Sample efficient learning of alternative splicing with active learning and deep generative models

Ting Chen
Department of Computer Science
Columbia University
Thesis Track
Supervised by: Dr. David A. Knowles

May 6, 2024

Abstract

Recent works have utilized massively parallel reporter assays (MPRAs) which combine high throughput sequencing with extensive sequence variation in order to measure the regulatory activity of sequences. However, these studies often rely on generating and assaying large libraries of randomized sequences. In this work, we look to develop data efficient methods for designing MPRAs by utilizing uncertainty estimates in an active learning setting. These uncertainty estimates are then used to identify subsequent sequences of interest to test and label. We evaluate our methods in an alternative splicing prediction task and overall find that Gaussian Process (GP) based methods outperform other baselines that utilize lower quality uncertainty estimates. We also present a preliminary findings motivating the use of deep generative models to produce synthetic candidate sequences to assay.

1 Introduction

Numerous recent works [Rosenberg et al., 2015, Bogard et al., 2019, Sample et al., 2018] have utilized large libraries of randomized sequences in order to model biological behavior from sequence data. MPRAs can assess thousands to millions of different sequences in a single experiment, and thus are powerful yet potentially expensive tools. In order to design more data efficient techniques for MPRAs, we explore an active learning procedure which utilizes uncertainty estimates to select subsequent sequences to test.

Active learning is the study of using machine learning models to select actions or make queries that influence the set of data they are trained on [Cohn et al., 1996]. This framework typically involves a

closed loop process where a model is trained on a small initial set of data and then subsequently used to identify a new data point (or points) from a potential pool of points to be labelled and added to its training set. The acquisition of a new point is typically based on a specified function known as an acquisition function.

In our work, we look to predict alternative RNA splicing behavior from DNA sequences using convolutional neural networks (CNNs) and Gaussian processes (GPs). We do so by utilizing Deep Kernel Learning (DKL) [Wilson et al., 2016] based models, which combine the useful learned representations and inductive biases of CNNs with the principled uncertainty estimates of GPs. To evaluate our models, we use the 5' splicing library from Rosenberg et al. [2015] as well as the active learning experiment setup from Nagpal and Knowles [2020]. Our results show that the GP based models outperform non GP based methods and that quality uncertainty estimates are indeed important to achieving good performance in an active learning setting.

Another approach of which we show promising preliminary results is the use of deep generative models to produce synthetic candidate sequences that you can either assay if you have a wet lab or run through an oracle model to quantify splicing outcomes and use as additional training data.

2 Methods

We use a baseline CNN model and experiment with different uncertainty estimation methods and acquisition functions in order to identify the best combination for our task. Figure 1 shows a high level depiction of our active learning experiment loop.

2.1 Uncertainty Estimation Methods

The base structure for our experiments is a CNN as they encode local translation invariance into useful learned representations of data. They also have been shown to be effective for applications in computational biology [Angermueller et al., 2016]. We follow the architecture from Nagpal and Knowles [2020] which stacks a 1 dimensional convolutional layer along with a max pooling layer as the basic unit. This basic unit is repeated twice and followed by two dense layers with rectified linear unit (ReLU) activations. Dropout is used before each dense layer.

The baseline for uncertainty estimation we use is Monte Carlo Dropout (MC Dropout) [Gal and Ghahramani, 2016] which is also used in Nagpal and Knowles [2020]. MC dropout reinterprets typical dropout training of a neural network as approximate Bayesian inference of deep GPs with a variational objective. Training the MC dropout version of the neural network is identical to training the vanilla version, the only difference is at test time dropout is kept on and repeated forward passes of the model with different dropout masks are interpreted as samples from the approximate posterior predictive distribution. These samples are then used to compute the mean prediction, the variance of predictions, and the covariance between predictions.

