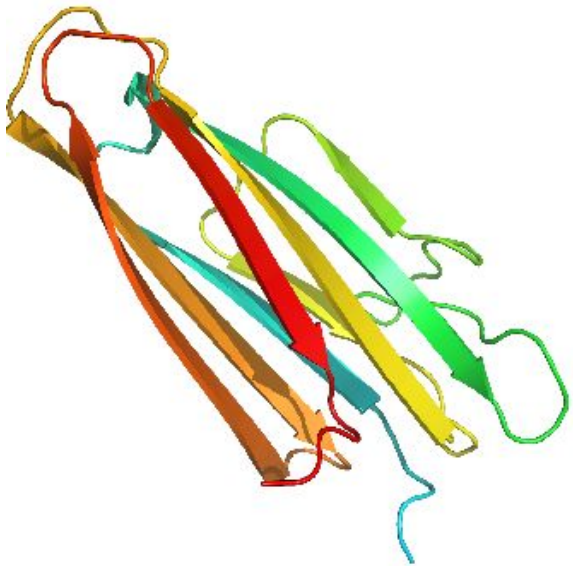
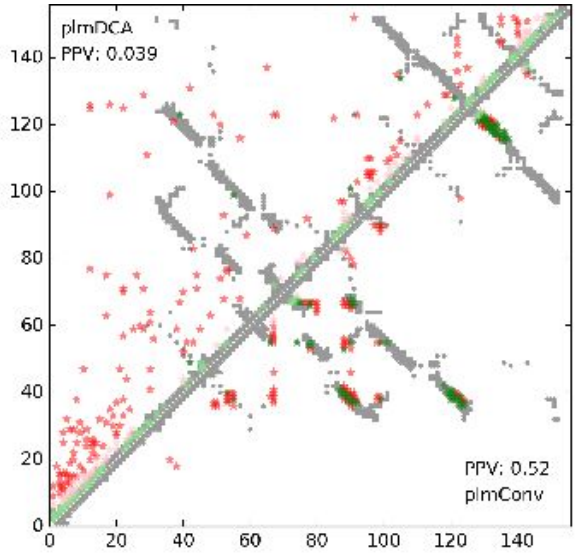
Machine learning 4c

