Scaling down protein language modeling with MSA Pairformer

Yo Akiyama, Zhidian Zhang, Milot Mirdita, Martin Steinegger, Sergey Ovchinnikov

LifeLU reading group

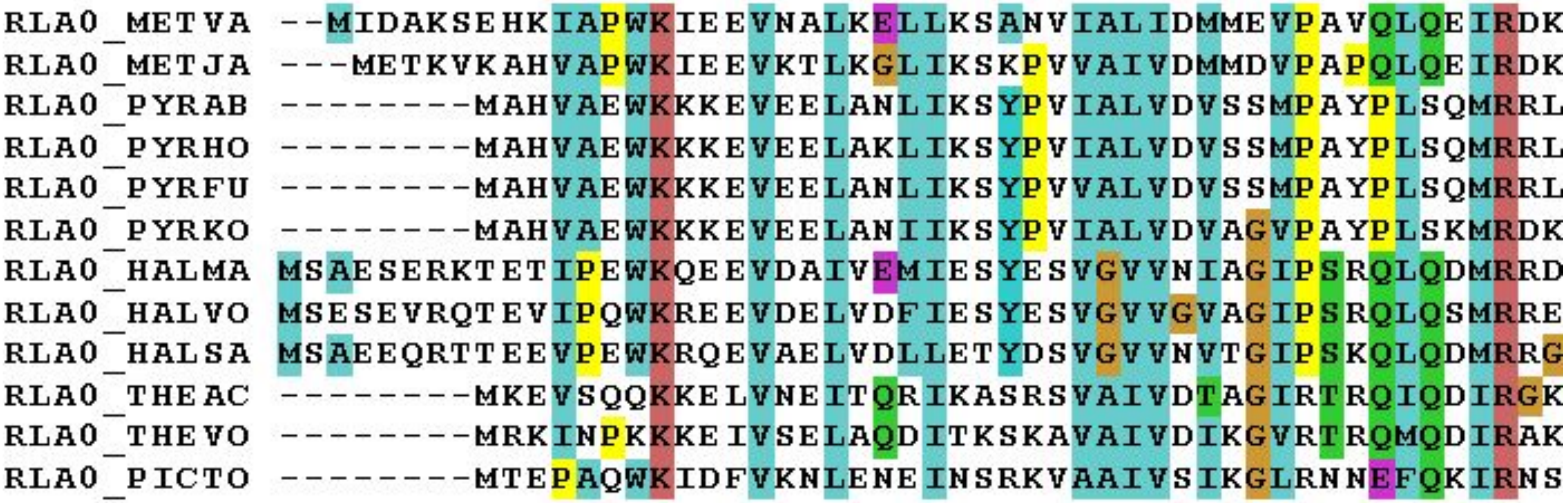
presented by Özdeniz Dolu

07.08.2025

Introduction

- We have seen performance improvements achieved via the use of protein language models (pLM for short). They also have a wide range of applicability.
- Studies show that they (single sequence PLM’s) store the evolutionary information in their weights -> Therefore, as we learn more (as our data grows), they need to scale in parameter size.
- If we extract evolutionary information from multiple sequence alignments (MSA’s for short) rather than storing it in the weights, we may achieve same tasks with smaller models.

Reminder: What is an MSA?



```
RLA0_METVA   --MIDAKSEHKIAPWKIEEVNALKELLKSANVIALIDMMEVPAVQLQEIRDK
RLA0_METJA   ---METKVKAHVAPWKIEEVKTLKGLIKSKPVVAIVDMMDVPAPQLQEIRDK
RLA0_PYRAB   --------MAHVAEWKKKEVEELANLIKSYPVIALVDVSSMPAYPLSQMRRL
RLA0_PYRHO   --------MAHVAEWKKKEVEELAKLIKSYPVIALVDVSSMPAYPLSQMRRL
RLA0_PYRFU   --------MAHVAEWKKKEVEELANLIKSYPVVALVDVSSMPAYPLSQMRRL
RLA0_PYRKO   --------MAHVAEWKKKEVEELANIIKSYPVIALVDVAGVPAYPLSKMRDK
RLA0_HALMA   MSAESERKTETIPEWKQEEVDAIVEMIESYESVGVVNIAGIPSRQLQDMRRD
RLA0_HALVO   MSESEVRQTEVIPQWKREEVDELVDFIESYESVGVVGVAGIPSRQLQSMRRE
RLA0_HALSA   MSAEEQRTTEEVPEWKRQEVAELVDLLETYDSVGVVNVTGIPSKQLQDMRRG
RLA0_THEAC   --------MKEVSQQKKELVNEITQRIKASRSVAIVDTAGIRTRQIQDIRGK
RLA0_THEVO   --------MRKINPKKKEIVSELAQDITKSKAVAIVDIKGVRTRQMQDIRAK
RLA0_PICTO   --------MTEPAQWKIDFVKNLENEINSRKVAAIVSIKGLRNNEFQKIRNS
```

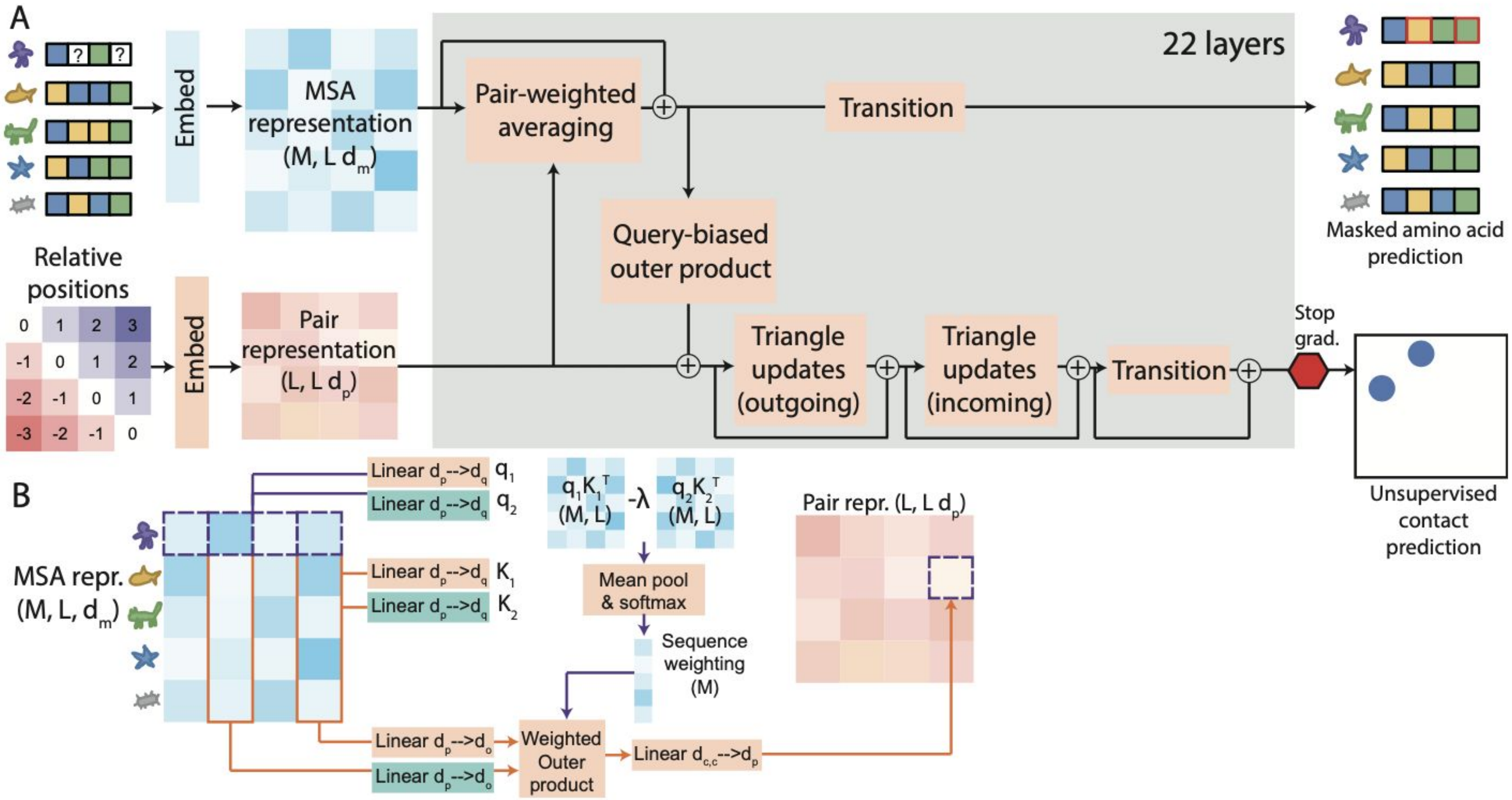
Introduction

- MSA Transformer model was competitive with ESM2-3B in long-range contact prediction and zero-shot variant effect prediction tasks, using only 118M parameters.
- SOTA MSA-based models (including MSA Transformer or AlphaFold3) suffer from a limitation. By taking the average of coevolutionary signals across all sequences in the alignment, they assume that an alignment (a family), share the same structure everywhere. However, subfamilies may show contrasting coevolutionary patterns. (single attention map across positions for all sequences)