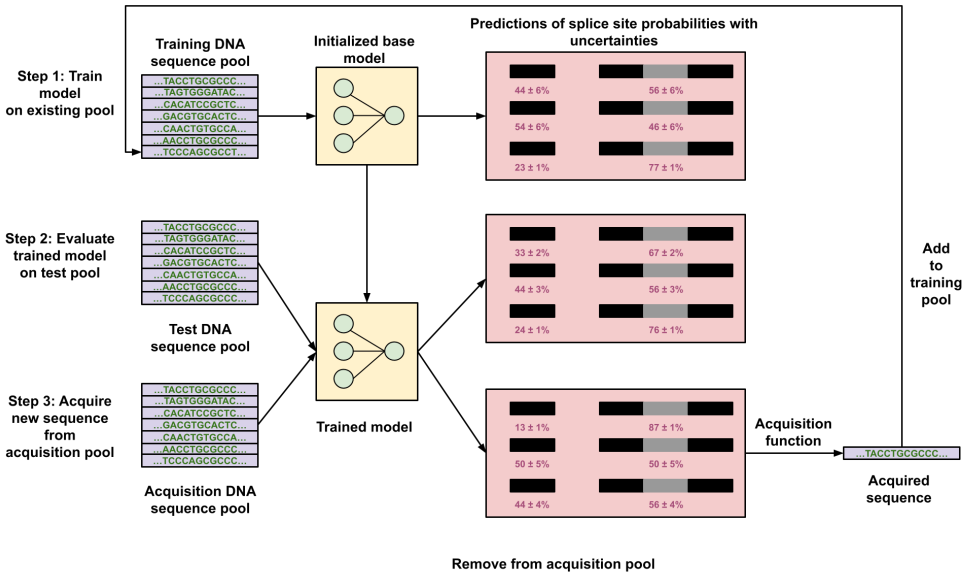

Figure 1: High level diagram of active learning experiments. For each active learning iteration, a model is trained from scratch on the training pool, evaluated on the test pool, and used to estimate the uncertainties of predictions on the acquisition pool. A specified acquisition function utilizes the uncertainties to choose a sequence to acquire, which is then removed from the acquisition pool and added to the training pool.

However, MC dropout is still approximation and its uncertainty estimates have been shown to vary in quality based on architecture and training procedures [Verdoja and Kyrki, 2020]. Recent work in scaling Gaussian processes to larger datasets [Wilson and Nickisch, 2015] along with combining neural networks with Gaussian processes [Wilson et al., 2016] have made it feasible to do exact Bayesian inference with models that use neural networks to learn the inputs to a kernel for a Gaussian process. These models combine the inductive biases and representation power of NNs with the principled uncertainty estimates of Gaussian processes, this approach is known as deep kernel learning (DKL) [Wilson et al., 2016]. Specifically in our work, we define our feature extractor to be the baseline CNN model and utilize a structured kernel interpolation (SKI) [Wilson and Nickisch, 2015] framework with a radial based function (RBF) base kernel. This allows us to train the system end to end with marginal likelihood, where inference and learning scale linearly with $O(n)$ time complexity for n training points.

2.2 Acquisition Functions

The simplest acquisition function that we use is the random function, where the next point to acquire is chosen randomly. Another acquisition function we experiment with is maximum variance, where point acquired is one with greatest estimated variance in its prediction. The largest variance would thus correspond to the largest uncertainty that the model has about the point. The final acquisition function we use in our experiments is the expected improvement based function from Nagpal and Knowles [2020]. Mathematically acquiring a point using this function can be represented as

$$x^* = \underset{x_{\text{new}} \in X_{\text{pool}}}{\arg\max} \operatorname{Tr}\left(\operatorname{Cov}_q(\hat{Y}_{\text{samp}}, \hat{y}_{\text{new}}) \operatorname{Var}_q(\hat{y}_{\text{new}})^{-1} \operatorname{Cov}_q(\hat{Y}_{\text{samp}}, \hat{y}_{\text{new}})^\top\right) \tag{1}$$

where x^* is the point to be acquired, X_{pool} is the set of points contained in acquisition pool, Cov_q is the covariance under our approximate posterior $q(\theta)$ with network parameters θ, $\hat{Y}_{\text{samp}}$ are the predictions corresponding to a sufficiently large random sample from the training and acquisition pools, $\hat{y}_{\text{new}}$ is the prediction corresponding the candidate sequence x_{new}, and Var_q is the variance under our approximate posterior $q(\theta)$.

