nature portfolio

https://doi.org/10.1038/s42256-021-00428-6

Supplementary information

Interpreting neural networks for biological sequences by learning stochastic masks

In the format provided by the authors and unedited

Supplementary Information

Interpreting Neural Networks for Biological Sequences by Learning Stochastic Masks

Johannes Linder[1,*], Alyssa La Fleur[1], Zibo Chen[2], Ajasja Ljubetič[2], David Baker[2], Sreeram Kannan[3], and Georg Seelig[1,3]

[1]Paul G. Allen School of Computer Science and Engineering, University of Washington
[2]Institute for Protein Design, University of Washington
[3]Department of Electrical and Computer Engineering, University of Washington
[*]Correspondence to: jlinder2@cs.washington.edu

Additional Attribution Methods

This section lists additional attribution methods and variations of the Scrambler model that were included in the supplementary benchmark comparisons for the PPI task (Extended Data Fig. 5).

Scrambler (Inclusion, Siamese, K Samples): By default, the Scrambler networks are trained by drawing 32 Gumbel samples from each scrambled PSSM in the training batch. In these versions, we instead draw K samples for each pattern in the training batch.

Scrambler (Inclusion, Cont. Mean Bg): Optimizes the same objective as the Inclusion-Scrambler, but uses a continuous interpolation perturbation with the mean training pattern $\tilde{B}$ instead of the temperature-based perturbator: $\min_{\mathcal{M}} \mathrm{KL}\big[\mathcal{P}(X \times \mathcal{M}(X) + \tilde{B} \times (1 - \mathcal{M}(X)))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}(X)_i\big)^2$, where $\mathcal{M}(X)$ is a real-valued, sigmoid masking model. Internally, network $\mathcal{M}$ is identical to the Scrambler.

Scrambler (Inclusion, Cont. No Bg): Optimizes the same objective as the Inclusion-Scrambler, but uses a continuous interpolation perturbation with the all-zero pattern instead of the temperature-based perturbator: $\min_{\mathcal{M}} \mathrm{KL}\big[\mathcal{P}(X \times \mathcal{M}(X))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}(X)_i\big)^2$, where $\mathcal{M}(X)$ is a real-valued, sigmoid masking model.

Scrambler (Inclusion, Cont. Rand Bg): Optimizes the same objective as the Inclusion-Scrambler, but uses a continuous interpolation perturbation with a random letter pattern $\mathbb{R} \in 0, 1^{N \times M}$ instead of the temperature-based perturbator: $\min_{\mathcal{M}} \mathrm{KL}\big[\mathcal{P}(X \times \mathcal{M}(X) + \mathbb{R} \times (1 - \mathcal{M}(X)))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}(X)_i\big)^2$, where $\mathcal{M}(X)$ is a real-valued, sigmoid masking model.

Scrambler (Occlusion, Cont. Mean Bg): Optimizes the same objective as the Occlusion-Scrambler, but uses a continuous interpolation perturbation with the mean training pattern $\tilde{B}$ instead of the temperature-based perturbator: $\min_{\mathcal{M}} -\mathrm{KL}\big[\mathcal{P}(X \times (1 - \mathcal{M}(X)) + \tilde{B} \times \mathcal{M}(X))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}(X)_i\big)^2$, where $\mathcal{M}(X)$ is a real-valued, sigmoid masking model.