Introduction

- Building on these observations and previous work, authors propose MSA Pairformer model.
- Uses bidirectional refinement between MSA and pair representations first introduced by AlphaFold2/3.
- Query biased outer-product operation selects most relevant sequences for the query sequence, therefore better extracts sub-family specific coevolutionary signals.
- Self-supervised learning by reconstructing a corrupted MSA.

MSA Pairformer



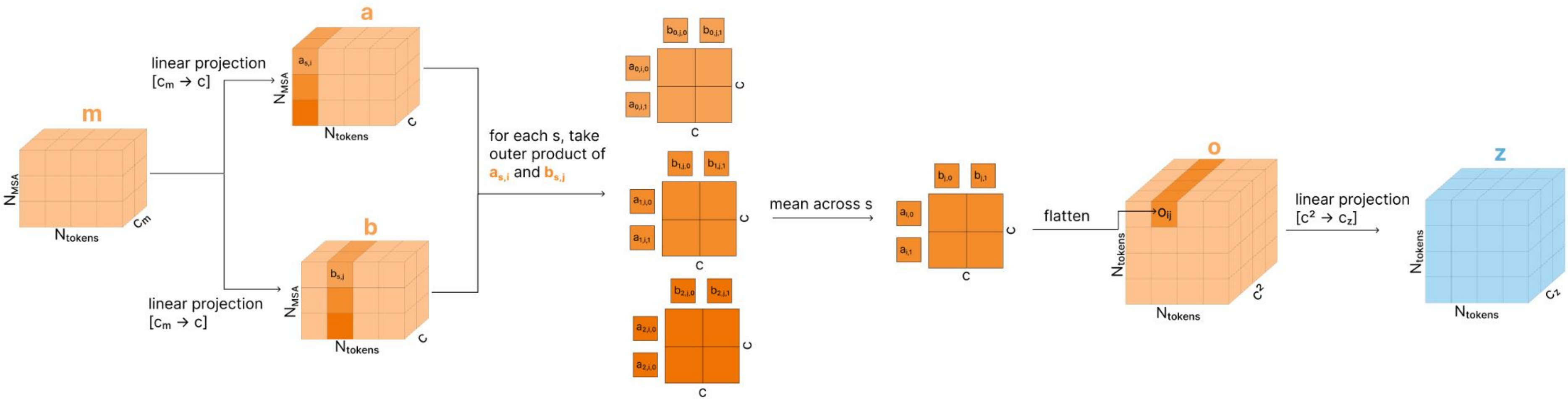
MSA Pairformer

- Has 111M parameters. Updates its representations in 22 layers (iterations).
- Input is the MSA. Rows are sequences, columns are residue positions.
- In each layer three operations:
  - Pair-weighted averaging updates MSA representation.
  - Query-biased outer product updates pair representation.
  - Triangle multiplications update pair representations (considering triplet interactions).
- Note: We have discussed a very similar architecture on AlphaFold presentations.

Recall: Outer-product mean

From the MSA module of AlphaFold. Outer-product mean updates pair representation using MSA representation.

Outer Product Mean



Query-biased outer product

Novel improvement over previous MSA-based models.

The main difference: Instead of taking the average of all outer products, use attention-weighted average (a_s).

Here, i and j are the positions (columns) in MSA and l_si and r_sj are learned projections of MSA representation for sequence s.

$$\mathrm{QueryBiasedOuterProduct}_{i,j} = \sum_{s=1}^{M} a_s * l_{si} \otimes r_{sj}$$

Pre-softmax differential attention

Based on several recent studies, differential attention has been used to calculate weights a_s. (Note: basically there are two keys K1, K2 and two queries Q1, Q2)

For sequence s, calculate a_s by taking the position-wise attention. (? not sure)

$$\mathrm{PresoftmaxDiffAttn}(X) = \mathrm{softmax}(\frac{q_1 K_1^T - \lambda q_2 K_2^T}{\sqrt{d}})$$

$$a_s = \mathrm{softmax}_s(\frac{1}{L}\sum_{i=1}^{L}\frac{q_{1,s,i}k^T_{1,s,i} - \lambda q_{2,s,i}k^T_{1,s,i}}{\sqrt{d}})$$

Training procedure

Training is divided into two phases. In both phases 270K Unitclust30 MSAs from OpenProteinSet with same training/validation splits has been used. In pre-training phase, uniform weighting is used in outer-product mean. (Basically they froze the “query-biased outer product” mechanism ?)

Also, a logistic regression classifier has been trained on 256-dim pair representation that predicts residue contacts. (Contacts are defined as less than 8A between Cb-Cb atoms). Same 20 training samples has been used as MSA Transformer and ESM-family contact predictors.

Interlude: OpenProteinSet (Ahdritz et al., 2023)

- A large-scale corpus of precomputed MSA’s.
- Aimed at training large scale structural biology models such as AF.
- Contains an updated reproduction of AlphaFold2’s unreleased training set (including structural template hits)
- Incorporates more than 16M~ MSA’s computed for each one of the Uniclust30 clusters.
- Uniclust30 is a clustering of UniprotKB at 30% pairwise sequence identity. (Mirdita et al. 2016)

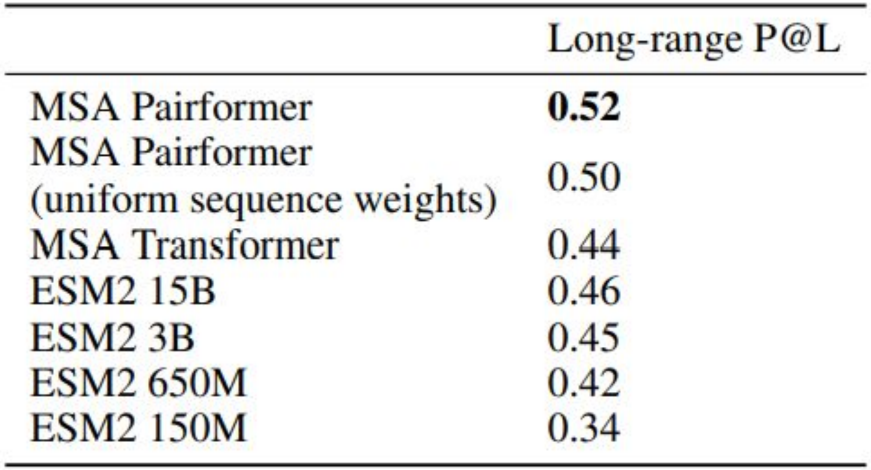
Results: Unsupervised Contact Prediction

Task: Predict long-range contacts (>= 24 residues apart).

