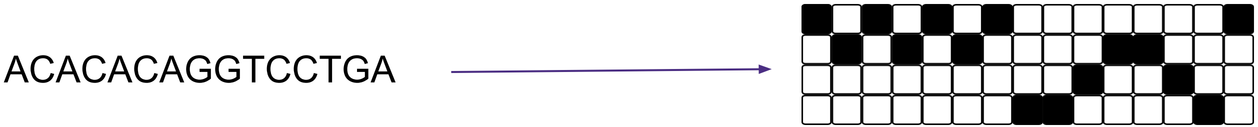
Interpreting Neural Networks for Biological Sequences by Learning Masks

Johannes Linder, Alyssa La Fleur, Sreeram Kannan, Zibo Chen, Ajasja Ljubetič, David Baker, Georg Seelig

UNIVERSITY *of* WASHINGTON

Feature attribution:

- *Feature attribution*: attributing a given prediction to the input values of a predictor
- One-hot encoded sequences:



ACACACAGGTCCTGA

Current feature attribution methods:

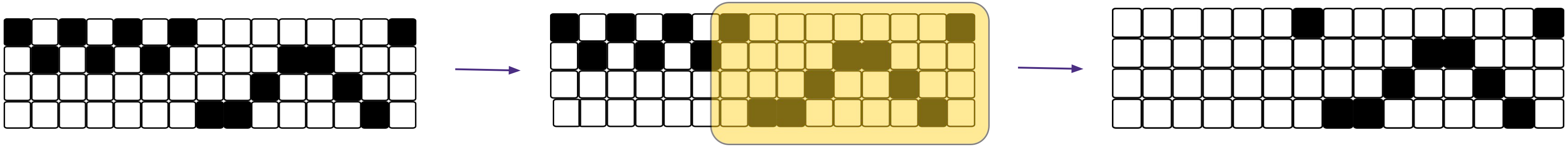
- Local approximation methods - base their estimation of importance on gradients or local linear models
- Generative masking methods from computer vision

The advantages of generative masking models for feature attribution:

- Generative attribution methods allow learning of overall patterns of important features from the training dataset
- May not be desirable in some cases - but in biology could be useful for uncovering regulatory logic

Discrete inputs & masking backgrounds:

- What kind of backgrounds should we be using for one-hot representation trained models when masking?
- Fading/blurring, zeros, random samples

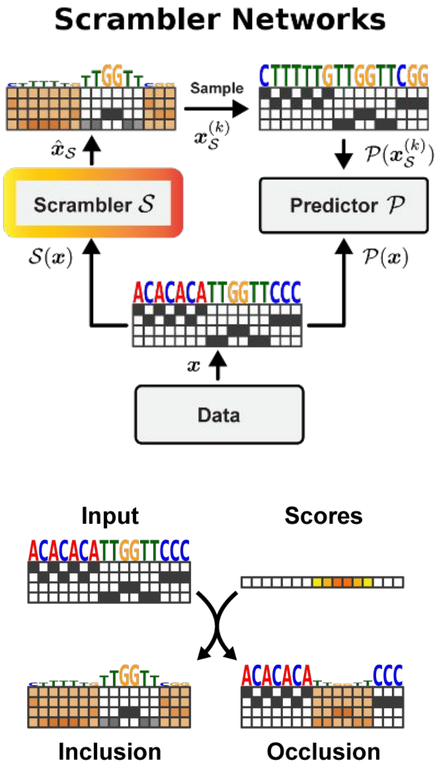


Scrambling Neural Networks

UNIVERSITY *of* WASHINGTON

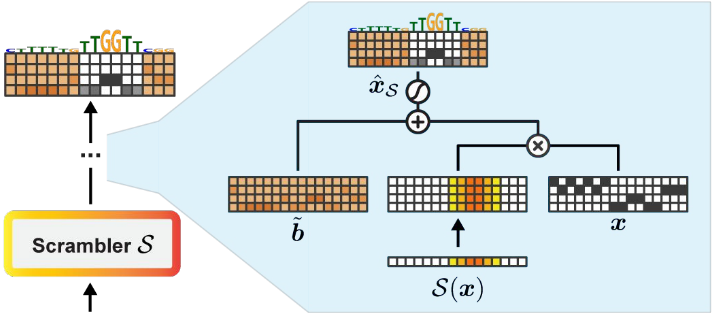
Scrambling neural networks (Scramblers):

- Inclusion: finding the smallest subset of features which, when preserved, preserve the prediction
- Occlusion: finding the smallest subset of features which, when perturbed, destroy the prediction



UNIVERSITY *of* WASHINGTON

Inclusion Objective


$\hat{\boldsymbol{x}}_{\mathcal{S}}$
Scrambler $\mathcal{S}$
$\tilde{\boldsymbol{b}}$
$\boldsymbol{x}$
$\mathcal{S}(\boldsymbol{x})$

