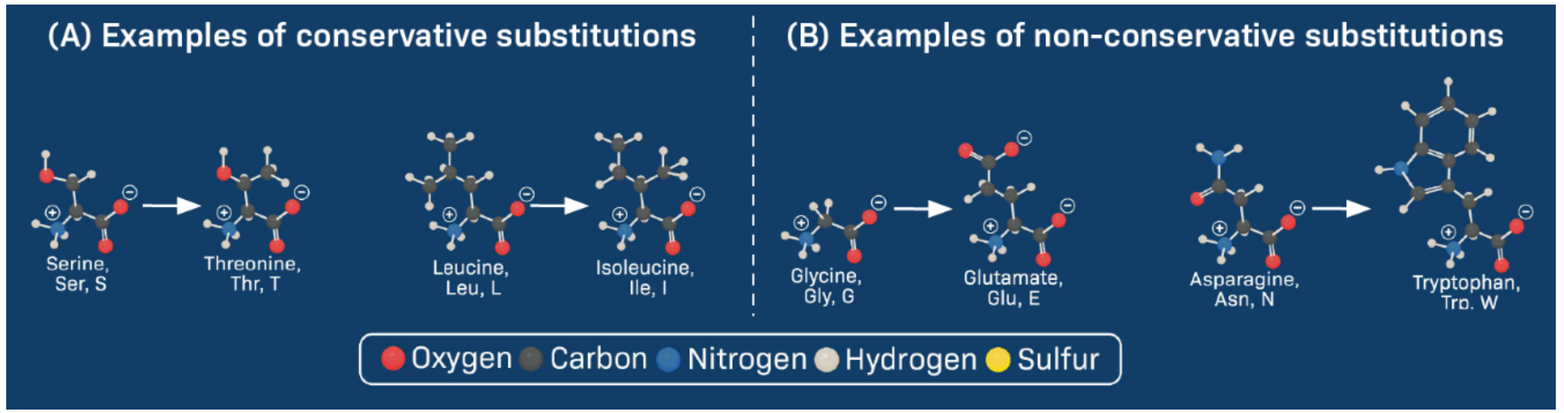
BLOSUM Encoding

BLOck SUbstitution Matrix

💡**Idea:** encodes amino acids based on evolutionary substitution scores.

Each amino acid is represented as a 20-dim vector, which reflects how likely it is to be substituted by other amino acids.

- Conservative substitutions (e.g., Lys↔Arg, Asp↔Glu, Ser↔Thr) are frequent and generally preserve protein structure/function.
- Higher score = more likely = more biologically similar.



BLOSUM Encoding

🔢 **Sequence representation:** a sequence of length $L \rightarrow$ matrix of size $L \times 20$.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

BLOSUM Encoding

✅ Advantages

- Captures evolutionary similarity
- Biologically meaningful
- Widely used in bioinformatics tools

❌ Limitations

- Fixed vectors: cannot adapt to specific context
- Limited scope: ignores physicochemical properties and structural context

Learned Protein Embeddings

Why learn instead of encode?

Traditional encodings:

→ Fixed, hand-designed, limited context

Learned embeddings:

→ Vectors learned from large protein datasets

→Captures sequence patterns, biochemical meaning, and similarities

📚 **Inspired by NLP models:** just like words in a sentence, amino acids have context and co-occurence patterns.

ProtVec Embedding

💡**Idea:** treats a protein sequence as a sentence, and overlapping 3-mers as words. Inspired by Word2Vec model from NLP.

- Split protein sequence into overlapping 3-mers.

  Example:

Original Sequence

$$^{(1)}\overrightarrow{M}{}^{(2)}\overrightarrow{A}{}^{(3)}\overrightarrow{F}SAEDVLKEYDRRRRMEAL..$$