CASP15 targets selected (49 monomeric proteins, filtered to 46)

To make the comparison more standardized

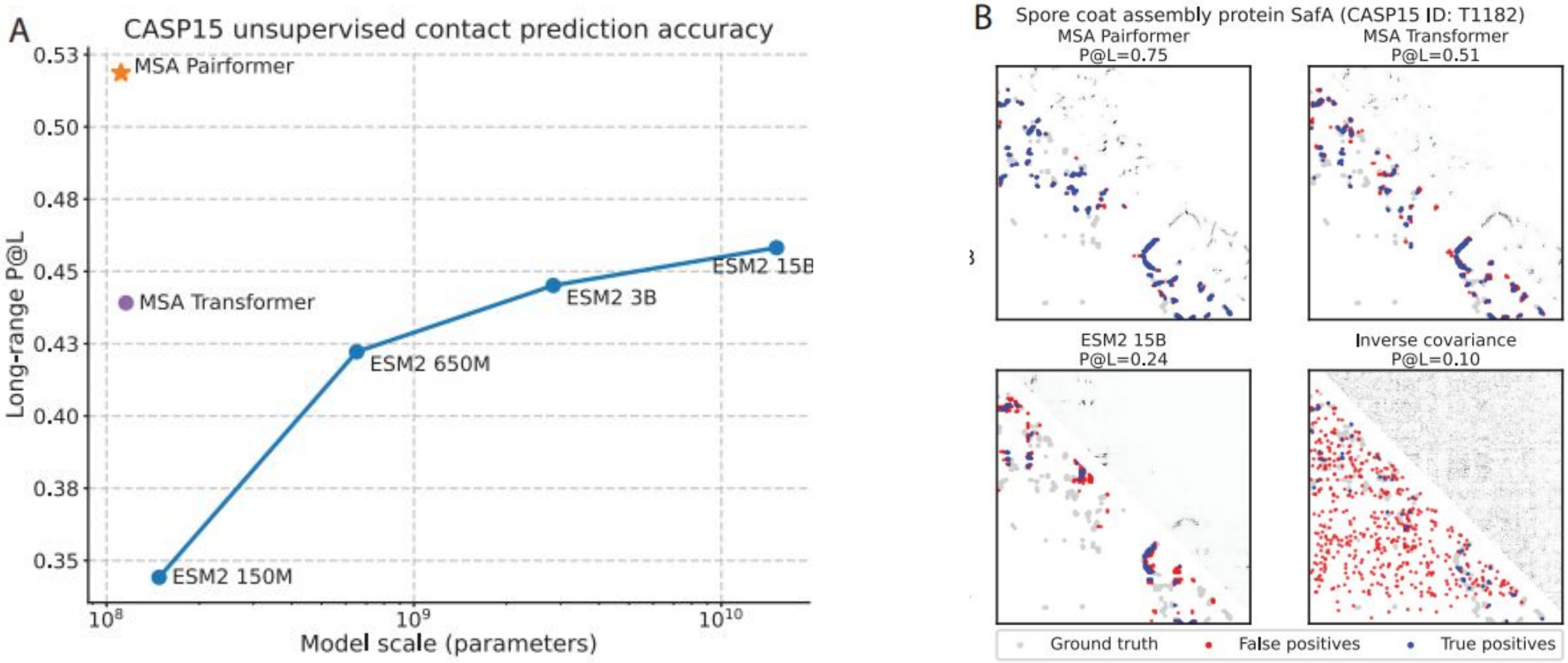
- Context window of MSA Transformer has been used for MSA depth (512).
- Since ESM2 family has been trained on UniRef, MSA’s have been generated from UniRef30.



	Long-range P@L
MSA Pairformer	**0.52**
MSA Pairformer (uniform sequence weights)	0.50
MSA Transformer	0.44
ESM2 15B	0.46
ESM2 3B	0.45
ESM2 650M	0.42
ESM2 150M	0.34

Results: Unsupervised Contact Prediction

- MSA Pairformer substantially outperformed baselines, at 100x lower parameter count than ESM2 15B.
- On the right, predictions of various models on target T1182.

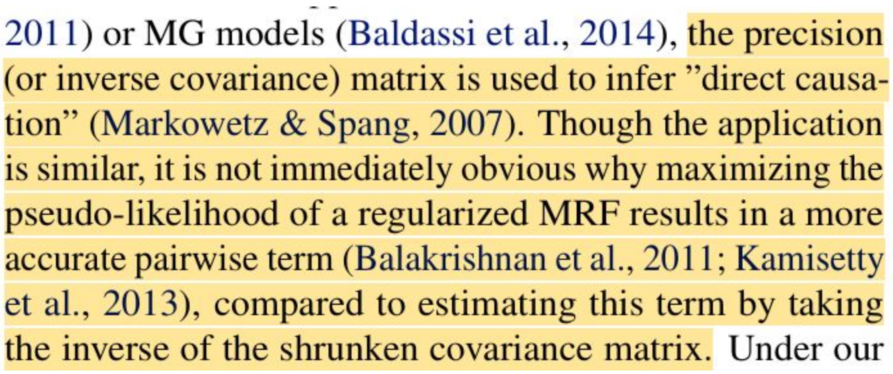


Interlude: Inverse Covariance?

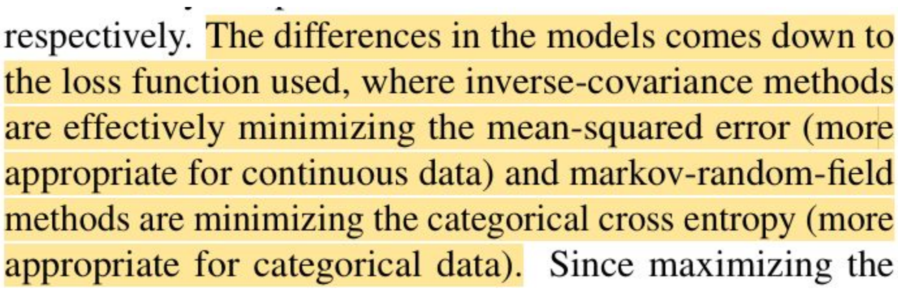
Excerpt from (Jones et al. 2012):

Thus, assuming the sample covariance matrix can in fact be inverted, the inverse covariance matrix provides information on the degree of direct coupling between pairs of sites in the given MSA. Off-diagonal elements of the inverse covariance matrix which are significantly different from zero are indicative of pairs of sites which have strong direct coupling (and are likely to be in direct physical contact in the native structure).

Excerpt from (Dauparas et al., 2019):



2011) or MG models (Baldassi et al., 2014), the precision (or inverse covariance) matrix is used to infer ”direct causation” (Markowetz & Spang, 2007). Though the application is similar, it is not immediately obvious why maximizing the pseudo-likelihood of a regularized MRF results in a more accurate pairwise term (Balakrishnan et al., 2011; Kamisetty et al., 2013), compared to estimating this term by taking the inverse of the shrunken covariance matrix. Under our



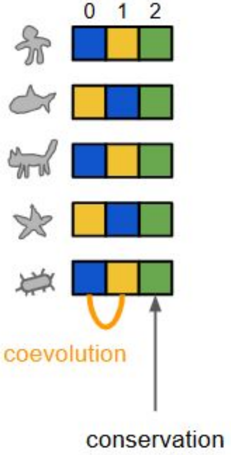
respectively. The differences in the models comes down to the loss function used, where inverse-covariance methods are effectively minimizing the mean-squared error (more appropriate for continuous data) and markov-random-field methods are minimizing the categorical cross entropy (more appropriate for categorical data). Since maximizing the

Interlude: Inverse Covariance?

Taken from a presentation by Sergey Ovchinnikov (one of the authors):

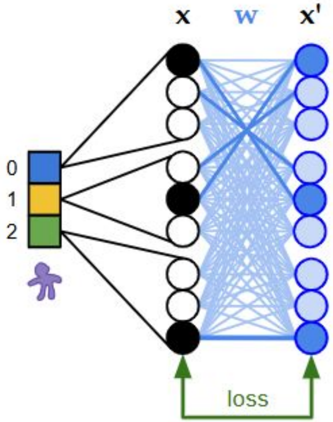
Direct Coupling Analysis

(analytical solution: **inverse** covariance)



Let's pretend there are:

- 3 positions
- 3 characters: 🟦 🟨 🟩



[w]eights = coevolution

Dauparas, J., Wang, H., Swartz, A., Koo, P., Nitzan, M. and Ovchinnikov, S., 2019. Unified framework for modeling multivariate distributions in biological sequences. *arXiv*

Morcos, F.,... Weigt, M., 2011. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *PNAS*