Pre-trained predictor: $\mathcal{P}$

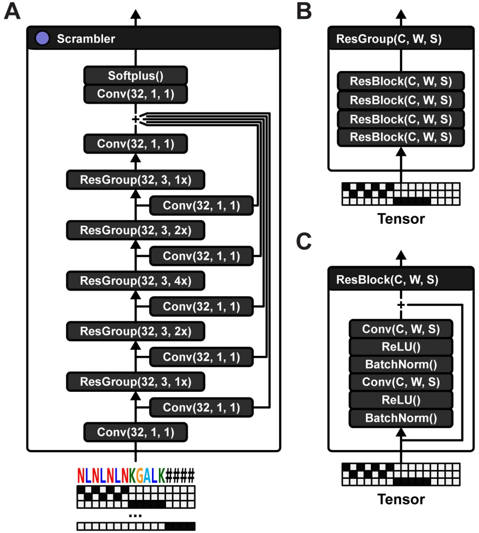
One-hot encoded input pattern: $\boldsymbol{x} \in \{0,1\}^{N \times M}$

Non-informative background distribution: $\tilde{\boldsymbol{b}} \in \mathbb{R}^{N \times M}$

Scrambler trainable network: $\mathcal{S}$, learns to generate real-valued importance scores $\mathcal{S}(\boldsymbol{x}) \in (0, \infty]^N$

$$\hat{\boldsymbol{x}}_{\mathcal{S}} = \sigma\left(\log \tilde{\boldsymbol{b}} + \boldsymbol{x} \times \dot{\mathcal{S}}(\boldsymbol{x})\right)$$

Where σ denotes the softmax $\sigma(\boldsymbol{l})_{ij} = \frac{e^{l_{ij}}}{\sum_{k=1}^{M} e^{l_{ik}}}$ and $\dot{\mathcal{S}}(\boldsymbol{x}) \in (0, \infty]^{N \times M}$ represent the importance scores $\mathcal{S}(\boldsymbol{x})$ which have been broadcasted at position i to all channels j.

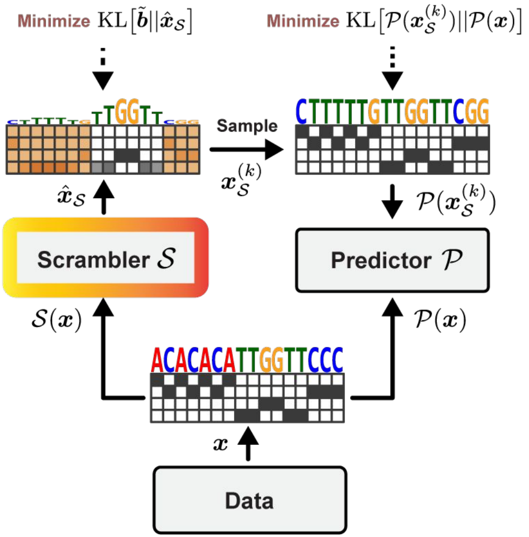


Inclusion Objective

To train, K discrete samples $\boldsymbol{x}_{\mathcal{S}}^{(k)}$ are drawn from $\hat{\boldsymbol{x}}_{\mathcal{S}}$ are passed to the predictor $\mathcal{P}$

Scrambled predictions $\mathcal{P}(\hat{\boldsymbol{x}}_{\mathcal{S}}^{(k)})$, original prediction $\mathcal{P}(\boldsymbol{x})$

$$\min_{\mathcal{S}} \left(\frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}\left[\mathcal{P}(\boldsymbol{x}_{\mathcal{S}}^{(k)})||\mathcal{P}(\boldsymbol{x})\right]\right) + \lambda\cdot\left(t_{\mathrm{bits}} - \frac{1}{N}\cdot \mathrm{KL}\left[\tilde{\boldsymbol{b}}||\hat{\boldsymbol{x}}_{\mathcal{S}}\right]\right)^2$$



Minimize $\mathrm{KL}[\tilde{\boldsymbol{b}}||\hat{\boldsymbol{x}}_{\mathcal{S}}]$
Minimize $\mathrm{KL}[\mathcal{P}(\boldsymbol{x}_{\mathcal{S}}^{(k)})||\mathcal{P}(\boldsymbol{x})]$
Sample
$\boldsymbol{x}_{\mathcal{S}}^{(k)}$
CTTTTTGTTGGTTCGG
$\hat{\boldsymbol{x}}_{\mathcal{S}}$
$\mathcal{P}(\boldsymbol{x}_{\mathcal{S}}^{(k)})$
Scrambler $\mathcal{S}$
Predictor $\mathcal{P}$
$\mathcal{S}(\boldsymbol{x})$
$\mathcal{P}(\boldsymbol{x})$
ACACACATTGGTTCCC
$\boldsymbol{x}$
Data

