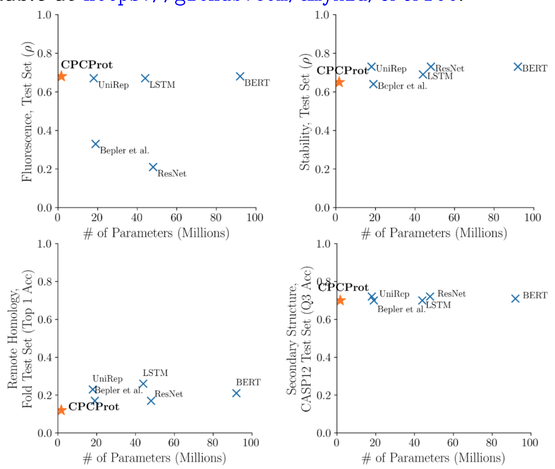
Self-Supervised Contrastive Learning of Protein Representations By Mutual Information Maximization

Amy X. Lu[1,2], Haoran Zhang[1,2], Marzyeh Ghassemi[1,2], Alan Moses[1,3]

[1]Department of Computer Science, University of Toronto [2]Vector Institute [3]Department of Cell and Systems Biology, University of Toronto

Summary

We present **CPCProt**, a protein sequence embedding model which achieves comparable results on TAPE downstream benchmark tasks with 2% to 10% less parameters. Our model is available at https://github.com/amyxlu/CPCProt.



Motivation

- Though recent works demonstrate promise, current methods take directly from large NLP language models.
- Since sequences are fundamentally vehicles for information transmission, capturing phenotypic information from protein sequences can be viewed as information transmission across the "noisy channels" of heredity and translation.

InfoNCE Loss

We adopt the contrastive InfoNCE objective [1] for proteins, which estimates the mutual information $I'_{NCE}(z_{t+k}; c_t)$:

$$\mathcal{L}_{t+k} = -\mathbb{E}\Big[\log \frac{\exp(f(z_{t+k}, c_t))}{\exp(f(z_{t+k}, c_t)) + \sum_{j=1}^{N-1} \exp(f(z'_j, c_t))}\Big], \qquad (1)$$

References

[1] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

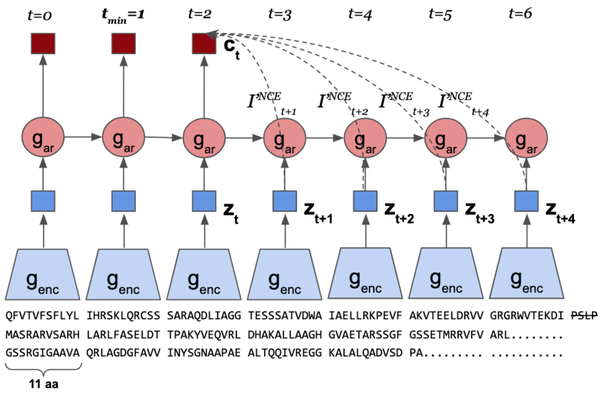
[2] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pages 9686–9698, 2019.

CPCProt: Methods

- **"Patching" Protein Sequences**: Each input x is divided into fixed-length patches. Each patch passes through g_{enc} to form z, which are concatenated and passed into g_{ar} to form c.
- **Aggregating Mutual Information Estimates:** At a given position t, we estimate the mutual information (MI) $I'_{NCE}(z_{t+k}; c_t)$ using Equation 1 for $k \in \{1, 2, \ldots, K\}$. The final loss minimized for each batch is:

$$\mathcal{L} = \frac{1}{L_z - K - t_{min}} \frac{1}{K} \sum_{t=t_{min}}^{L_z-K} \sum_{k=1}^{K} \mathcal{L}_{t+k} \qquad (2)$$

- **Negative Sampling:** In each batch of N samples, we have a single "correct" pair of $\{z_{t+k}, c_t\}$ where the $c_t = g_{ar}(z_t)$, and $N-1$ "fake" pairs where z' is drawn from other mismatched samples in-batch to create $\{z'_j, c_t\}_{j=1}^{N-1}$.