$\mathbf{x} = \text{msa - msa.mean}(0)$
$\mathbf{x'} = \mathbf{x}@\mathbf{w}$

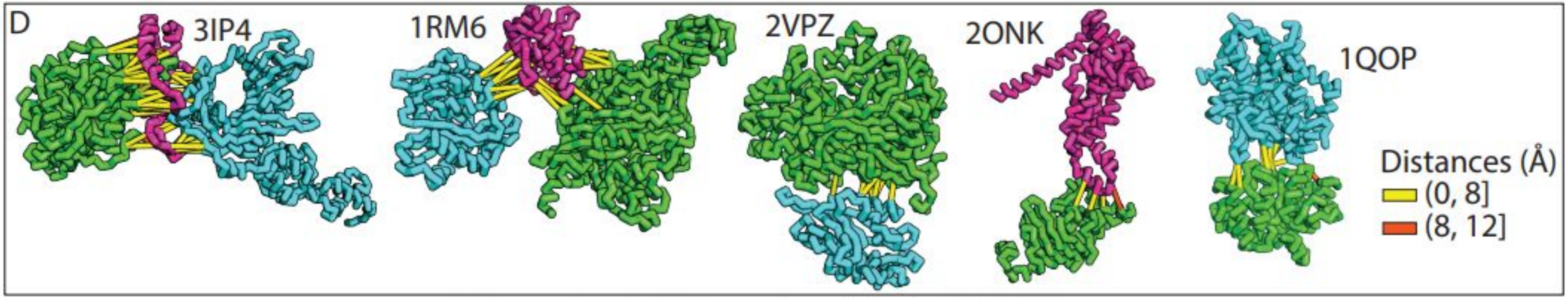
$\text{loss} = \text{MSE}(\mathbf{x'}, \mathbf{x})^2 - 2*\text{Trace}(\mathbf{w}) + \lambda*\text{L2}(\mathbf{w})$
$\mathbf{w} = (1-\lambda)*\text{inv}(\text{cov}(\mathbf{x}) + \lambda\text{I})$

Results: Protein-Protein Interaction Prediction

Task: Predict contact interfaces in protein complexes.

Evaluated on 25 paired MSA’s of evolutionarily conserved protein complexes from (Ovchinnikov et al., 2014) involving 31 interacting proteins.

Precision of the top K predicted contacts (P@K), where K equals the total number of interface contacts.



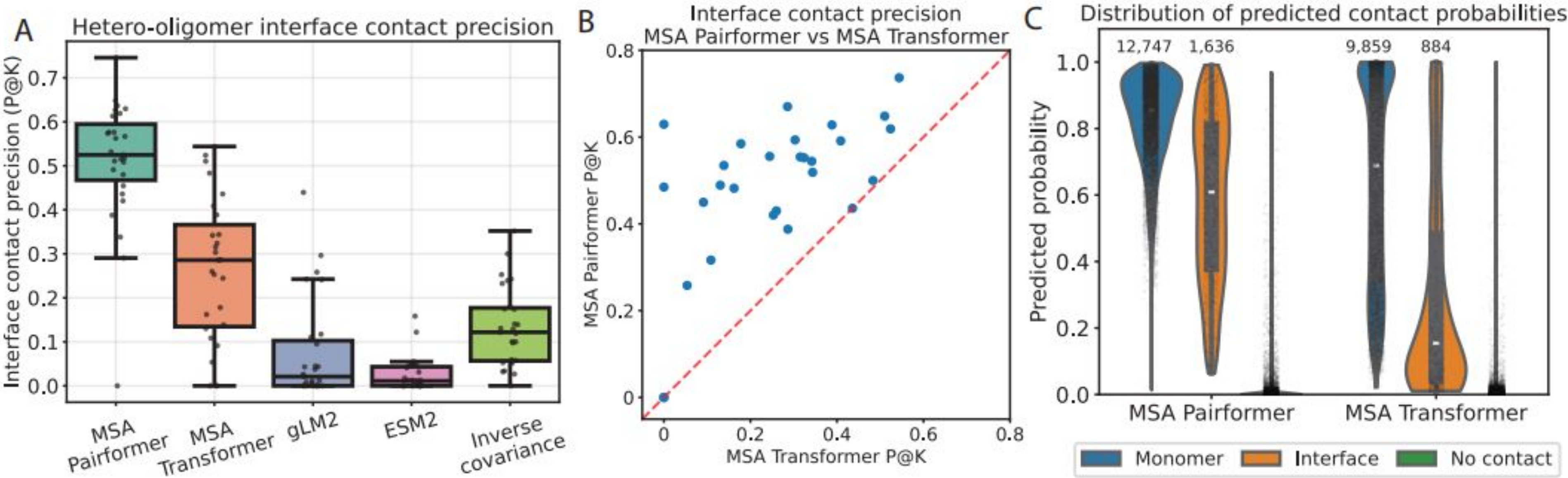
ABC transporter ModBC in complex with ModA (PDB: 2ONK)

Results: Protein-Protein Interaction Prediction

A) MSA Pairformer outperforms other models with $p <= 0.05$

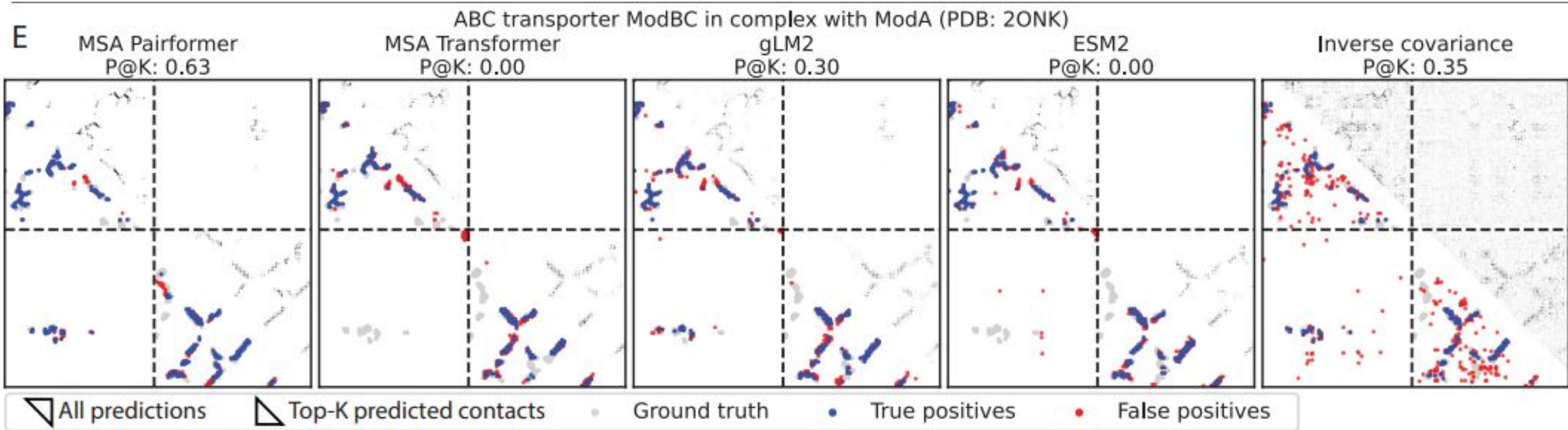
B) MSA Pairformer consistently outperforms MSA Transformer on each individual sample.

C) MSA Transformer has very low confidence on interface predictions.



Results: Protein-Protein Interaction Prediction

- Single sequence models significantly underperformed in this task.
- One interesting example where MSA Transformer completely misses contact interface although there is a strong coevolutionary signal as evidenced by inverse covariance.

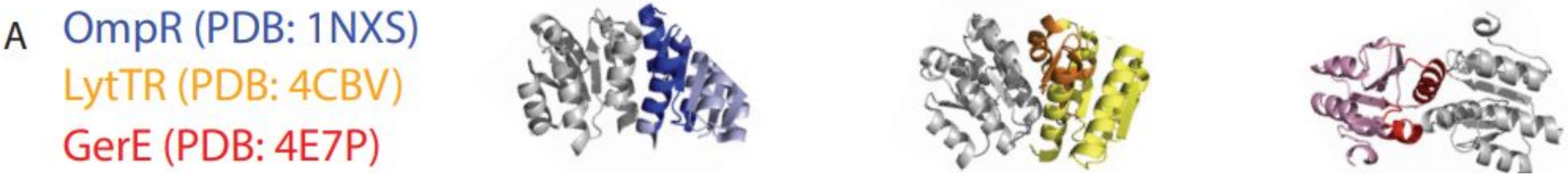


Results: Extracting Subfamily-specific Structural Properties

Purpose: Further validate if query-biased outer product really capture sub-family specific properties.