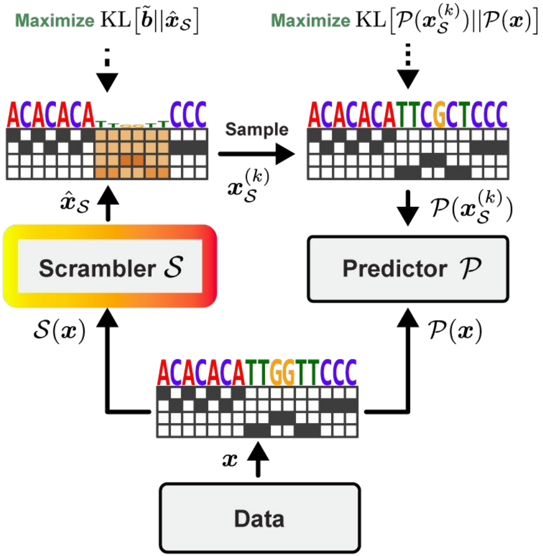
Occlusion objective

Occlusion scrambling operation:

$$\hat{\boldsymbol{x}}_{\mathcal{S}} = \sigma\big(\log \tilde{\boldsymbol{b}} + \boldsymbol{x}/\dot{\mathcal{S}}(\boldsymbol{x})\big)$$

Occlusion objective:

$$\min_{\mathcal{S}} \left(-\frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}\big[\mathcal{P}(\boldsymbol{x}_{\mathcal{S}}^{(k)})||\mathcal{P}(\boldsymbol{x})\big]\right) + \lambda \cdot \left(t_{\text{bits}} - \frac{1}{N}\cdot \mathrm{KL}\big[\tilde{\boldsymbol{b}}||\hat{\boldsymbol{x}}_{\mathcal{S}}\big]\right)^2$$

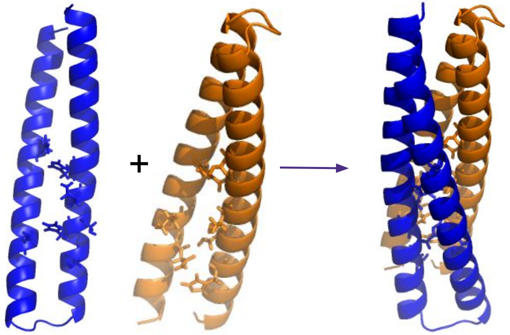

Maximize $\mathrm{KL}[\tilde{\boldsymbol{b}}||\hat{\boldsymbol{x}}_{\mathcal{S}}]$
Maximize $\mathrm{KL}[\mathcal{P}(\boldsymbol{x}_{\mathcal{S}}^{(k)})||\mathcal{P}(\boldsymbol{x})]$
ACACACA CCC
Sample
$\boldsymbol{x}_{\mathcal{S}}^{(k)}$
ACACACATTCGCTCCC
$\hat{\boldsymbol{x}}_{\mathcal{S}}$
$\mathcal{P}(\boldsymbol{x}_{\mathcal{S}}^{(k)})$
Scrambler $\mathcal{S}$
Predictor $\mathcal{P}$
$\mathcal{S}(\boldsymbol{x})$
$\mathcal{P}(\boldsymbol{x})$
ACACACATTGGTTCCC
$\boldsymbol{x}$
Data

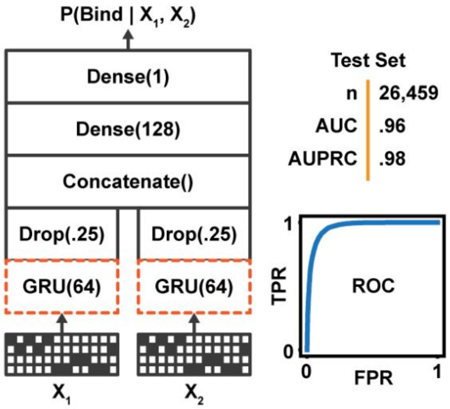
Protein Attributions

UNIVERSITY *of* WASHINGTON

Dimerization predictor:

- Set of coiled-coil dimers designed to interact
- HBNet - designed hydrogen bond network to induce binding specificity (Maguire et al., 2018; Chen et al., 2019)
- RNN trained for predicting if two dimers were designed to interact or not