2.3 Deep Exploration Networks

For our deep generative modeling approach, we explore using a series of pretrained generative models that are conditioned to generate synthetic sequences of a specific splicing isoform ratio, get the exact isoform ratio from an oracle model, and then augment our real training data with the generated sequences and their corresponding splicing isoform ratios.

The pretrained model that we're using to generate those sequences are deep exploration networks (DENs) [Linder et al., 2020]. The goal of DENs is to generate diverse sequences which maximize some predicted biological fitness score, in the paper the authors apply them towards the engineering alternative polyadenylation isoforms, differentially used splice sites, functional GFP (protein) variants, and transcriptionally active gene enhancer regions. The pretrained models we use have a predictor network that's been trained on a augmented version of the Rosenberg et al. [2015] dataset we've been using for our experiments.

The DEN architecture is based off of the deep convolutional generative adversarial networks (DCGAN) architecture [Radford et al., 2016]. DENs consist of a trainable generator network is trainable and a fixed pre-trained predictor network. The predictor network's job is to guide to generator network to maximize the fitness score from the loss (i.e. help the generator network generate sequences that match a given target isoform, etc.). The generator network generates position weight matrices that are then masked and templated so it can match whatever type of sequence you're looking to generate and sample from. The way the training procedure ensures diversity in the distribution of generated sequences is by adding a contrastive loss term to minimize the similarity between the generated sequences originating from two separate random seeds.

3 Experiments and Results

3.1 Data

For our experiments we use the 5' splicing library from Rosenberg et al. [2015] with processed data and preprocessing procedures from Nagpal and Knowles [2020]. DNA sequences with lengths

of 101 nucleotides are one hot encoded as inputs. Within this library, two 25 nucleotide regions of each sequence are randomized to be able to attribute differences in gene expression solely to variations in these randomized regions. The labels in this case are the modified probabilities of alternative splicing at one of two splice donor sites. The overall number of samples available is 265,137, with a heat map shown in Figure 2. Code for all data, experiments, and plots can be found at `https://github.com/tingtang2/active-learning-cnns-gps`.

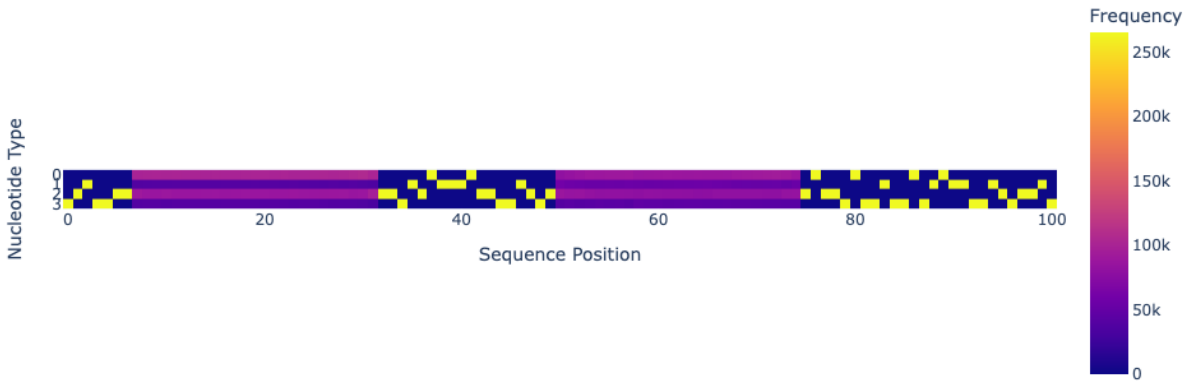

Figure 2: Heatmap of nucleotides within sequences in the library. 0 corresponds to A, 1 corresponds to C, 2 corresponds to G, and 3 corresponds to T. Notice how we can clearly see the randomized regions in purple.