Case: **Bacterial response regulator protein family**, widely used as a case study for sub-family specific co-evolutionary signals.

Representatives for three distinct subfamilies, OmpR, LytTR, and GerE with known structures. All of them are homo dimers. Their intra-chain contacts should be similar, but their inter-chain contacts are different.



Results: Extracting Subfamily-specific Structural Properties

A single MSA with 4096 sequences (2506 from GerE, 1074 from LytTR and 516 from OmpR) has been used. And different query sequences belonging to these subfamilies have been used.

For each prediction, top N predictions have been selected where N is the total number of inter-chain contacts.

Query biased outer product have had the strongest effect on OmpR specific contacts increasing the performance from 3/19 to 8/19. This makes sense as it was the most underrepresented subfamily.

A
OmpR (PDB: 1NXS)
LytTR (PDB: 4CBV)
GerE (PDB: 4E7P)

Results: Extracting Subfamily-specific Structural Properties

In the most dominant subfamily GerE, query biased outer product increased the predictions from 3/23 to 5/23.

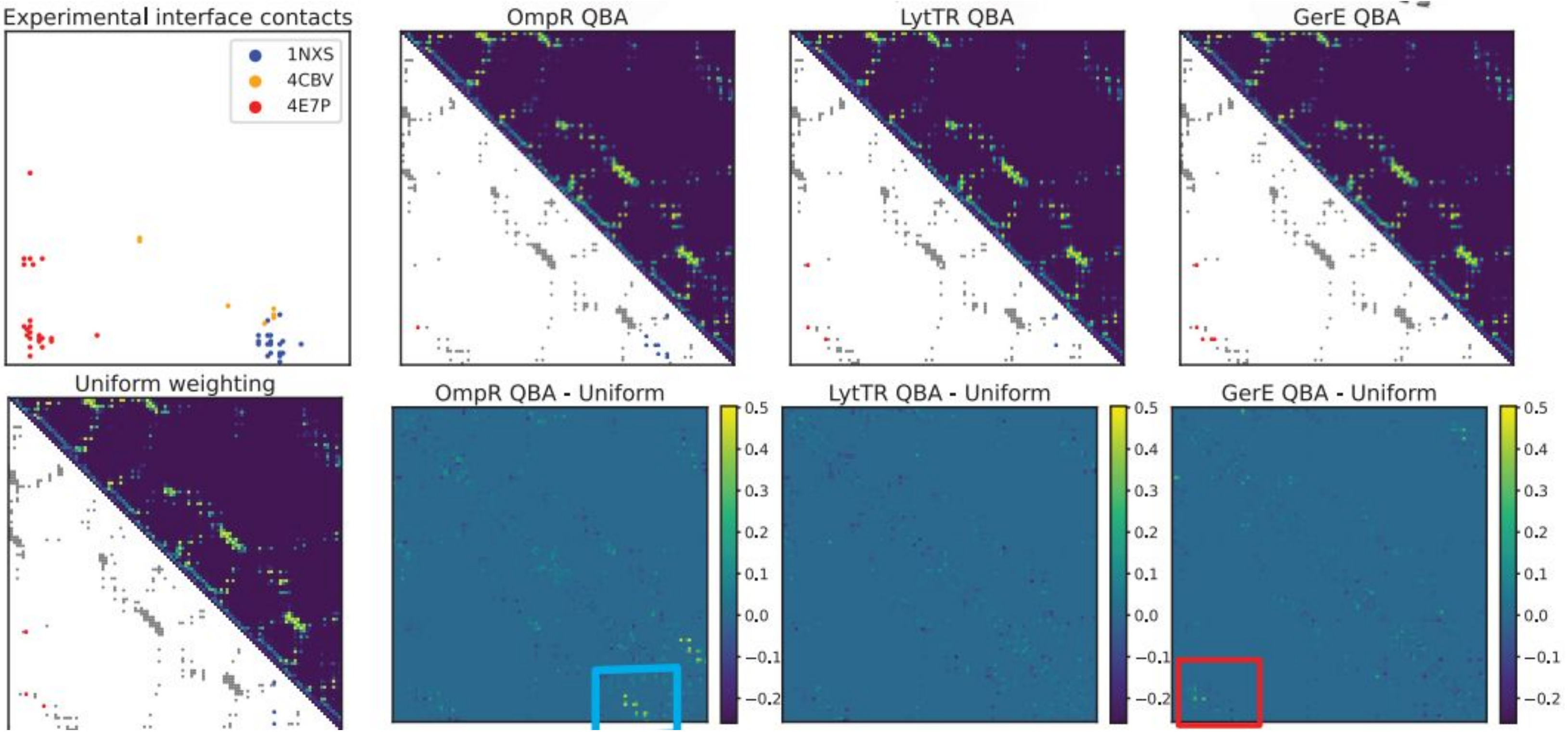
When the query changed from OmpR to GerE sequence, assignment of probability to OmpR specific contact sites decreased by 14(in percent) points.

A
OmpR (PDB: 1NXS)
LytTR (PDB: 4CBV)
GerE (PDB: 4E7P)

Results: Extracting Subfamily-specific Structural Properties

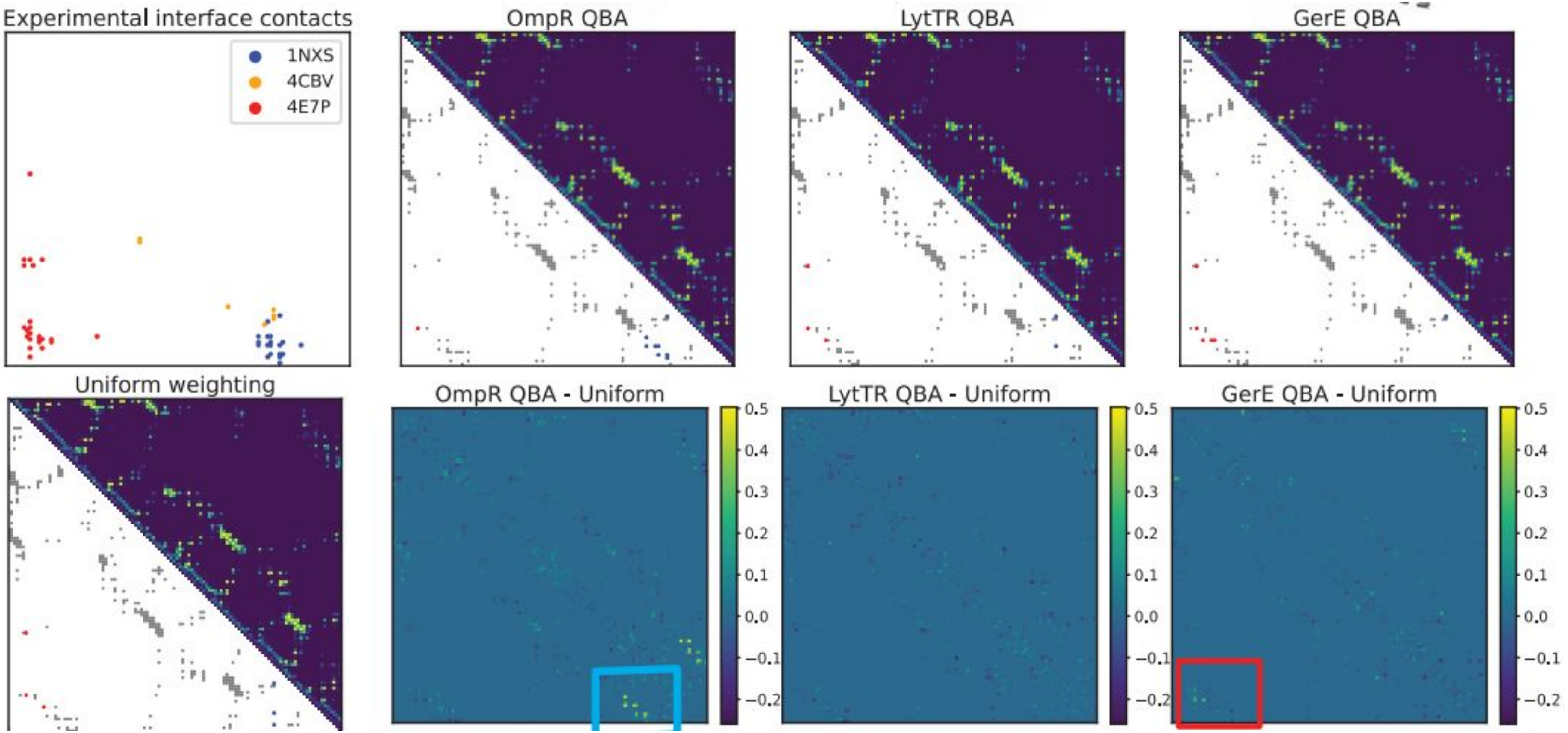
Top row: Upper triangle shows probabilities

Lower triangle shows top predictions. (They are colored but hard to see).