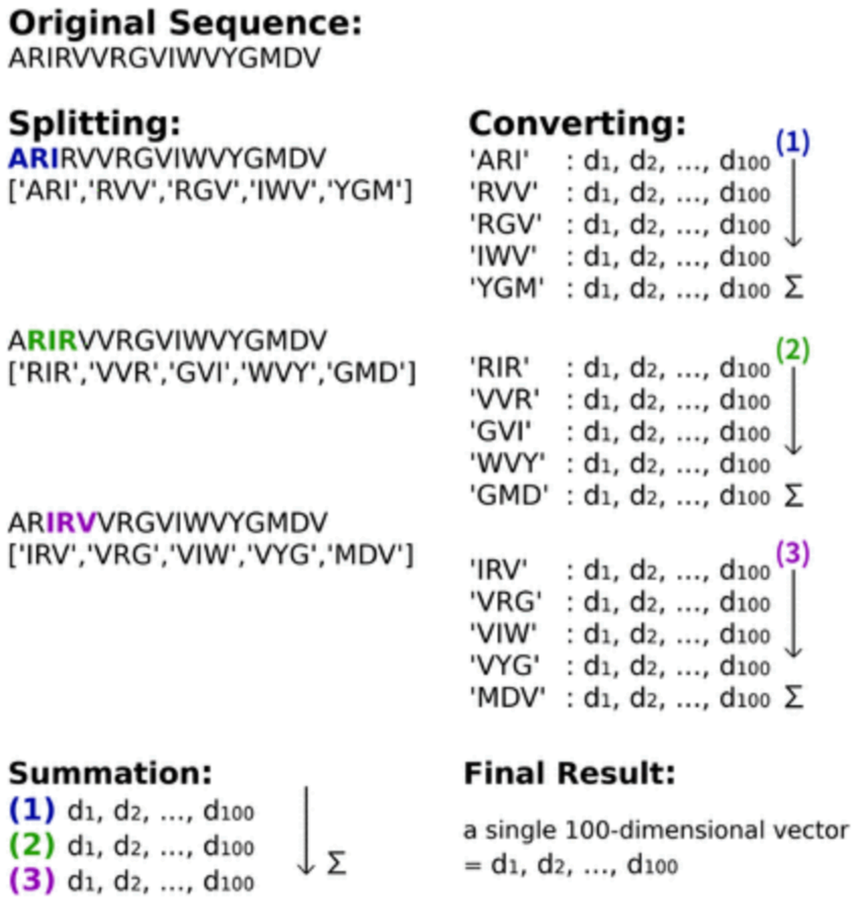
Splittings

$$\begin{cases} 1) & \text{MAF, SAE, DVL, KEY, DRR, RRM, ..} \\ 2) & \text{AFS, AED, VLK, EYD, RRR, RME, ..} \\ 3) & \text{FSA ,EDV, LKE, YDR, RRR, MEA, ..} \end{cases}$$

- Each 3-mer is mapped to a 100-dim vector based on its context.
- Embeddings of 3-mers that appear in similar context become similar.

ProtVec Embedding

🔢 **Sequence representation:** a sequence of length L → matrix of size $(L-2) \times 100$ → vector of length 100 (after mean pooling)



DOI:https://doi.org/10.1101/2021.08.02.454701

ProtVec Embedding

✅ Advantages

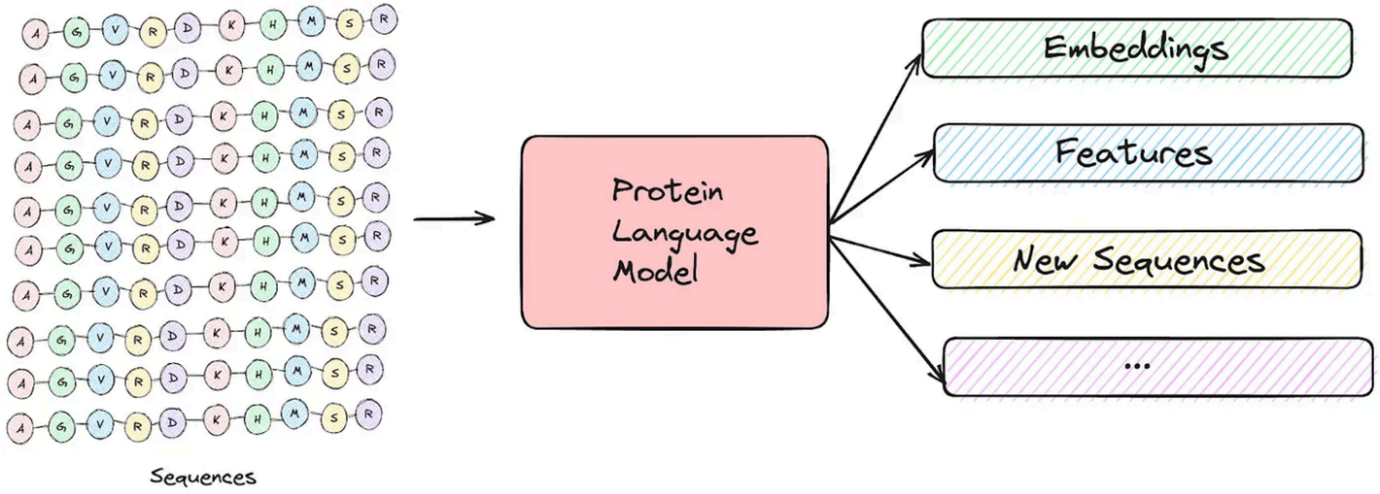
- Fast: can process thousands of sequences per seconds
- Easy to implement
- Useful for quick screening