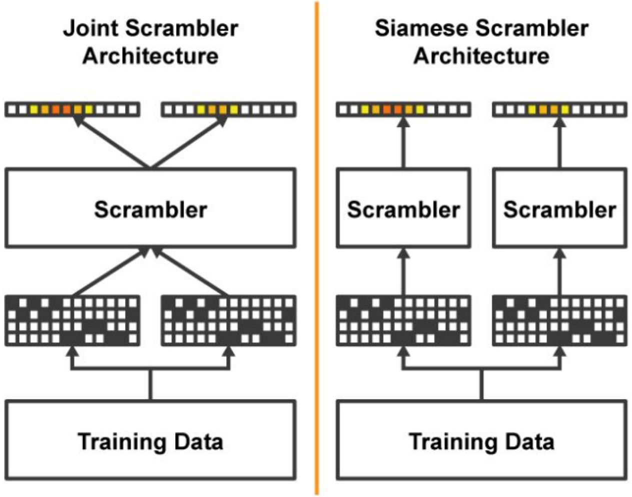
Test Set

n	26,459
AUC	.96
AUPRC	.98

Interpreting a Siamese network:

> Due to structure of how network takes in inputs, there are two ways we can structure Scramblers

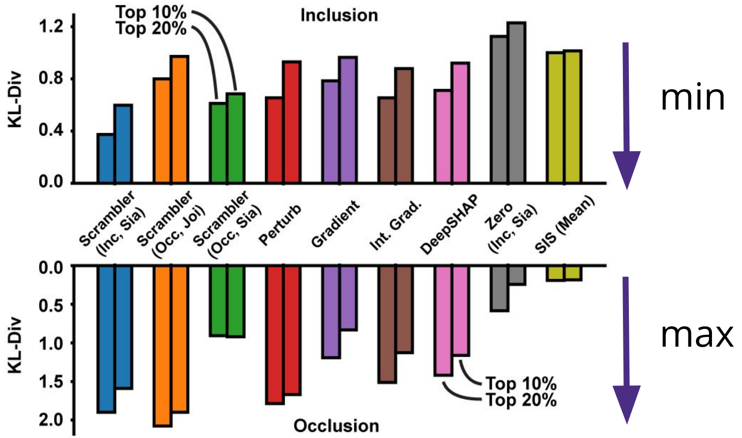
Can see both binder at a time, should be able to learn binding pair dependent features



Restricted to seeing one binder at a time, should learn features which are independent of binding pair

KL-Divergence & other methods

- Tested Scramblers against:
  - *Perturbation* - estimating importance by changing one position at a time
  - *Gradient Saliency* (Simonyan et al., 2013)
  - *Integrated gradients* (Sundararajan et al., 2017)
  - *DeepSHAP* (Lundberg et al., 2017)
  - *Zero* masking (similar to computer vision methods L2X (Chen et al., 2018) & INVASE (Yoon et al., 2018))
  - *Sufficient Input Subsets (SIS)* (Carter et al., 2019)

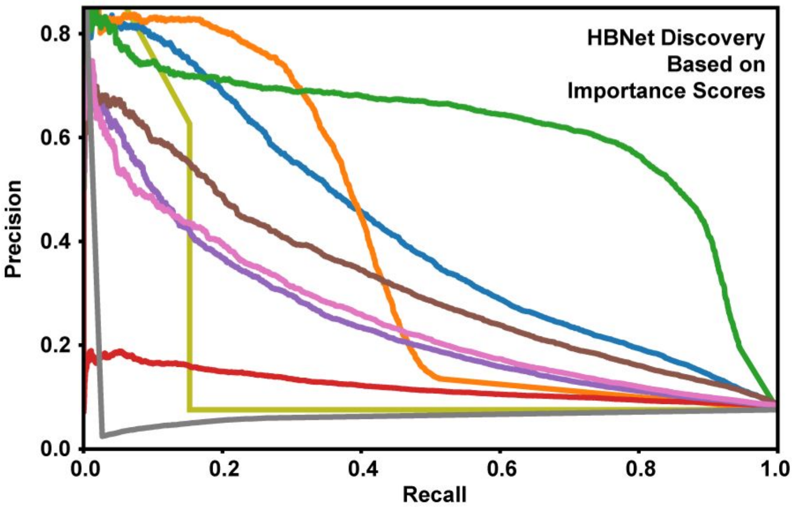


min

max

Benchmark 1: HBNet Recovery

	HBNet AP	
Scrambler (Inclusion, Siamese)	0.42	🟦
Scrambler (Occlusion, Joint)	0.37	🟧
Scrambler (Occlusion, Siamese)	<u>0.61</u>	🟩
Perturbation	0.12	🟥
Gradient	0.25	🟪
Integrated Gradients	0.32	🟫
DeepSHAP	0.26	■
Zero (Inclusion, Siamese)	0.07	⬛
SIS (Mean)	0.16	🟨