Benchmark Downstream Tasks and Datasets

For consistency with downstream benchmarks [2], we use Pfam for pretraining, and the same datasets and tasks for downstream evaluation:

- Remote Homology
  - Classify structural folds (1195 classes). Top-1 accuracy.
- Secondary Structure
  - Sequence-to-sequence task mapping positions to {helix, strand, other}. Q3 (Three-class) accuracy.
- Fluorescence
  - Deep mutational scan dataset mapping mutant GFP sequences to log-intensity. Spearman's ρ.
- Stability
  - Log-difference of the actual and predicted EC_{50} of a mutant protein. Spearman's ρ.

We use three methods for selecting pretraining hyperparameters, to avoid overfitting to benchmarks:

- Validation set performance on downstream tasks
- Accuracy on pretraining contrastive task
- Simple 1-nearest-neighbor classifier for a toy classification task

Downstream Tasks Results

Using the default MLP/CNN classification heads in TAPE [2]:

	# of Embedding Parameters	Remote Homology			Secondary Structure			Stability	Fluorescence
		Fold	Superfamily	Family	CB513	CASP12	TS115		
BERT	92M	0.21	0.34	0.88	0.73	0.71	0.77	**0.73**	**0.68**
ResNet	48M	0.17	0.31	0.77	**0.75**	**0.72**	**0.78**	**0.73**	0.21
LSTM	44M	**0.26**	**0.43**	**0.92**	**0.75**	0.70	**0.78**	0.69	0.67
Bepler et al.	19M	0.17	0.20	0.79	0.73	0.70	0.76	0.64	0.33
Unirep	18M	0.23	0.38	0.87	0.73	**0.72**	0.77	**0.73**	0.67
One Hot	0	0.09	0.08	0.39	0.69	0.68	0.72	0.19	0.14
CPCProt	1.7M	0.12	0.12	0.48	0.69	0.70	0.73	0.65	**0.68**
$\text{CPCProt}_{\text{GRU\_large}}$	8.4M	0.13	0.14	0.52	0.70	0.70	0.73	0.65	**0.68**
$\text{CPCProt}_{\text{LSTM}}$	71M	0.11	0.11	0.47	0.68	0.66	0.70	0.68	**0.68**

Using simple logistic regression (LR) and kNN downstream classifiers:

	Remote Homology					
	Fold		Superfamily		Family	
	LR	kNN	LR	kNN	LR	kNN
UniRep	0.08	0.06	0.18	0.11	0.48	0.38
BERT	**0.20**	0.11	**0.30**	**0.24**	**0.76**	**0.74**
CPCProt	0.14	**0.12**	0.13	0.10	0.50	0.51
$\text{CPCProt}_{\text{GRU\_large}}$	0.13	**0.12**	0.14	0.10	0.50	0.55
$\text{CPCProt}_{\text{LSTM}}$	0.14	0.11	0.15	0.12	0.52	0.55

	Secondary Structure		
	CB513	CASP12	TS115
	LR	LR	LR
UniRep	0.66	0.80	0.70
BERT	**0.72**	**0.82**	**0.77**
CPCProt	0.61	0.80	0.68
$\text{CPCProt}_{\text{GRU\_large}}$	0.62	0.80	0.69
$\text{CPCProt}_{\text{LSTM}}$	0.62	0.80	0.69

	Stability			
	LR		kNN	
	MSE	ρ	MSE	ρ
UniRep	**0.21**	**0.62**	0.24	**0.57**
BERT	0.36	0.39	0.23	0.49
CPCProt	0.34	0.55	**0.18**	0.51
$\text{CPCProt}_{\text{GRU\_large}}$	0.31	**0.62**	**0.18**	0.52
$\text{CPCProt}_{\text{LSTM}}$	0.22	**0.62**	0.19	0.54

	Fluorescence			
	LR		kNN	
	MSE	ρ	MSE	ρ
UniRep	1.32	0.55	**1.66**	0.37
BERT	1.15	0.52	1.75	0.46
CPCProt	1.13	0.54	1.82	0.49
$\text{CPCProt}_{\text{GRU\_large}}$	**0.81**	0.63	1.84	0.50
$\text{CPCProt}_{\text{LSTM}}$	0.85	**0.67**	1.80	**0.51**

Discussion

- In settings with limited compute resources, a parameter-efficient model such as CPCProt may be more desirable than marginal increases in accuracy.
- Using different downstream classifiers and metrics can change the ordering of embedding performances.
- Reflection on best practices for quantitative assessment for protein embeddings is needed as a community. Directly taking practices from NLP or CV (i.e. benchmarks on downstream tasks) fail to capture the greater diversity of use cases for biological sequence embeddings.