3.2 Experiment setup and results

For our active learning experiments we follow the setup of Nagpal and Knowles [2020] where we train a base CNN model and use different acquisition point strategies to minimize mean squared error (MSE) on a held out test set. In addition to that set up, we experiment with the different uncertainty estimation methods described in Section 2.1 when training the base CNN model. All active learning experiment runs start with the same initial training set and as each new point is acquired from the acquisition pool, the model is retrained from scratch. The number of acquired points for each active learning experiment run is 600 and for each acquisition iteration, each model was trained for 300 epochs. The hyperparameters from Nagpal and Knowles [2020] were used for their models. All DKL based model results are based on training with noiseless targets, except for those using noisy targets for comparison in Section 3.4

For our deep generative modeling approach, in order to benchmark the addition of synthetic generated sequences for training, we used 3 different models. The first was the base CNN model which we used during the active learning experiments. The second and third models we used were adapted

from [Nair et al., 2019] which used a ResNet architecture and factorized convolutions, respectively. Both models are adaptations of the original Basset model [Kelley et al., 2016]. These papers were mainly concerned with predicting chromatin accessibility directly from sequence, but we've adapted them to predicting alternative splicing outcomes. In terms of the generator networks we combined the sequences and labels of 5 pretrained DENs targeting the following isoform ratios: 0, .25, .5, .75, 1. We then used the sequences and labels used as a form of training data augmentation for all of the the above models.

3.3 Comparison between methods

The overall results of our experiments are shown in Figure 3, where results of each configuration is the average across three random seeds.

Figure 3: Overall results of active learning experiments with random, maximum variance, and expected improvement (EI) based acquisition functions, along with MC dropout and DKL based uncertainty estimation methods.

From the results we can see that for both the maximum variance and the EI based acquisition functions, using DKL in place of MC dropout improved performance on the active learning task. The EI DKL version was able to mostly perform better than both random baselines, even for the last 100 acquired points, something the EI MC dropout version struggled to do. One interesting note is that for the first 100 acquired points, the maximum variance versions of MC dropout and DKL both seem to perform better than the EI versions.

We can see similar trends when looking at the number of acquisition points required in order to achieve a certain level of test MSE shown in Table 1. The DKL based models did better across all of the MSE levels when compared to the MC dropout models.

Table 1: Number of acquisition points required in order to achieve the specified 5' splicing test MSE for each method and acquisition function.

Method	Acquisition Function	Test MSE			
		0.11	0.10	0.095	0.09
MC Dropout	Random	80	180	306	387
	Max Variance	92	318	492	708
	Expected Improvement	69	109	174	369
Deep Kernel Learning	Random	56	125	177	**263**
	Max Variance	**13**	75	405	539
	Expected Improvement	53	**70**	**159**	301

3.4 Analysis of Deep Kernel Learning methods

The deep kernel learning methods were implemented in `GPyTorch` [Gardner et al., 2018], which allows for GPU acceleration of training and inference as well as different choices of methods for formulating and implementing deep kernels and GPs. One choice popular choice is to use Lanczos Variance Estimates (LOVE) [Pleiss et al., 2018] for faster computation of predictive variances and covariances. We experimented with using exact computations of predictive variance and covariance as well as using the LOVE method with the default maximum number of Lanczos iterations set to 100. The results are shown in Figure 4.