❌ Limitations

- Not contextual: same 3-mer always has the same vector
- Ignores context, long-range dependencies, position-specific effects

Protein Language Models

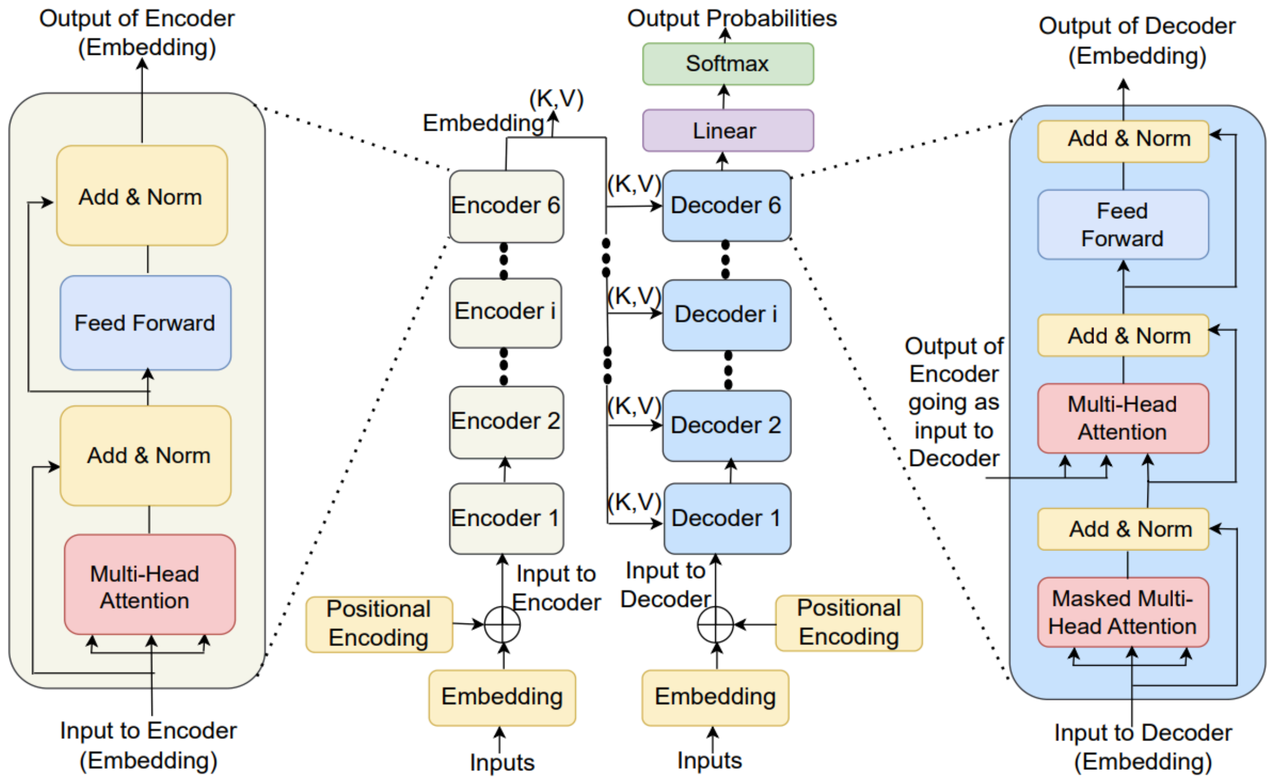
Model	# Parameters	Training Objective	Released By	Architecture
ProtTrans (T5-xxl)	11B	Masked Language Modelling	Technical University of Munich	Encoder
ProteinBERT	16M	Bidirectional Language Modelling, GO annotation prediction	The Hebrew University of Jerusalem	Encoder
ProGen2	6.4B	Next-Token Prediction	Salesforce	Decoder
ProtGPT2	738M	Next-Token Prediction	University of Bayreuth	Decoder
ESM-2	15B	Masked Language Modelling	Meta AI	Encoder