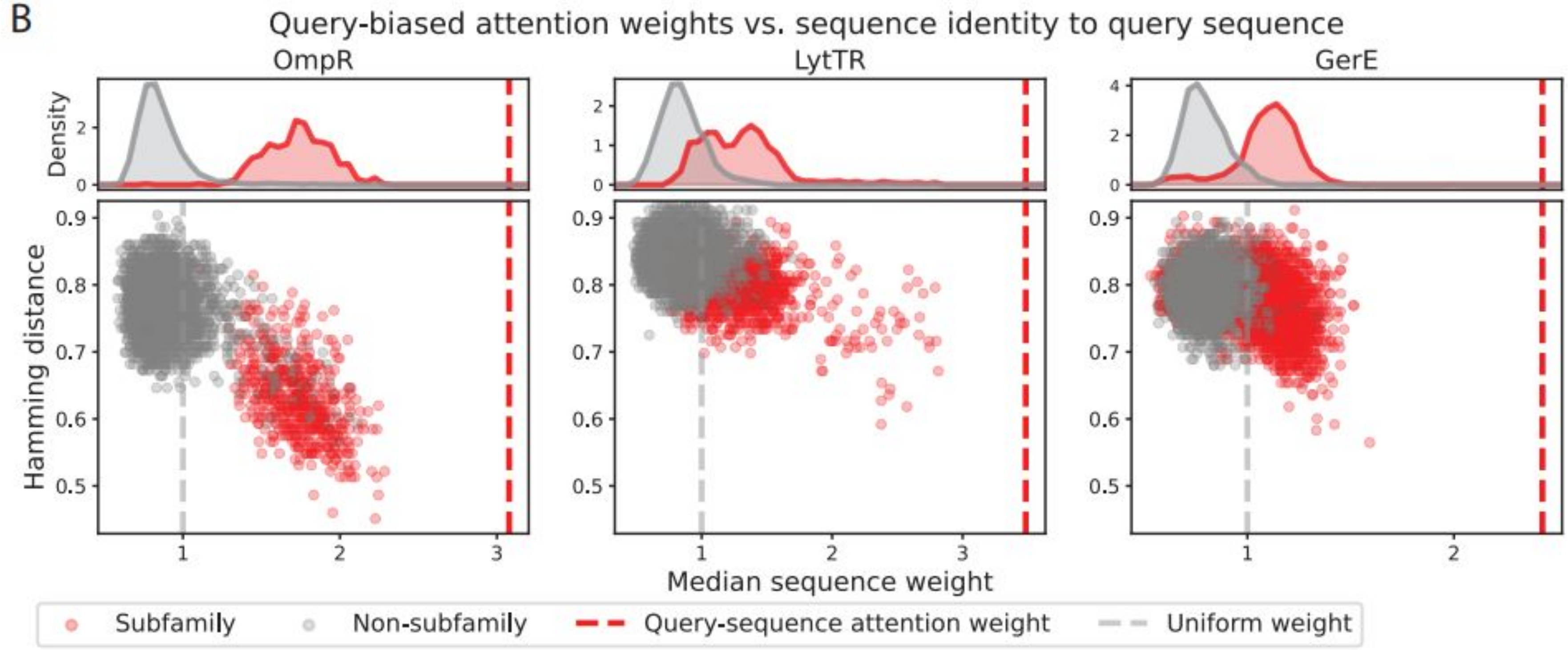
Results: Extracting Subfamily-specific Structural Properties

Bottom row: Difference in prediction probabilities when used uniform weighting instead of query biased outer product.



Results: Extracting Subfamily-specific Structural Properties

Median attention weights across all layers plotted against hamming distance to the query sequence, where query sequence belong to different subfamilies.



Results: Contact-Variant Effect Prediction Trade-off

Although scaling single-sequence pLMs increase performance in contact prediction, it has been observed that the same is not true for zero-shot variant effect prediction where performance peaks at 650M param in the case of ESM.

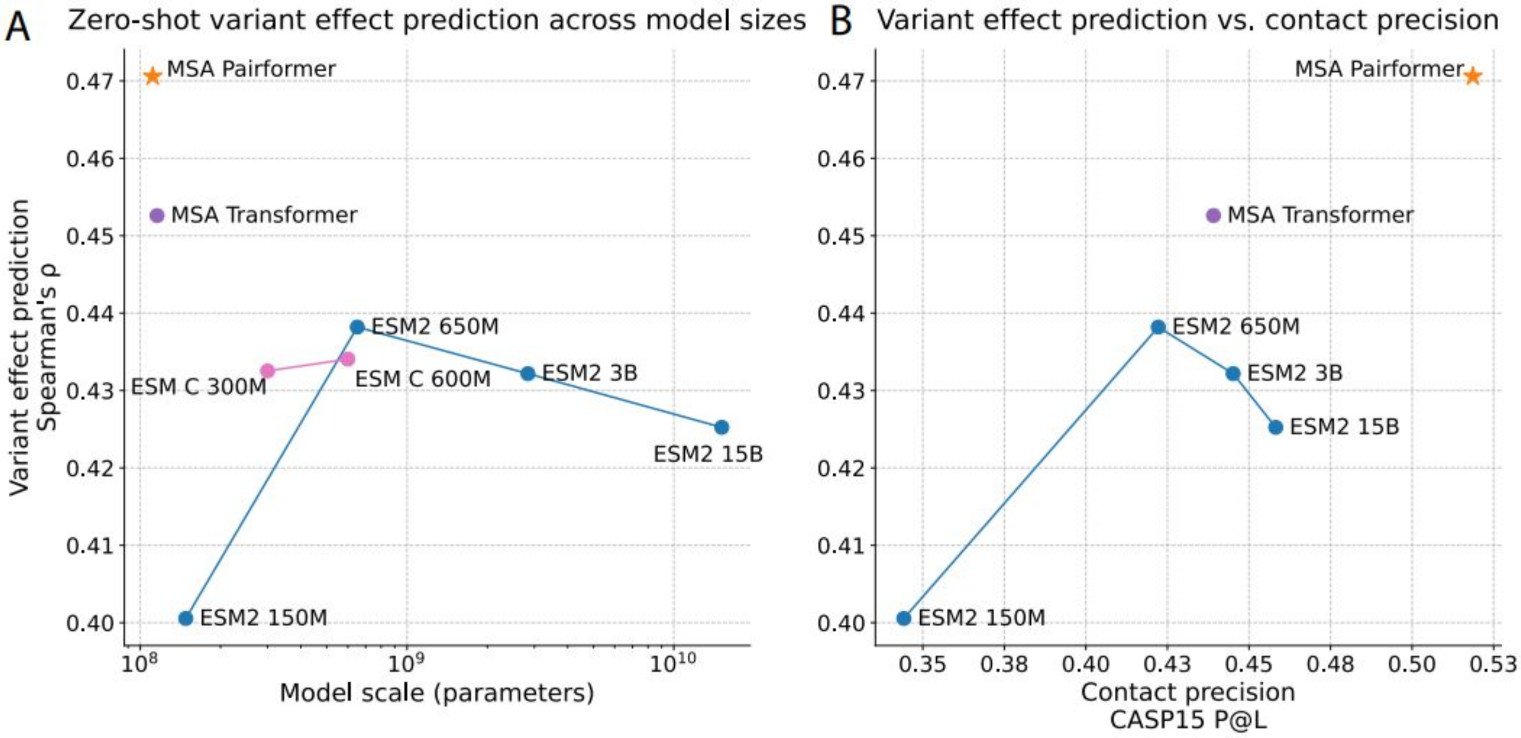
Benchmarked on the 219 ProteinGym deep mutational scan (DMS for short) substitution experiments.