Scrambler (Inclusion, Per Example): Optimizes the same objective as the Inclusion-Scrambler, but does not train a network to predict importance scores. Instead, this method optimizes a set of importance scores $\boldsymbol{s}$ for each pattern X that we wish to interpret individually: $\min_{\boldsymbol{s}} \big(1/K \cdot \sum_{k=1}^{K} \mathrm{KL}\big[\mathcal{P}(X_{\boldsymbol{s}}^{(k)})||\mathcal{P}(X)\big]\big) + \lambda \cdot \big(t_{\text{bits}} - 1/N \cdot \mathrm{KL}\big[\tilde{B}_1||\hat{X}_{\boldsymbol{s}}\big]\big)^2$, where $X_{\boldsymbol{s}}^{(k)} \sim \hat{X}_{\boldsymbol{s}}$ (letter-by-letter Gumbel-sampling) and $\hat{X}_{\boldsymbol{s}} = \sigma\big(\log \tilde{B} + X \times \dot{\boldsymbol{s}}\big)$. $\dot{\boldsymbol{s}} \in \mathbb{R}^{N \times M}$ is a channel-broadcasted copy of the importance scores $\boldsymbol{s} \in \mathbb{R}^N$. We obtain $\boldsymbol{s}$ from a vector of real-valued (trainable) numbers that we instance-normalize and apply the softplus activation on. We train $\boldsymbol{s}$ until convergence using the Adam optimizer with learning rate $= 0.01$, $\beta_1 = 0.5$ and $\beta_2 = 0.9$.

Scrambler (Inclusion, Per Example, Cont. Mean Bg): Optimizes the same objective as the Per-example Inclusion-Scrambler, but uses a continuous interpolation perturbation with the mean training pat-

tern $\tilde{B}$ instead of the temperature-based perturbator: $\min_{\boldsymbol{m}} \mathrm{KL}\big[\mathcal{P}(X \times \dot{\boldsymbol{m}} + \tilde{B} \times (1 - \dot{\boldsymbol{m}}))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \boldsymbol{m}_i\big)^2$, where $\boldsymbol{m} \in [0,1]^N$ is a continuous sigmoid-restricted mask pattern.

Scrambler (Inclusion, Per Example, Cont. No Bg): Optimizes the same objective as the Per-example Inclusion-Scrambler, but uses a continuous interpolation perturbation with the all-zero pattern instead of the temperature-based perturbator: $\min_{\boldsymbol{m}} \mathrm{KL}\big[\mathcal{P}(X \times \dot{\boldsymbol{m}})||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \boldsymbol{m}_i\big)^2$, where $\boldsymbol{m} \in [0,1]^N$ is a continuous sigmoid-restricted mask pattern.

Scrambler (Inclusion, Per Example, Cont. Rand Bg): Optimizes the same objective as the Per-example Inclusion-Scrambler, but uses a continuous interpolation perturbation with a random letter pattern $\mathbb{R} \in 0, 1^{N \times M}$ instead of the temperature-based perturbator: $\min_{\boldsymbol{m}} \mathrm{KL}\big[\mathcal{P}(X \times \dot{\boldsymbol{m}} + \mathbb{R} \times (1 - \dot{\boldsymbol{m}}))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \boldsymbol{m}_i\big)^2$, where $\boldsymbol{m} \in [0,1]^N$ is a continuous sigmoid-restricted mask pattern.

Scrambler (Occlusion, Per Example, Cont. Mean Bg): Optimizes the same objective as the Occlusion-Scrambler (for a single pattern), but uses a continuous interpolation perturbation with the mean training pattern $\tilde{B}$ instead of the temperature-based perturbator: $\min_{\boldsymbol{m}} -\mathrm{KL}\big[\mathcal{P}(X \times (1 - \dot{\boldsymbol{m}}) + \tilde{B} \times \dot{\boldsymbol{m}})||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \boldsymbol{m}_i\big)^2$, where $\boldsymbol{m} \in [0,1]^N$ is a continuous sigmoid-restricted mask pattern.

Zero (Occlusion, Joint): Given pairs of input patterns X_1 and X_2, we optimize $\min_{\mathcal{M}} -\mathrm{KL}\big[\mathcal{P}(X_1 \times (1 - \mathcal{M}^{(G)}(X_1, X_2)_1), X_2 \times (1 - \mathcal{M}^{(G)}(X_1, X_2)_2))||\mathcal{P}(X_1, X_2)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1, X_2)_{1i}\big)^2 + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1, X_2)_{2i}\big)^2$, where $\mathcal{M}$ is a pair-wise binary 0/1 masking model. See 'Zero Scrambler' for more details.