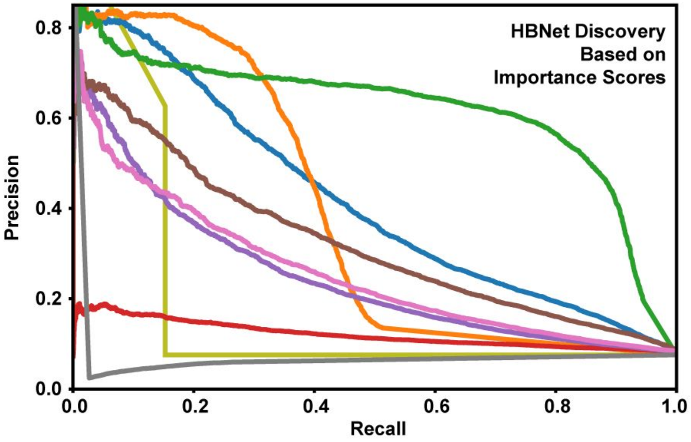
Test set of n=480 dimers, recovered HBNets from dimer pairs

UNIVERSITY *of* WASHINGTON

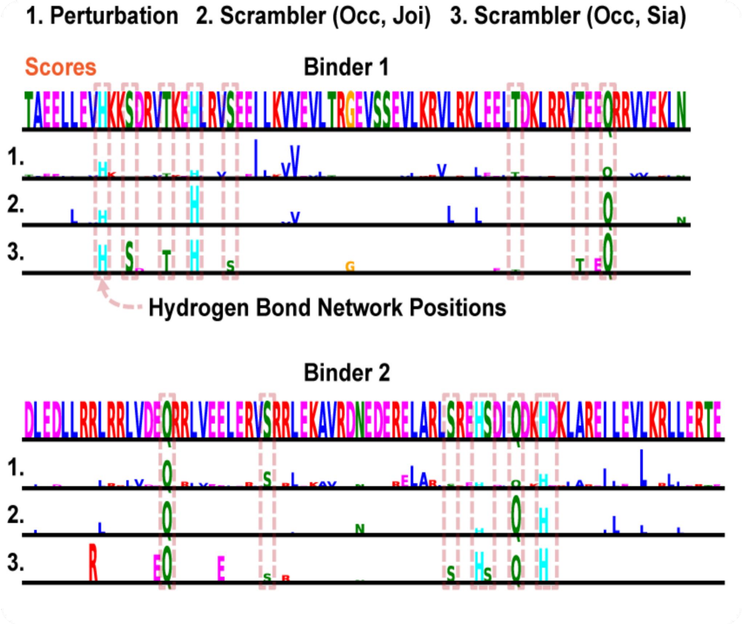
Benchmark 2: Mean Alanine scanning DDG

- Conducted *in silico* Ala scanning with PyRosetta for all residues in a dimer pair
- Calculated mean DDG for each & did permutation tests with 10,000 relabelings - all methods p <0.05

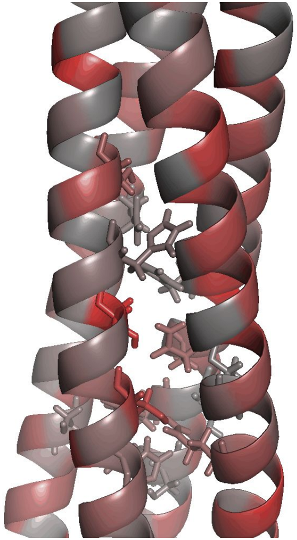
	ddG Score	HBNet AP	
Scrambler (Inclusion, Siamese)	1.70	0.42	🟦
Scrambler (Occlusion, Joint)	2.20	0.37	🟧
Scrambler (Occlusion, Siamese)	0.90	0.61	🟩
Perturbation	1.74	0.12	🟥
Gradient	0.86	0.25	🟪
Integrated Gradients	1.13	0.32	🟫
DeepSHAP	1.06	0.26	■
Zero (Inclusion, Siamese)	0.73	0.07	⬛
SIS (Mean)	0.44	0.16	🟨



Example dimer attribution

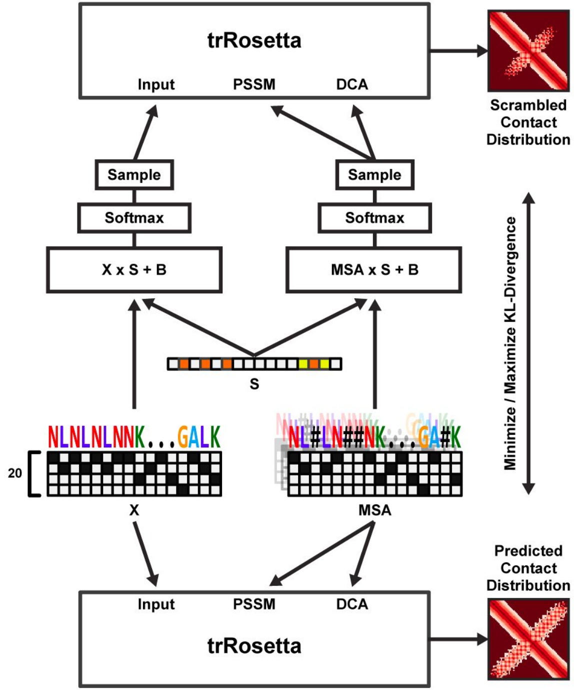


Iteration: 0



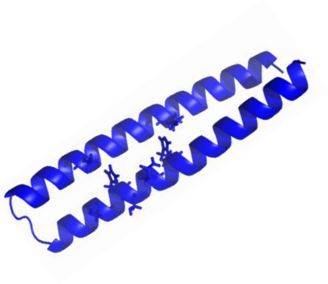
Protein structure prediction attribution