Alignments provided by the ProteinGym team has been used. 4096 sequences were sampled from these where selection probability of a sequence is inversely proportional to number of other sequences it has at least 80% identity to.

Results: Contact-Variant Effect Prediction Trade-off

Both in terms of the trade-off (B figure) and in terms of performance (A figure),

MSA pairformer outperformed its competitors with a much lighter model size.



Results: MSA Pairformer pseudolikelihoods are better at distinguishing non-binding and binding PPI

A library of mutants at 4 key interface residues of antitoxin ParD3.

For toxin-antitoxin pairs, a dataset containing 9194 sequences and binding fitness scores. 252 are good binders.

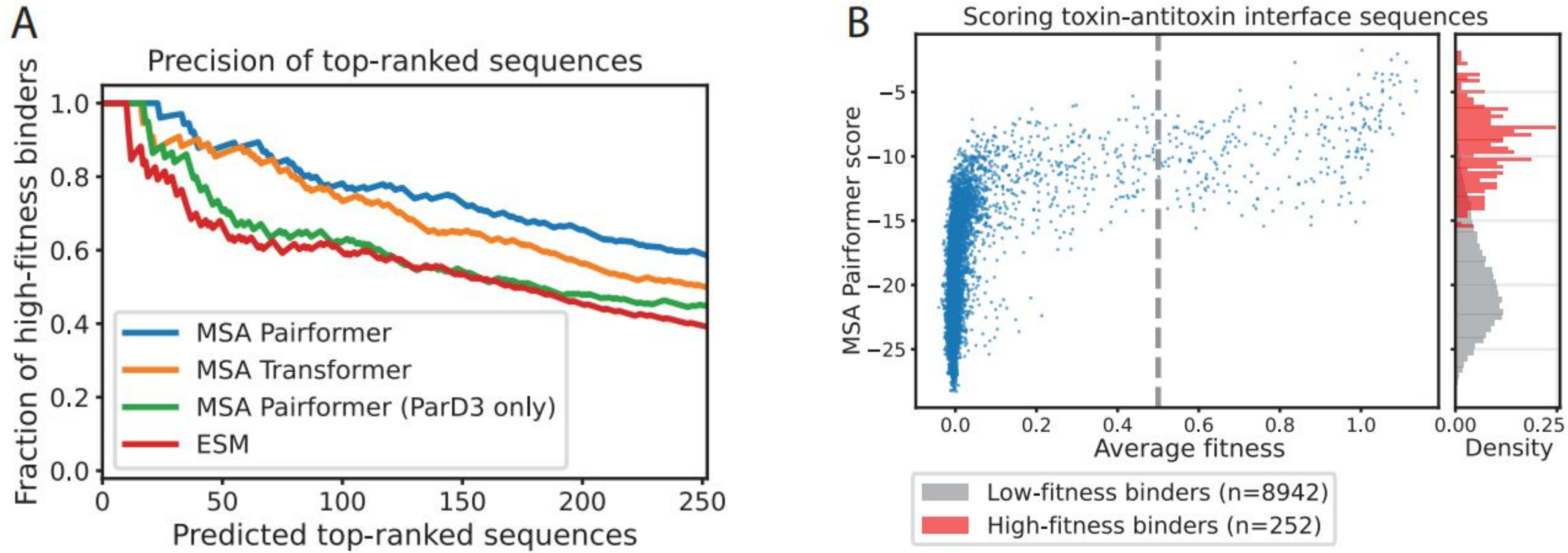
Compute pseudolikelihoods of 4 mutated positions and rank the sequences. Select the top 252 and report the precision.

(???Take the log of predicted probabilities at these 4 positions in the contact map against the toxin and sum them up???)

Results: MSA Pairformer pseudolikelihoods are better at distinguishing non-binding and binding PPI

MSA Pairformer outperforms (although it is close to MSA Transformer).

(“ParD3 only”) refers to removing the toxin(?) from the sequence (query only as a single sequence rather than PPI?)

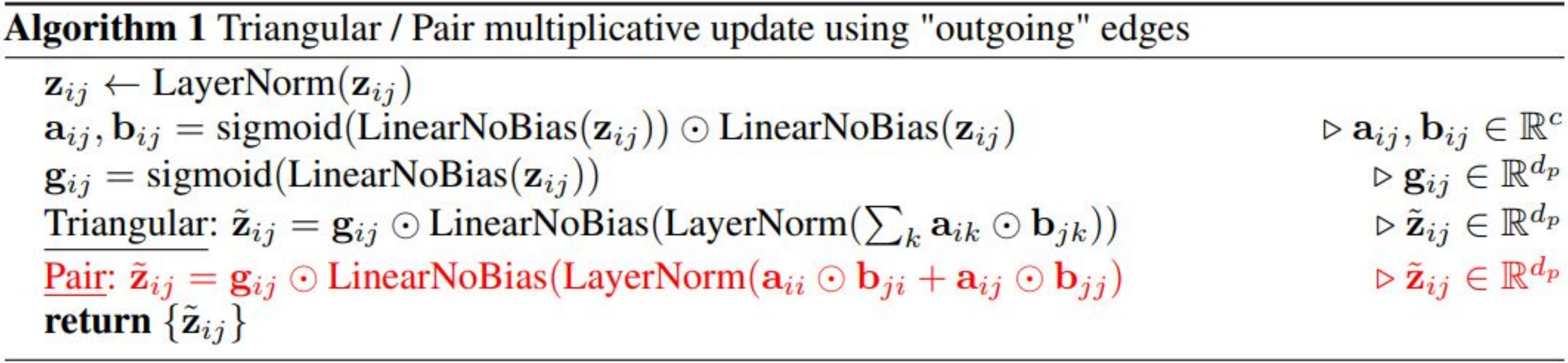


Results: Triangle Multiplicative Update Ablation

Triangle Multiplicative Updates and Triangle Attention is first used by AlphaFold2.

To ablate this mechanism, suggested alternative: Pair Update.

It keeps the parameter size same while removing the “third party” from triangular approach.



Algorithm 1 Triangular / Pair multiplicative update using "outgoing" edges

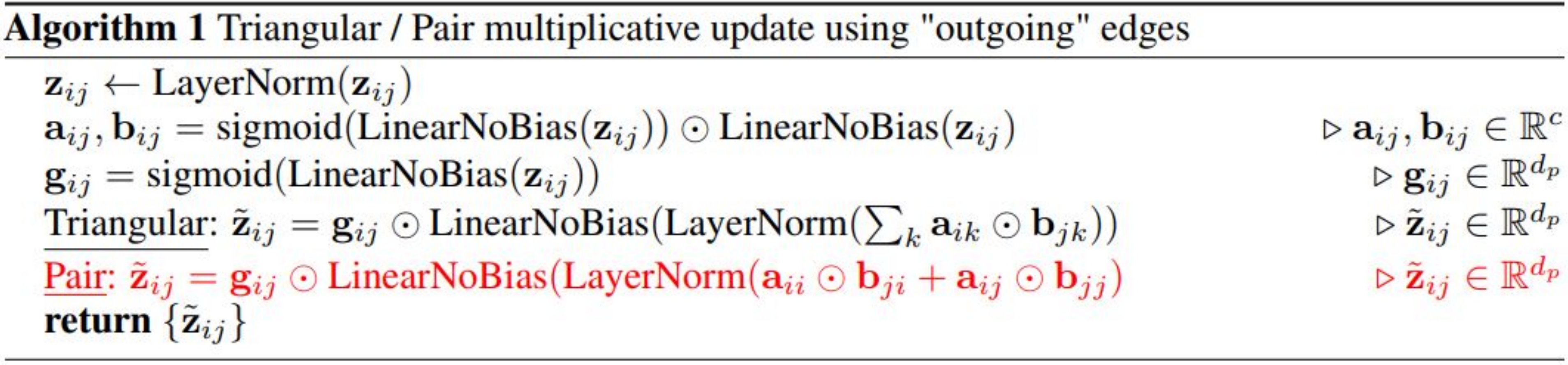
```
$\mathbf{z}_{ij} \leftarrow \mathrm{LayerNorm}(\mathbf{z}_{ij})$
$\mathbf{a}_{ij}, \mathbf{b}_{ij} = \mathrm{sigmoid}(\mathrm{LinearNoBias}(\mathbf{z}_{ij})) \odot \mathrm{LinearNoBias}(\mathbf{z}_{ij})$    $\triangleright\ \mathbf{a}_{ij}, \mathbf{b}_{ij} \in \mathbb{R}^c$
$\mathbf{g}_{ij} = \mathrm{sigmoid}(\mathrm{LinearNoBias}(\mathbf{z}_{ij}))$    $\triangleright\ \mathbf{g}_{ij} \in \mathbb{R}^{d_p}$
Triangular: $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \mathrm{LinearNoBias}(\mathrm{LayerNorm}(\sum_k \mathbf{a}_{ik} \odot \mathbf{b}_{jk}))$    $\triangleright\ \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{d_p}$
Pair: $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \mathrm{LinearNoBias}(\mathrm{LayerNorm}(\mathbf{a}_{ii} \odot \mathbf{b}_{ji} + \mathbf{a}_{ij} \odot \mathbf{b}_{jj})$    $\triangleright\ \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{d_p}$
return $\{\tilde{\mathbf{z}}_{ij}\}$
```