Contact predictions using a CNN

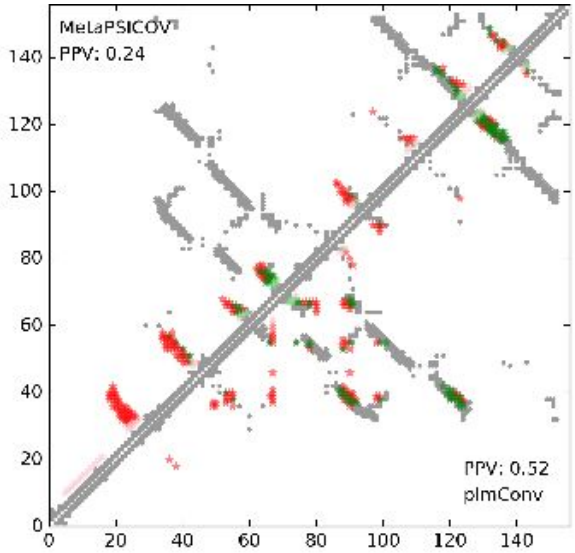
Protein Contact Predictions



(a) Structure



(b) Contact maps predicted by our method vs. plmDCA

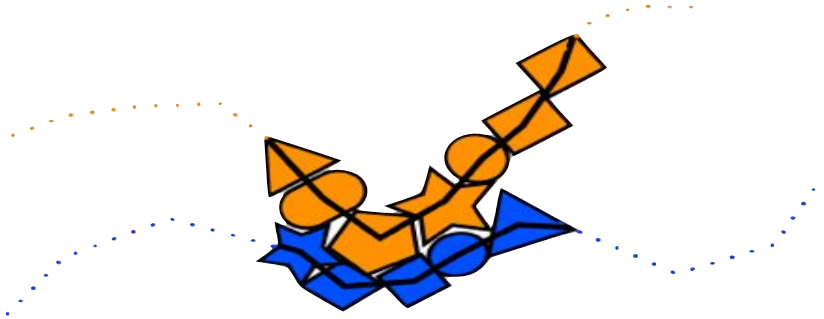


(c) Contact maps predicted by our method vs. MetaPSICOV

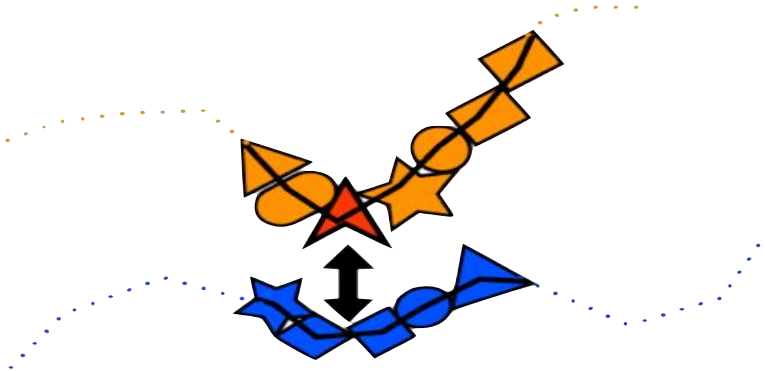