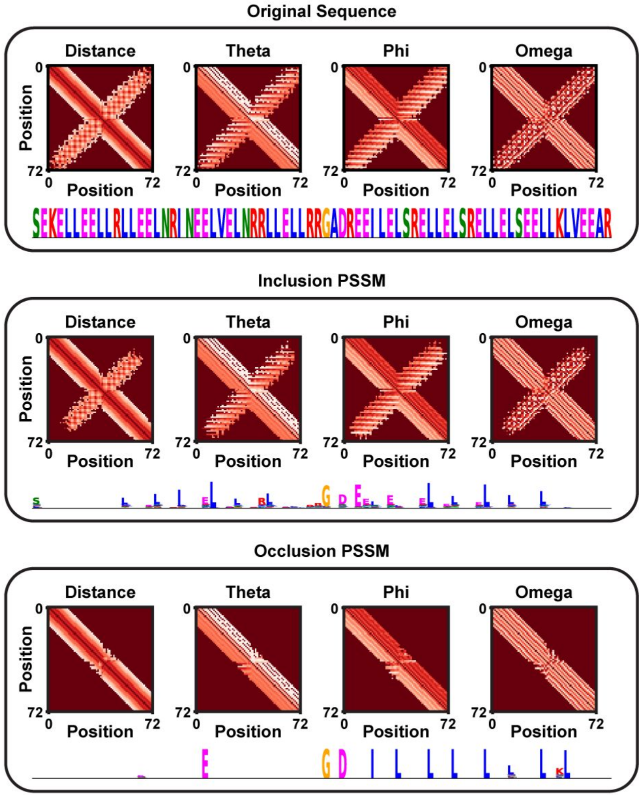
- trRosetta predicts distance and backbone angles for a tertiary structure (Yang et al., 2020)
- Used same Scrambler for protein sequence and MSA importance scores



Protein structure prediction attribution

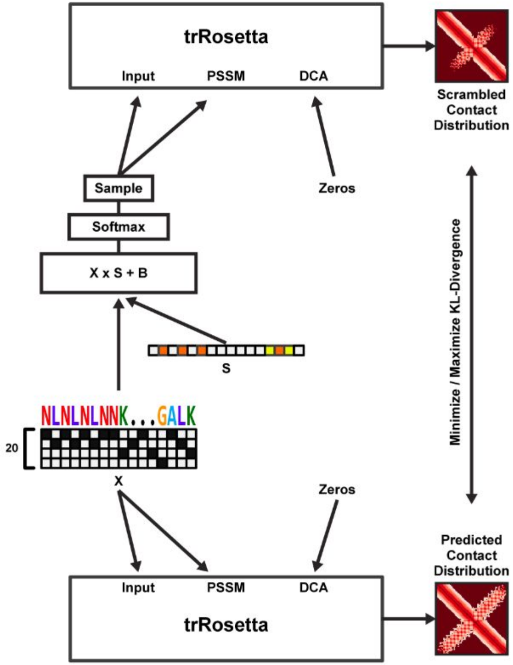
- > Hydrophobic leucines and a symmetry-breaking glycine in the hairpin region
- > Aligns well with previous results (Chen et al., 2019)