Results: Triangle Multiplicative Update Ablation

Motivation behind this analysis:

The inverse covariance matrix enables us to calculate partial correlations between positions i and j. Which is equivalent to factoring out effects on the position pair coming from other positions(?).

Since triangle updates are made for every possible third position k, maybe it achieves a similar effect.



Algorithm 1 Triangular / Pair multiplicative update using "outgoing" edges

```
$\mathbf{z}_{ij} \leftarrow \mathrm{LayerNorm}(\mathbf{z}_{ij})$
$\mathbf{a}_{ij}, \mathbf{b}_{ij} = \mathrm{sigmoid}(\mathrm{LinearNoBias}(\mathbf{z}_{ij})) \odot \mathrm{LinearNoBias}(\mathbf{z}_{ij})$        $\triangleright\ \mathbf{a}_{ij}, \mathbf{b}_{ij} \in \mathbb{R}^c$
$\mathbf{g}_{ij} = \mathrm{sigmoid}(\mathrm{LinearNoBias}(\mathbf{z}_{ij}))$        $\triangleright\ \mathbf{g}_{ij} \in \mathbb{R}^{d_p}$
Triangular: $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \mathrm{LinearNoBias}(\mathrm{LayerNorm}(\sum_k \mathbf{a}_{ik} \odot \mathbf{b}_{jk}))$        $\triangleright\ \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{d_p}$
Pair: $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \mathrm{LinearNoBias}(\mathrm{LayerNorm}(\mathbf{a}_{ii} \odot \mathbf{b}_{ji} + \mathbf{a}_{ij} \odot \mathbf{b}_{jj})$        $\triangleright\ \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{d_p}$
return $\{\tilde{\mathbf{z}}_{ij}\}$
```

Results: Triangle Multiplicative Update Ablation

Using the best pre-trained model, after fine-tuning and validation, long-range P@L on the CASP15 targets decreased from 52% to just 34% when Pair Updates approach is employed.

To further analyze, 16 targets where MSA Pairformer with triangle updates has at least 75% long-range P@L have been selected. Then considering Pair Updates:

For each false positive (i, j):

- Check whether there exists a k where (i, k) and (j, k) are true contacts. This suggests the model incorrectly inferred (i, j) through k. Call this mediation rate.
- To build a null distribution of mediation rate: select a random residue p (>= 24 apart) and check whether (i, k) and (p, k) are true contacts

Results: Triangle Multiplicative Update Ablation

Now, check whether false positives are enriched in terms of this “mediation rate”, meaning existence of a third contact point, a mediator.

False positives from 12 of 16 targets were enriched (significantly, $p < 0.05$) in high mediation rates. The remaining 4 targets were among the shortest sequences in the dataset.

On average, 41% of false positives have had a third residue which both are in contact.

Results: Triangle Multiplicative Update Ablation

Furthermore, to better check this effect, MSA Pairformer predictions are worsened with random noise until it matches the performance of Pair Updates and these enrichment values are recalculated. These false positive predictions were not significantly enriched in higher mediation rates.

Also as an alternative experiment, Pair Updates vs Triangle Updates models trained from scratch using same hyperparameters. On a held-out test set of 400 MSA's, although both models have shown good masked amino-acid prediction performance (perplexity 3.98 vs 4.12, accuracy 0.59 vs 0.58), Pair Updates model have had 16% points worse performance on long range contact prediction (P@L 0.36 vs 0.52)

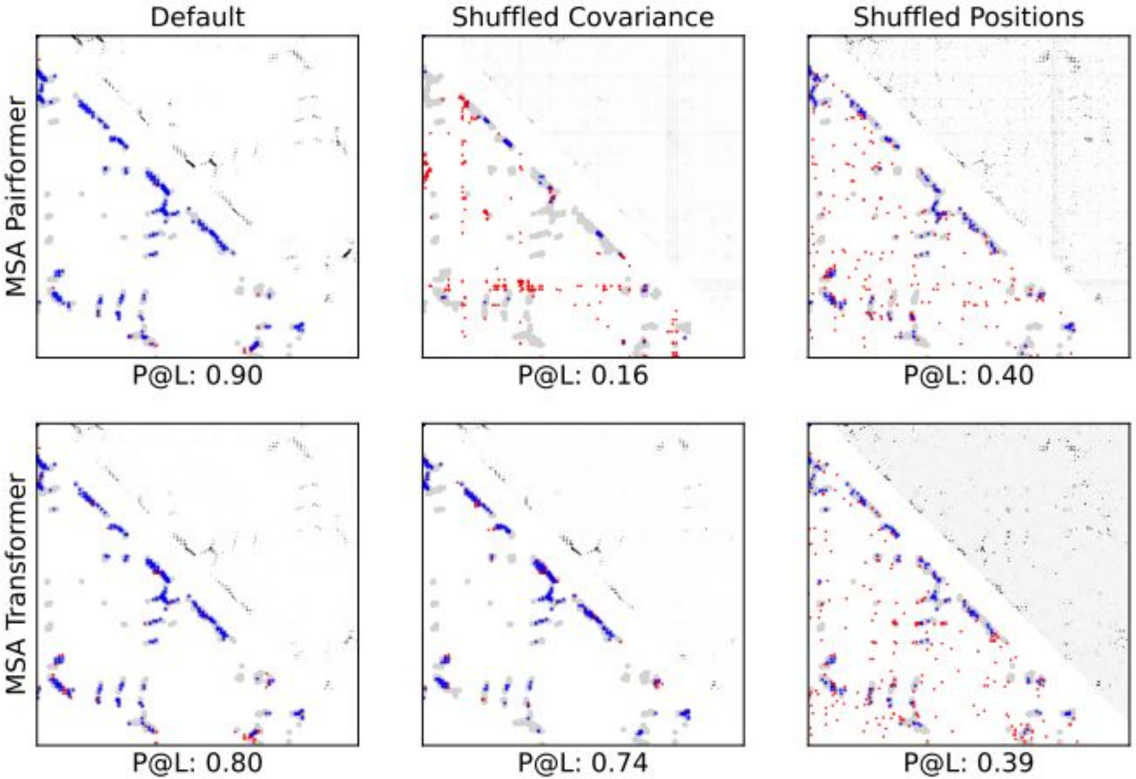
Results: MSA perturbations reveal distinct mechanisms between MSA Pairformer and MSA Transformer

They tested how the models respond in two different scenarios:

- Shuffled covariance: Randomly permute the values in each column of MSA. (Preserve profiles, destroy pairwise correlations)
- Shuffled positions: Randomly permute order of columns. (Both preserve profiles and pairwise correlations)

Use 96 randomly selected targets from trRosetta training set (max length 512).

Results: MSA perturbations reveal distinct mechanisms between MSA Pairformer and MSA Transformer



Discussion/Conclusion

- MSA Pairformer have shown better performance across many tasks while having 100x less parameters than some larger pLM models.
- Single-sequence pLM models were favored due to circumvention of MSA generation. However, methods for generating MSA’s are much faster now.
- MSA Pairformer may adapt to a newly discovered family of proteins in inference time where single sequence pLM’s might have to be retrained.
- Performance of approaches such MSA Pairformer challenge the scaling paradigm.

Thanks for listening