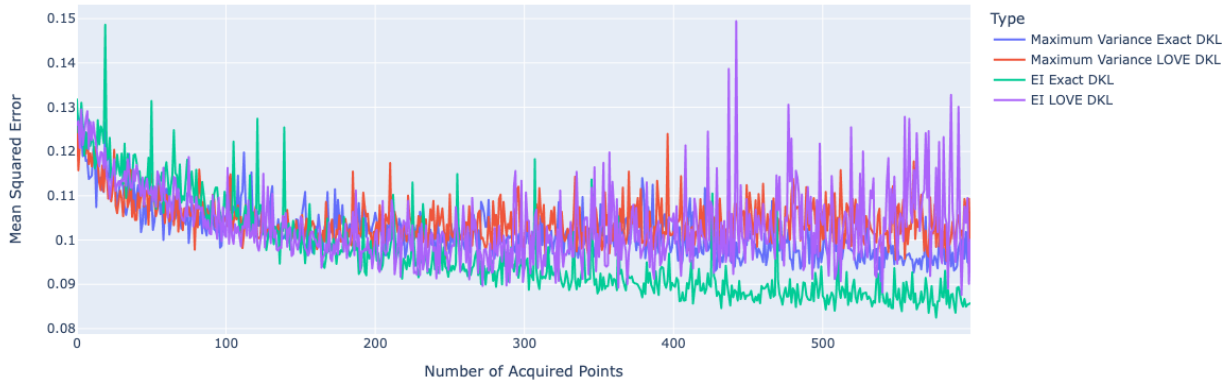

Figure 4: Comparison of LOVE vs. exact computations for GP predictive variances and covariances.

As we can see from the plot, the active learning runs using exact computations performed better than the ones using LOVE, especially for the EI based acquisition function where estimation of the predictive covariances are key. Interestingly, using LOVE with the EI based method resulted in large fluctuations in MSE towards the latter half of acquired data points.

Another choice we can make is between noisy and noiseless predictions. Typically GP regression is done with the assumption that the underlying distribution of targets is noisy, parameterized with a Gaussian likelihood. However, in Garriga-Alonso et al. [2019], their methods for GP regression are trained with noiseless targets. We experiment with both versions and present the results in Figure 5.

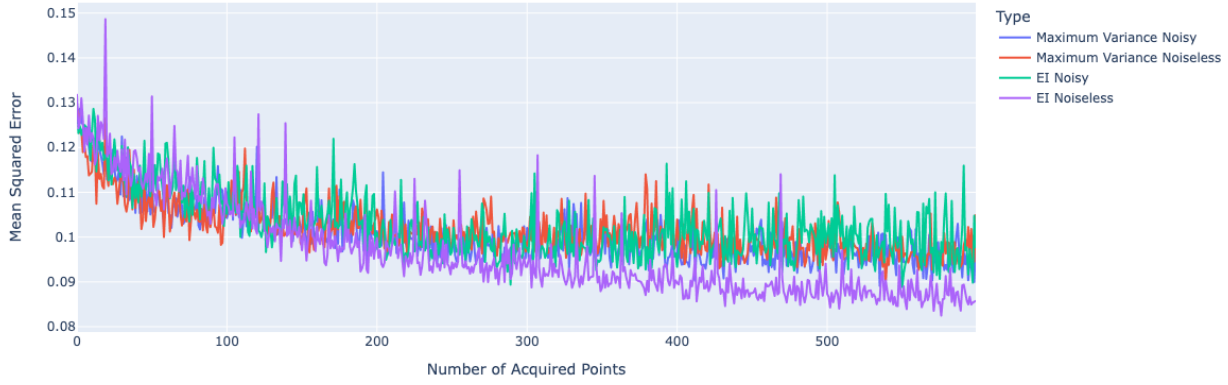

Figure 5: Comparison of noisy vs. noiseless predictions for DKL based models.

From the results we see that noisy and noiseless predictions for the maximum variance acquisition function are practically identical, while noiseless predictions for the EI based acquisition function are superior performance-wise. We hypothesize that this may be due to the flexibility of the underlying neural network.

3.5 Deep Exploration Network Results

Each model configuration was trained and evaluated independently across 3 random seeds, reported are the mean test metrics across the seeds. Across all models the augmentation does help a little bit in both MSE and Pearson, notably the augmented CNN performs comparatively to the non augmented ResNet and Deep Factorized model. Overall the improvements are pretty small though, but show some promise for continuing to look at developing methods.