Source:https://blog.ml6.eu/unlocking-the-secrets-of-life-ai-protein-models-demystified-f286b222d571

Transformers

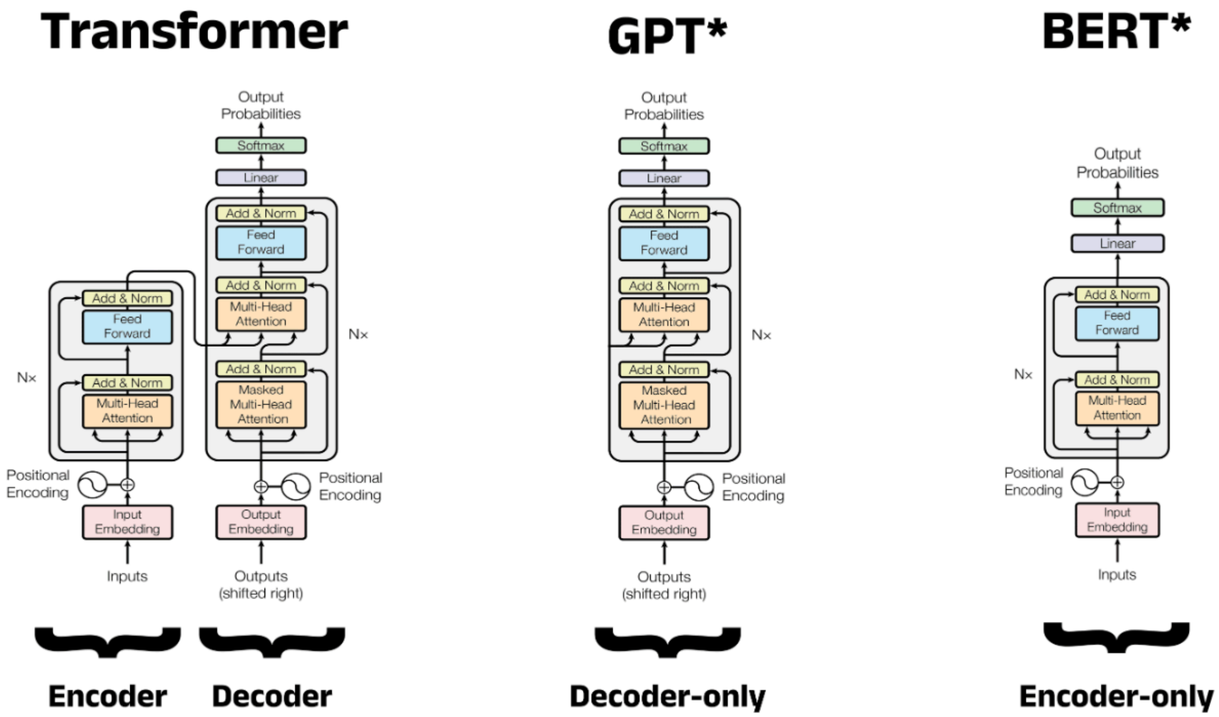
- Each amino acid = a “token” (word)
- **Self-attention** mechanism learns relationships between distant residues.
- Core components:
  - → **Encoder:** generates contextual embeddings
  - → **Decoder:** used for generative tasks