Zero (Inclusion, Siamese): Given pairs of input patterns X_1 and X_2, we optimize $\min_{\mathcal{M}} \mathrm{KL}\big[\mathcal{P}(X_1 \times \mathcal{M}^{(G)}(X_1), X_2 \times \mathcal{M}^{(G)}(X_2))||\mathcal{P}(X_1, X_2)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1)_i\big)^2 + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_2)_i\big)^2$, where $\mathcal{M}$ is a binary 0/1 masking model. See 'Zero Scrambler' for more details.

Zero (Occlusion, Siamese): Given pairs of input patterns X_1 and X_2, we optimize $\min_{\mathcal{M}} -\mathrm{KL}\big[\mathcal{P}(X_1 \times (1 - \mathcal{M}^{(G)}(X_1)), X_2 \times (1 - \mathcal{M}^{(G)}(X_2)))||\mathcal{P}(X_1, X_2)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1)_i\big)^2 + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_2)_i\big)^2$, where $\mathcal{M}$ is a binary 0/1 masking model. See 'Zero Scrambler' for more details.

Rand (Inclusion): Optimizes the same high-level objective as the Scrambler, but this model uses a perturbation operator where de-selected features are replaced with a random nucleotide (or residue). Specifically, Rand (Inclusion) optimizes $\min_{\mathcal{M}} \mathrm{KL}\big[\mathcal{P}(X \times \mathcal{M}^{(\mathrm{G})}(X) + \mathbb{R} \times (1 - \mathcal{M}^{(\mathrm{G})}(X)))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(\mathrm{S})}(X)_i\big)^2$, where $\mathcal{M}$ is a binary 0/1 masking model and $\mathbb{R}$ is a random letter pattern. See 'Zero Scrambler' for more details.

Rand (Occlusion): Optimizes $\min_{\mathcal{M}} -\mathrm{KL}\big[\mathcal{P}(X \times (1 - \mathcal{M}^{(\mathrm{G})}(X)) + \mathbb{R} \times \mathcal{M}^{(\mathrm{G})}(X))||\mathcal{P}(X)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(\mathrm{S})}(X)_i\big)^2$, where $\mathcal{M}$ is a binary 0/1 masking model and $\mathbb{R}$ is a random letter pattern. See 'Zero Scrambler' for more details.

Rand (Occlusion, Joint): Given pairs of input patterns X_1 and X_2, we optimize $\min_{\mathcal{M}} -\mathrm{KL}\big[\mathcal{P}(X_1 \times (1 - \mathcal{M}^{(G)}(X_1, X_2)_1) + \mathbb{R}_1 \times \mathcal{M}^{(G)}(X_1, X_2)_1, X_2 \times (1 - \mathcal{M}^{(G)}(X_1, X_2)_2) + \mathbb{R}_2 \times \mathcal{M}^{(G)}(X_1, X_2)_2)||\mathcal{P}(X_1, X_2)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1, X_2)_{1i}\big)^2 + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1, X_2)_{2i}\big)^2$, where $\mathcal{M}$ is a pair-wise binary 0/1 masking model and $\mathbb{R}_1, \mathbb{R}_2$ are random letter patterns. See 'Zero Scrambler' for more details.

Rand (Inclusion, Siamese): Given pairs of input patterns X_1 and X_2, we optimize $\min_{\mathcal{M}} \mathrm{KL}\big[\mathcal{P}(X_1 \times \mathcal{M}^{(G)}(X_1) + \mathbb{R}_1 \times (1 - \mathcal{M}^{(G)}(X_1)), X_2 \times \mathcal{M}^{(G)}(X_2) + \mathbb{R}_2 \times (1 - \mathcal{M}^{(G)}(X_2)))||\mathcal{P}(X_1, X_2)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1)_i\big)^2 + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_2)_i\big)^2$, where $\mathcal{M}$ is a binary 0/1 masking model and $\mathbb{R}_1, \mathbb{R}_2$ are random letter patterns. See 'Zero Scrambler' for more details.