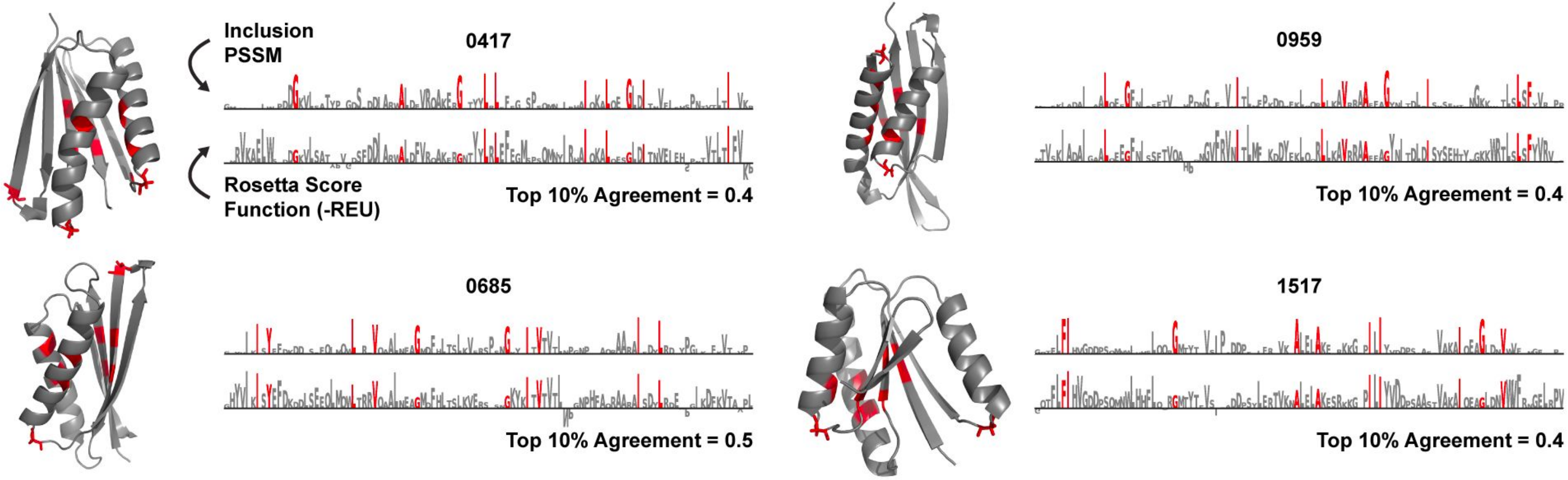
trRosetta *de novo* protein Scrambler:

- MSA free interpretation of *de novo* proteins without much natural sequence homology (Anishchenko et al., 2020)
- Unclear standard for validation
  - per-residue Rosetta energy breakdown



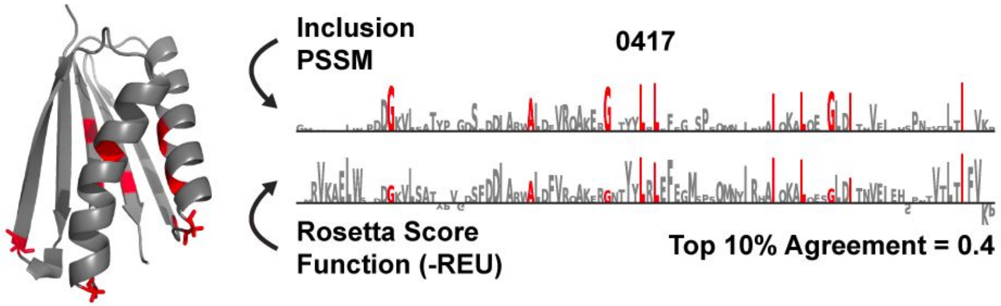
Per-residue -REU and scores:

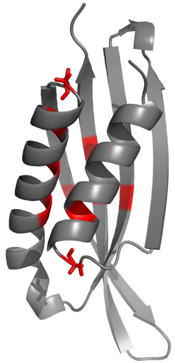
Measured agreement between top 10% of importance score positions & top 10% of -REU positions

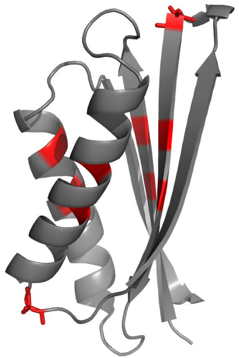


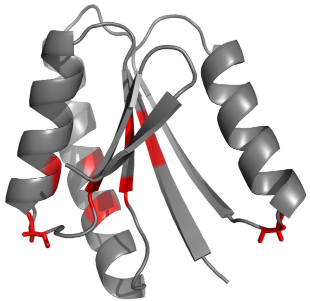
Scramblers identify loop glycines

- Glycines are known to occur on loops, thought to be important for maintaining loop flexibility