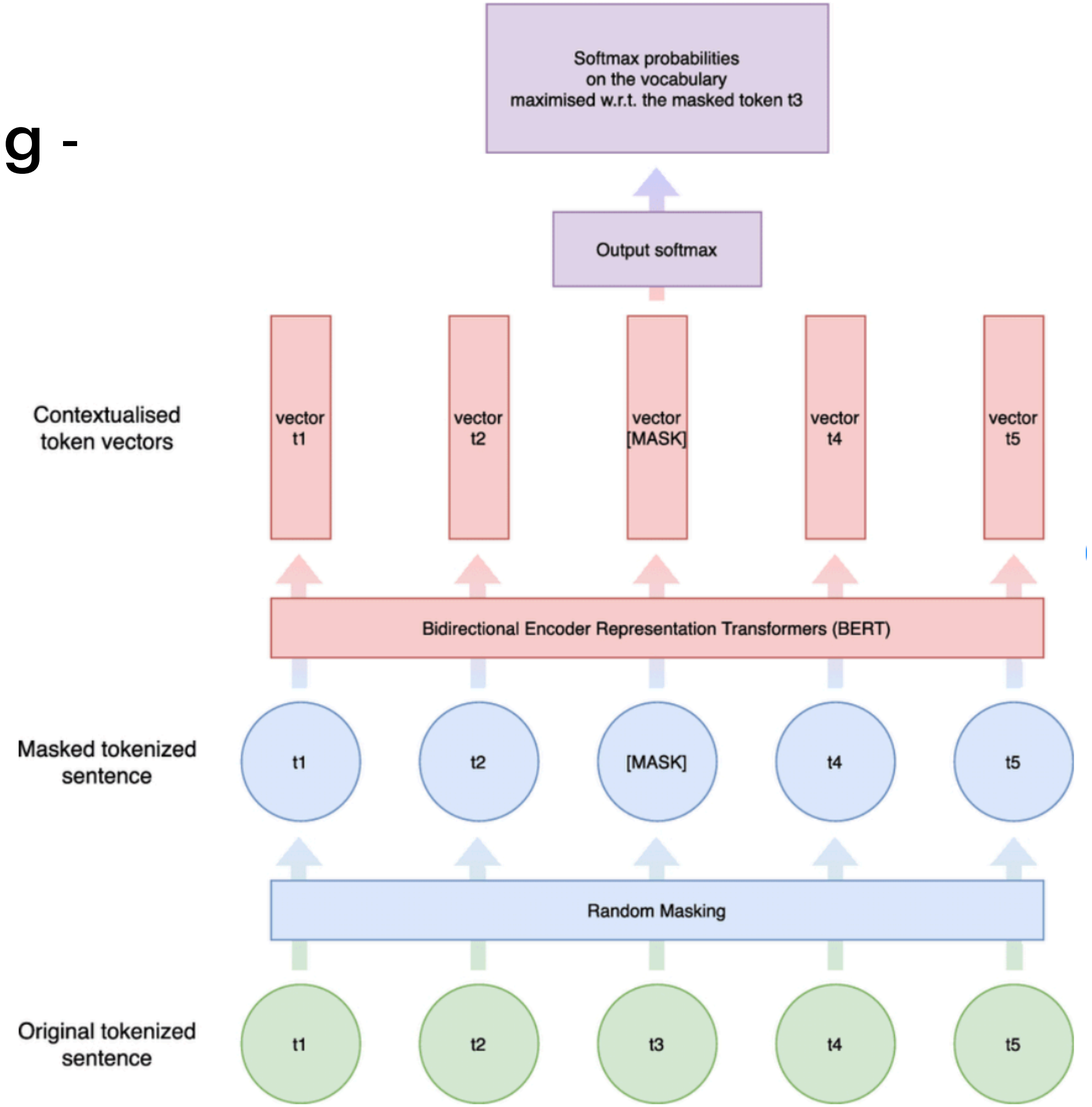
DOI:https://doi.org/10.48550/arXiv.2507.13646

Transformers

- **BERT**-style models use encoder only
- Trained with **Masked Language Modelling** - Semi-Supervised learning



*Illustrative example, exact model architecture may vary slightly



DOI: https://towardsdatascience.com/a-complete-guide-to-bert-with-code-9f87602e4a11/