Table 2: Test metrics across model configurations

Dataset	Method	MSE	Pearson
5' MPRA splicing	CNN	0.0386	0.856
	Augmented CNN	0.0377	0.860
	ResNet	0.0379	0.859
	Augmented ResNet	0.0373	0.861
	Deep Factorized Model	0.0376	0.860
	Augmented Deep Factorized Model	**0.0372**	**0.862**

4 Discussion and Conclusion

Our experiments confirm that quality of uncertainty estimation is important for active learning. They also present a way to use deep kernel learning based approaches for active learning and demonstrate their effectiveness in this setting.

The next steps for the active learning approach would be to implement the GP CNN models from Garriga-Alonso et al. [2019] for use in this experiment as well as to repeat these experiments across other MPRA datasets from Bogard et al. [2019] and Sample et al. [2018].

For the generative model approach an obvious follow up would be to fine tune Fine-tuning the pretrained DEN generator models and also incorporate some meta learning methods to better utilize the synthetic data, in particular conditional neural processes [Garnelo et al., 2018].

References

Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7):878, 2016. doi: https://doi.org/10.15252/msb.20156651. URL https://www.embopress.org/doi/abs/10.15252/msb.20156651.

Nicholas Bogard, Johannes Linder, Alexander B. Rosenberg, and Georg Seelig. A deep neural network for predicting and engineering alternative polyadenylation. *Cell*, 178(1):91–106.e23, 2019. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2019.04.046. URL https://www.sciencedirect.com/science/article/pii/S0092867419304982.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. 1996. doi: 10.48550/ARXIV.CS/9603104. URL https://arxiv.org/abs/cs/9603104.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In S. Bengio, H. Wallach,

H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/27e8e17134dd7083b050476733207ea1-Paper.pdf.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/garnelo18a.html.

Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bklfsi0cKm.

David R. Kelley, Jasper Snoek, and John L. Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 26(7): 990–999, July 2016. ISSN 1088-9051, 1549-5469. doi: 10.1101/gr.200535.115. URL http://genome.cshlp.org/lookup/doi/10.1101/gr.200535.115.

Johannes Linder, Nicholas Bogard, Alexander B. Rosenberg, and Georg Seelig. A Generative Neural Network for Maximizing Fitness and Diversity of Synthetic DNA and Protein Sequences. *Cell Systems*, 11(1):49–62.e16, July 2020. ISSN 24054712. doi: 10.1016/j.cels.2020.05.007. URL https://linkinghub.elsevier.com/retrieve/pii/S2405471220301927.

Udai G. Nagpal and David A Knowles. Active learning in cnns via expected improvement maximization, 2020. URL https://arxiv.org/abs/2011.14015.

Surag Nair, Daniel S Kim, Jacob Perricone, and Anshul Kundaje. Integrating regulatory DNA sequence and gene expression to predict genome-wide chromatin accessibility across cellular contexts. *Bioinformatics*, 35(14):i108–i116, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz352. URL https://doi.org/10.1093/bioinformatics/btz352.

Geoff Pleiss, Jacob Gardner, Kilian Weinberger, and Andrew Gordon Wilson. Constant-time predictive distributions for Gaussian processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4114–4123. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/pleiss18a.html.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06434.

Alexander B. Rosenberg, Rupali P. Patwardhan, Jay Shendure, and Georg Seelig. Learning the sequence determinants of alternative splicing from millions of random sequences. *Cell*, 163 (3):698–711, 2015. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2015.09.054. URL https://www.sciencedirect.com/science/article/pii/S0092867415012714.

Paul J. Sample, Ban Wang, David W. Reid, Vlad Presnyak, Iain McFadyen, David R. Morris, and Georg Seelig. Human 5' utr design and variant effect prediction from a massively parallel translation assay. *bioRxiv*, 2018. doi: 10.1101/310375. URL https://www.biorxiv.org/content/early/2018/04/29/310375.

Francesco Verdoja and Ville Kyrki. Notes on the behavior of mc dropout, 2020. URL https://arxiv.org/abs/2008.02627.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1775–1784. JMLR.org, 2015.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/wilson16.html.