0959

0686



1517

Ongoing work

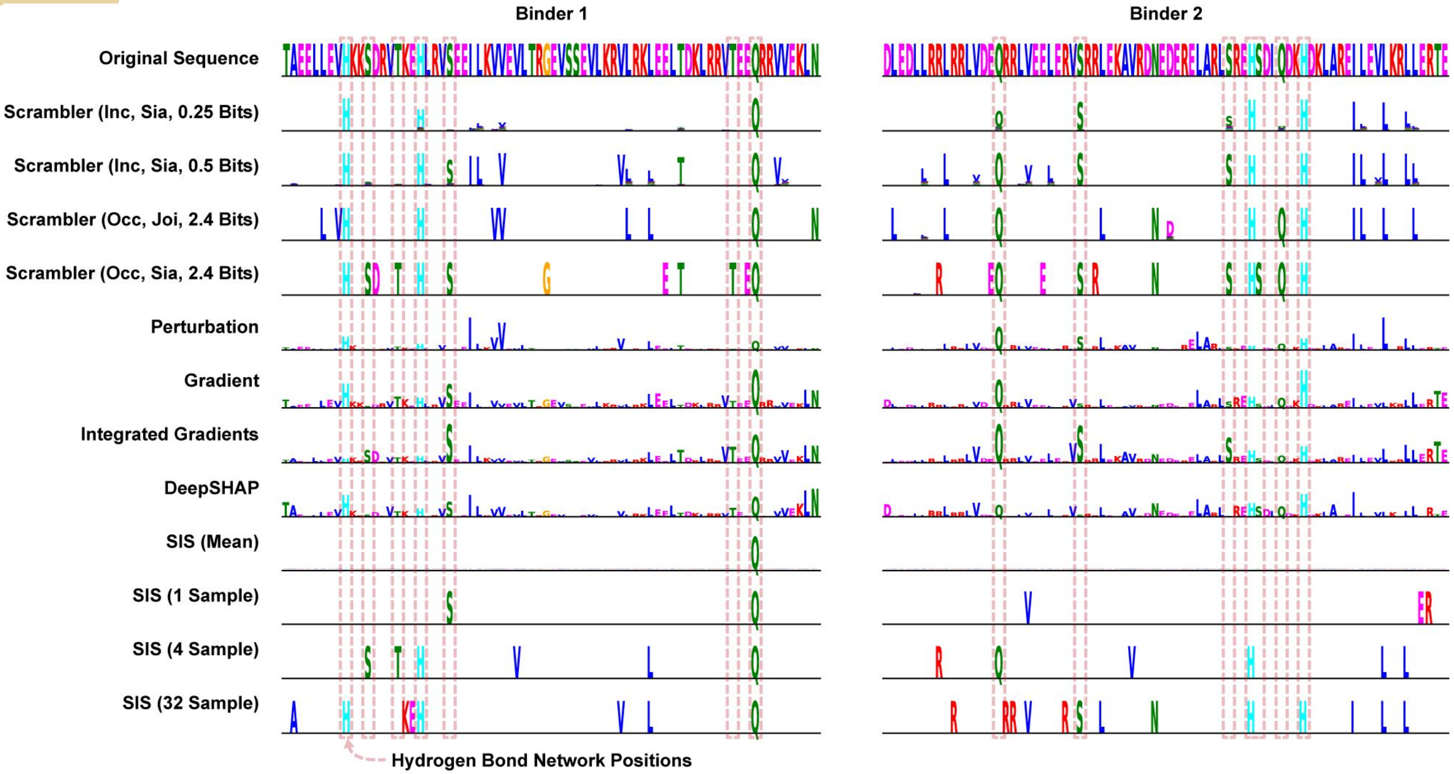
- Scrambler target bit ‘over-explanation’ corrections and attributions of sequences which have target values near the background

Thank you & any questions?

Github: johli/scrambler

Bioarxiv: Coming soon (hopefully end of week)

Example dimer comparison



References

- Anishchenko, I., Chidyausiku, T.M., Ovchinnikov, S., Pellock, S.J. and Baker, D., 2020. De novo protein design by deep network hallucination (bioRxiv).
- Carter, B., Mueller, J., Jain, S. and Gifford, D., 2019, April. What made you do this? understanding black-box decisions with sufficient input subsets. In The 22nd International Conference on Artificial Intelligence and Statistics, 567 -- 576.
- Chen, Z., Boyken, S.E., Jia, M., Busch, F., Flores-Solis, D., Bick, M.J., Lu, P., VanAernum, Z.L., Sahasrabuddhe, A., Langan, R.A. and Bermeo, S., 2019. Programmable design of orthogonal protein heterodimers. Nature, 565, 106--111.
- Cheng, J., Nguyen, T.Y.D., Cygan, K.J., Çelik, M.H., Fairbrother, W.G. and Gagneur, J., 2019. MMSplice: modular modeling improves the predictions of genetic variant effects on splicing. Genome biology, 20, 1--15.
- Lundberg, S.M. and Lee, S.I., 2017. A unified approach to interpreting model predictions. In Advances in neural information processing systems, 4765--4774.
- Maguire, J., Boyken, S., Baker, D., Kuhlman, B., 2018. Rapid Sampling of Hydrogen Bond Networks for Computational Protein Design. J Chem Theory Comput., 14, 2571--2760.
- Simonyan, K., Vedaldi, A. and Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps (arXiv).
- Sundararajan, M., Taly, A. and Yan, Q., 2017. Axiomatic attribution for deep networks (arXiv).
- Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S. and Baker, D., 2020. Improved protein structure prediction using predicted interresidue orientations. Proceedings of the National Academy of Sciences.
- Yoon, J., Jordon, J. and van der Schaar, M., 2018, September. INVASE: Instance-wise variable selection using neural networks. In International Conference on Learning Representations.