DOI:10.1007/s41060-021-00242-8

Examples of models with transformer architecture

Architecture Type	Category	Model	Generative or Discriminative	Pretrained or Finetuned
Encoder-Only	Gene Ontology	OntoProtein [38]	Discriminative	Pretrained (based on ProBert [18]), further finetuned for downstream tasks
		GO Models [39]	Discriminative	Pretrained embeddings (ProtBert) and Finetuned (ProteinBERT [19])
		Zhao et al. [40]	Discriminative	Pretrained embeddings (ESM-1b [20])
		GALA [41]	Discriminative	Pretrained embeddings (ESM-1b)
	Functional and Structural Protein Cluster Identification	TooT-BERT-M [42]	Discriminative	Finetuned (ProtBERT-BFD [18])
		TooT-BERT-C [43]	Discriminative	Finetuned (ProtBERT-BFD and TooT-BERT-M)
		LMPhosSite [44]	Discriminative	Pretrained embeddings (ProtT5 [18])
		CaLMPhosKAN [45]	Discriminative	Pretrained embeddings (Codon adaptation Language Model and ProtT5)
		DeepZF [46]	Discriminative	Finetuned (ProteinBERT)
		LMNglyPred [47]	Discriminative	Pretrained embeddings (ProtT5)
		DeepLoc-2.0 [48]	Discriminative	Pretrained embeddings (ESM-2 [22], ESM-1b, ProtT5)
		Adaptor [49]	Discriminative	Neither (Uses Transformer encoder block as part of the proposed model)
		DAttProt [50]	Discriminative	Pretrained (based on Transformer Encoder and BERT-styled Masked LM), further fine-tuned for downstream tasks
		PD-BertEDL [51]	Discriminative	Pretrained embeddings (BERT-mini [52])
		MTL [53]	Discriminative	Finetuned (BERT [36])
		SPRoBERTa [54]	Discriminative	Pretrained (based on RoBERTa [55]), further finetuned for downstream tasks
		DistilProtBert [56]	Discriminative	Pretrained (based on ProtBert), further fine-tuned for downstream tasks
		Erckert et al. [57]	Discriminative	Pretrained embeddings (SeqVec [58], ProtBert, ProtT5)
		TopLapGBT [59]	Discriminative	Pretrained embeddings (MSA Transformer)
		PTSP-BERT [60]	Discriminative	Pretrained embeddings (BERT-bfd [18])
		VUS model [61]	Discriminative	Pretrained embeddings (ESM-2)
	Generating *de novo* proteins	PeTriBERT [62]	Generative	Trained from scratch (based on BERT)
	Binding	DeepHomo2.0 [63]	Discriminative	Pretrained embeddings (MSA Transformer)
		DTI-BERT [64]	Discriminative	Pretrained embeddings (ProtBert)
		TUnA [65]	Discriminative	Pretrained embeddings (ESM-2)
		LBCE-XGB [66]	Discriminative	Pretrained embeddings ([67])
		IDBindT5 [68]	Discriminative	Pretrained embeddings (ProtT5)
Decoder-Only	Generating *de novo* proteins	ProGen [23]	Generative	Trained from scratch (based on Transformer architecture), further finetuned for generating novel and functional protein sequences
		ProGen2 [24]	Generative	Pretrained (based on Transformer Decoder), further finetuned for specific tasks
		ProtGPT2 [25]	Generative	Pretrained (Based on Transformer Decoder), further finetuned for specific protein families
Encoder-decoder	Functional and Structural Protein Cluster Identification	MFTrans [69]	Discriminative	Pretrained embeddings (MSA Transformer) [28] and Transformer architecture
		ProsT5 [70]	Both	Pretrained (based on ProtT5), further fine-tuned for translating between protein sequences and structures
	Generating *de novo* proteins	xTrimoPGLM [71]	Both	Pretrained (based on General Language Model (GLM) [72]), further finetuned for downstream tasks
		CFP-GEN [73]	Generative	Trained from scratch (based on [74]), further finetuned for downstream tasks
		Regression Transformer [75]	Both	Pretrained (based on Transformer architecture), further finetuned for downstream tasks
	Binding	Trans-MoRFs [76]	Discriminative	Trained from scratch (based on Transformer architecture), further finetuned for downstream task
		AppendFormer and MergeFormer [77]	Generative	Trained from scratch (based on Transformer architecture)
		GeoDock [78]	Generative	Pretrained embeddings (ESM-2)