Decoupling direct interactions
from indirect ones
Giraud, Phys Rev E Stat 1999

SPATIAL PROXIMITY INDUCES SEQUENCE COUPLING

Native interactions



Unfavourable mutation



Compensating mutation

Indirect couplings

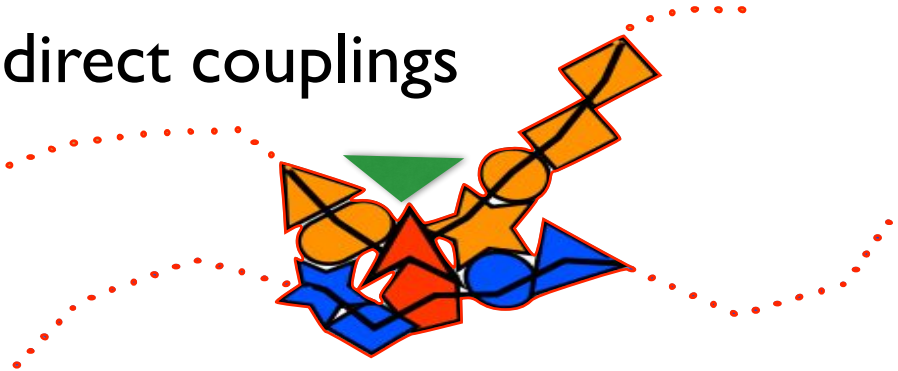
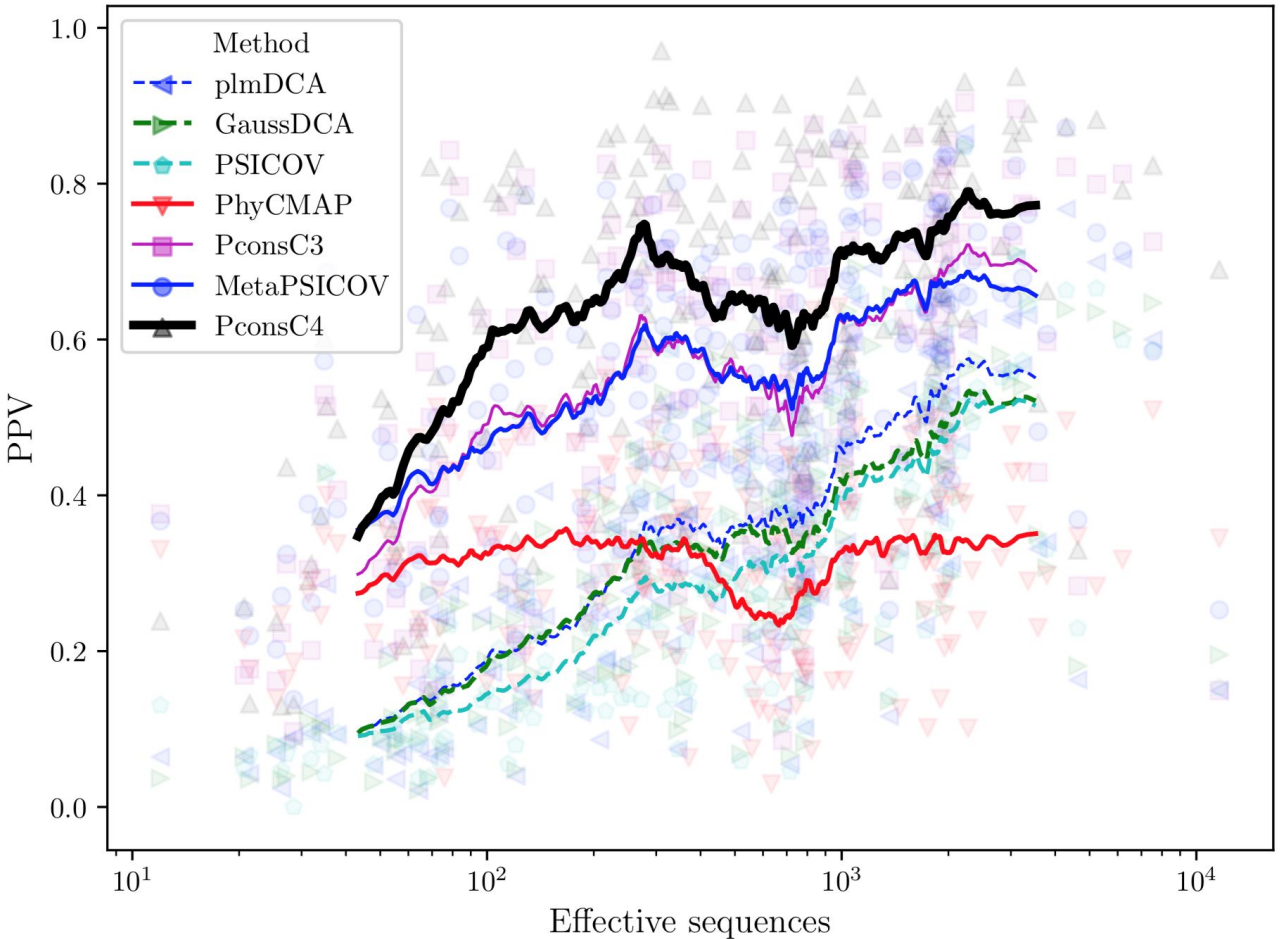