Rand (Occlusion, Siamese): Given pairs of input patterns X_1 and X_2, we optimize $\min_{\mathcal{M}} -\mathrm{KL}\big[\mathcal{P}(X_1 \times (1 - \mathcal{M}^{(G)}(X_1)) + \mathbb{R}_1 \times \mathcal{M}^{(G)}(X_1), X_2 \times (1 - \mathcal{M}^{(G)}(X_2)) + \mathbb{R}_2 \times \mathcal{M}^{(G)}(X_2))||\mathcal{P}(X_1, X_2)\big] + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_1)_i\big)^2 + \lambda \cdot \big(t_{\text{area}} - 1/N \cdot \sum_{i=1}^{N} \mathcal{M}^{(S)}(X_2)_i\big)^2$, where $\mathcal{M}$ is a binary 0/1 masking model and $\mathbb{R}_1, \mathbb{R}_2$ are random letter patterns. See 'Zero Scrambler' for more details.

Supplementary Tables

Supplementary Table 1: An overview of interpretation methods based on masking. The columns describe (in order): Optimization Algorithm – Whether the method uses a parametric model to infer masks, or whether the masks are optimized individually; Mask – Whether masked features are faded, blurred, zeroed, or replaced with sampled values from either marginal distributions or a generator ("Counterfactual"); Task – What task(s) the method was demonstrated on (C = Computer Vision, N = NLP, B = Biology); Predictor – Whether the method interprets an existing predictor model or if a new predictor is trained as part of the interpretation algorithm.

	Optimization Algorithm	Mask	Task	Predictor
Scrambler Network (this paper)	Parametric Model	Samples	C, B	Existing
Fong et al. (2017) (1)	Per-example SGD	Fade, Blur	C	Existing
Dabkowski et al. (2017) (2)	Parametric Model	Fade, Blur	C	Existing
Yoon et al. (2018) (INVASE) (3)	Parametric Model	Zero	C, N	New
Chen et al. (2018) (L2X)(4)	Parametric Model	Zero	C, N	New
Carter et al. (2019) (SIS) (5)	Per-example Recursion	Mean, Samples	C, N, B	Existing
Zintgraf et al. (2017) (6)	Per-example Sampling	Samples	C	Existing
Chang et al. (2018) (FIDO-CA) (7)	Per-example SGD	Counterfactual	C	Existing

References

1. Fong, R. C. & Vedaldi, A. Interpretable Explanations of Black Boxes by Meaningful Perturbation. in *2017 IEEE International Conference on Computer Vision (ICCV)* 3449–3457 (2017). doi:10.1109/ICCV.2017.371

2. Dabkowski, P. & Gal, Y. Real Time Image Saliency for Black Box Classifiers. in *Advances in Neural Information Processing Systems 30* (2017).

3. Yoon, J., Jordon, J. & van der Schaar, M. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations* (2018).

4. Chen, J., Song, L., Wainwright, M. J. & Jordan, M. I. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. Preprint at https://arxiv.org/abs/1802.07814 (2018).

5. Carter, B., Mueller, J., Jain, S., & 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. PMLR.

6. Zintgraf, L. M., Cohen, T. S., Adel, T. & Welling, M.. Visualizing deep neural network decisions: Prediction difference analysis. Preprint at https://arxiv.org/abs/1702.04595 (2017).

7. Chang, C. H., Creager, E., Goldenberg, A. & Duvenaud, D. Explaining image classifiers by counterfactual generation. Preprint at https://arxiv.org/abs/1807.08024 (2018).