TABLE II: Summary of the models based on their architecture

DOI:https://doi.org/10.48550/arXiv.2507.13646

ESM-2 Embedding

🔡 **Input:**

→ Tokenised protein sequence of length L

→ Vocabulary: 20 amino acids + special tokens (MASK, PAD, BOS, EOF)

🔁 **Processing:**

→ Passed through 12-48 encoders

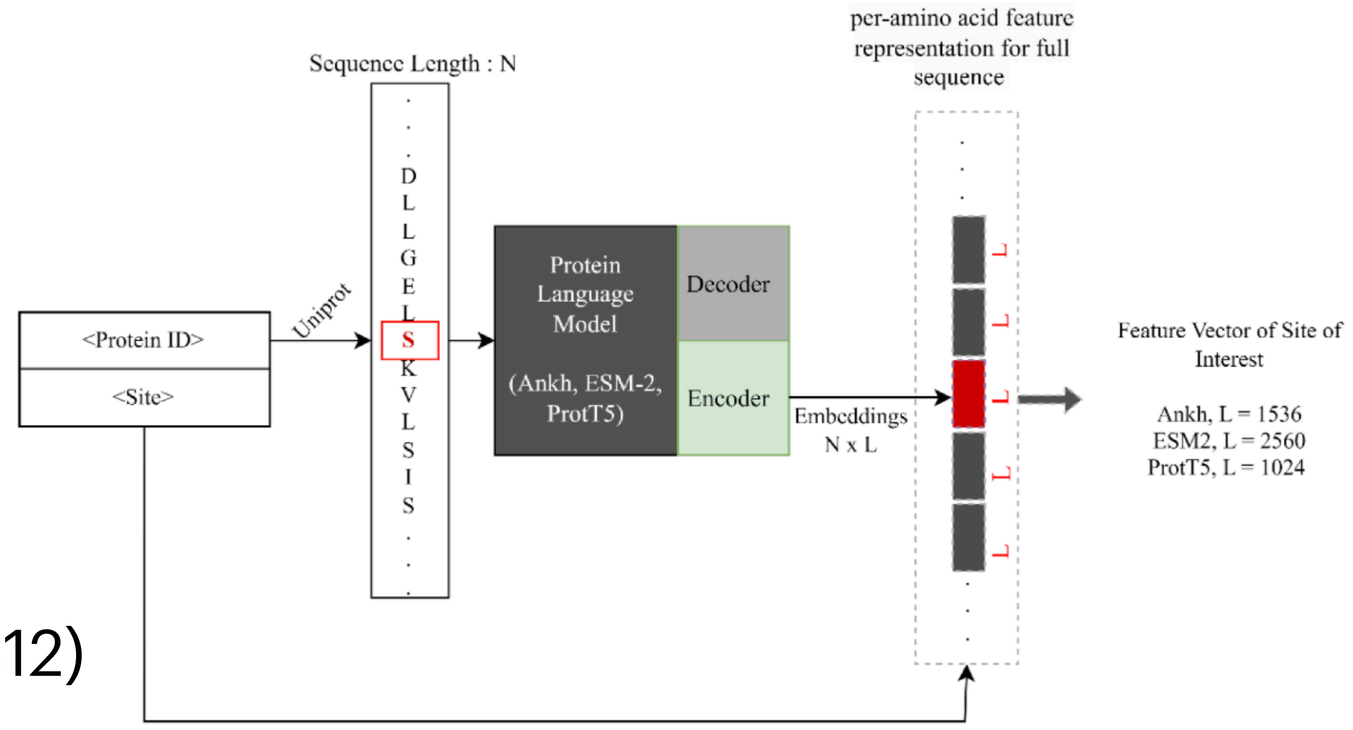
→ Self-attention allows each token to “communicate” with others

🔢 **Output:**

→ Matrix of shape $L \times D$,

where D = embedding size (e.g., 512)

→ Vector of length D (after mean pooling)



ESM-2 Embedding

✅ Advantages

- Context-aware
- Capture deep evolutionary and structural signals
- Scalable: multiple model sizes available

❌ Limitations

- High computational cost
- Limited tokenisation: standard amino acid alphabet only
- Black box: difficult to interpret what the model “understands”