Image source: Andrea Pagnani, International School on Physics of Complex Systems 2012

Improvement over time



Train a network

Background

Proteins fold into 3D structures, and knowing which amino acids are in contact (distance < 8Å) helps predict these structures. Tools like trRosetta u
deep learning to predict contacts from sequence information.

What we'll build

1. **Input representation**: Encode protein sequences (20 amino acids + features) into 2D grids
2. **Contact labels**: Create synthetic distance maps (1 = contact, 0 = no contact)
3. **CNN architecture**: Adapt Conf4/Conf5 from Chapter 7 for contact prediction
4. **Training & evaluation**: Train on synthetic data, visualize predictions

Note: This is an educational example with toy data. Real contact predictors use MSA features, evolutionary information, and much larger datasets.

```
✓ import numpy as np ⋯
```

```
Device: cuda
```

1) Protein Sequence Representation

We'll encode amino acids and create synthetic protein "features."

```
✓ # Standard amino acids (20) ⋯
```

```
Amino acids: ACDEFGHIKLMNPQRSTVWY
Example: A=0, W=18

Sequence length: 20
One-hot shape: (20, 20)
PSSM shape: (20, 20)
Contact map shape: (20, 20)
Num contacts: 64 / 400
```

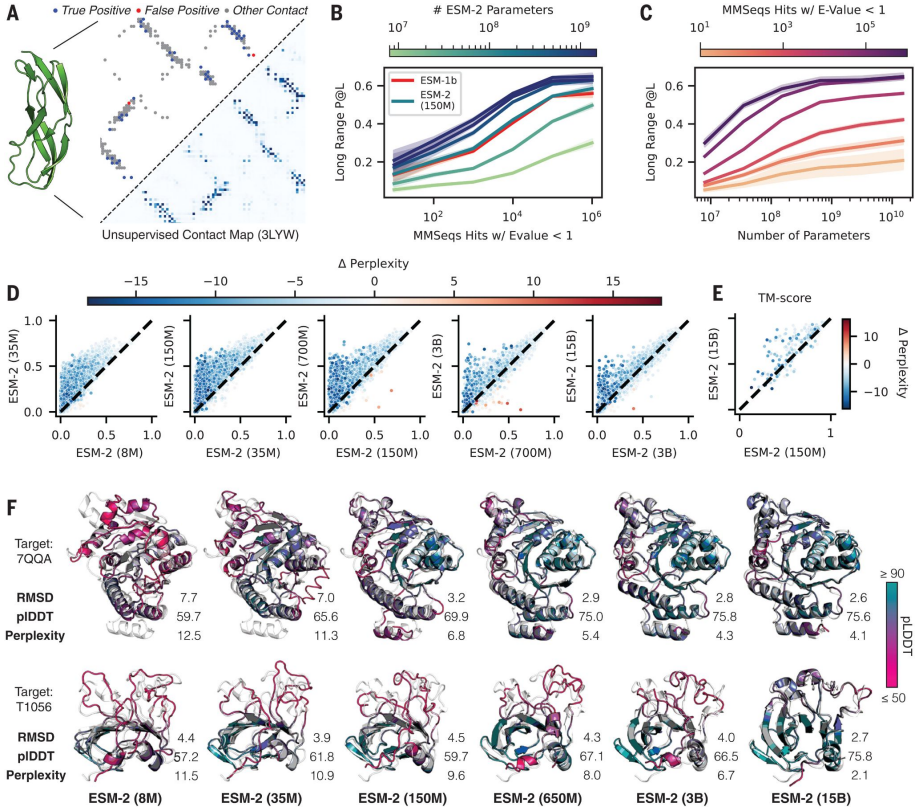
ESM-2: Protein Language Model Features

ESM-2 (Evolutionary Scale Modeling) is a transformer-based protein language model from Meta AI. It learns representations directly from protein sequences without alignment information, capturing:

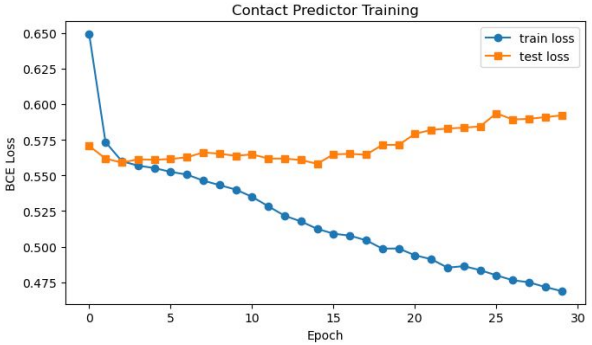
- Structural properties
- Functional information
- Evolutionary patterns

Key advantage: ESM-2 embeddings are learned from 2.7 billion proteins and are more informative than simple one-hot + PSSM features.

ESM-2 (instead of MSAs)



Training



2) Create Training Dataset

Generate synthetic protein sequences with contact labels.

```
class ProteinDataset(Dataset): ⋯
```

```
Train: 80 proteins
Test: 20 proteins
Batch input shape: torch.Size([4, 40, 64, 64]) (batch, channels, height, width)
Batch target shape: torch.Size([4, 1, 64, 64]) (batch, 1, height, width)
Contact density: 0.235
```

Generate + Code + Markdown

3) Contact Prediction CNN (adapted from Conf4/Conf5)

We'll use a U-Net-like architecture for dense per-position predictions.

```
class ContactPredictorCNN(nn.Module): ⋯
```

```
Model parameters: 1.58M
Input shape: torch.Size([4, 40, 64, 64])
Output shape: torch.Size([4, 1, 64, 64])
```

4) Training Loop

Train the contact predictor with binary cross-entropy loss.

```
criterion = nn.BCEWithLogitsLoss() ⋯
```

```
Epoch 01/30 | train loss 0.6490 | test loss 0.5707
Epoch 05/30 | train loss 0.5552 | test loss 0.5610
Epoch 10/30 | train loss 0.5401 | test loss 0.5637
Epoch 15/30 | train loss 0.5125 | test loss 0.5582
Epoch 20/30 | train loss 0.4987 | test loss 0.5714
Epoch 25/30 | train loss 0.4836 | test loss 0.5844
Epoch 30/30 | train loss 0.4689 | test loss 0.5921
Training complete!
```

Not that good (we will get back to this later)