Gaussian Process Regression and Bayesian Optimization on Latent Embedding of Protein Sequences

Xinran Lian xlian@uchicago.edu
Department of Chemistry
University of Chicago
Chicago, IL 60637, USA

March 11, 2023

1 Introduction

Latent-space based deep-learning generative models represent an exciting direction in understanding and designing proteins. Locality in the low-dimensional (e.g. d=3) latent space represents the function of protein homologs, making it controllable for designing specific functions. In our previous work [Lian et al., 2022], we trained an InfoMax Variational Autoencoder (InfoVAE), randomly sampled and decoded the latent embedding from an anisotropic Gaussian distribution estimated by the functional sequences embedded in the 3D latent space. Novel sequences designed by this means resulted in a 44% functionality compared to 2.5% in the natural training set. Both natural and designed functional sequences clustered tightly in the latent space and designed sequences extended the function niche. Proven by these findings together with the previous work [Ding et al., 2019], the latent space of VAEs can be regarded, at least to some degree, as a low-dimensional projection of the protein fitness landscape.

The VAE latent space has shown potential in exploring and extending function constraints of protein sequences in an unsupervised manner. To improve the sampling accuracy and efficiency, it would be beneficial to incorporate supervised methods to detect the ability of extension before experimentally testing. We aim to filter potential functional sequences by supervising the fitness score of training protein data. In this regard, Gaussian Processes (GP) and Support Vector Regression (SVR) are promising methods we would like to test for this task. Furthermore, we aim to test Bayesian optimization (BO) based on the GP model for identifying good locations to sample in the latent space.

2 Related work

My coworker N. Praljak developed a semi-supervised VAE model which forces the functional sequence cluster to move towards to a direction of the latent space [Praljak and Ferguson, 2022], creating an artificial gradient for the functional latent embedding to sample new sequences from. This method however reduces generalizability of the latent space by conditioning it on score of a specific function, because the training data is composed of proteins with diverse but valid structures and functions.

With the latent space structure kept as the original unsupervised model, one simple method is to use Bayesian Optimization (BO) to query the latent space. BO has been used for VAEs for automatic chemical design of small moleculars [Griffiths and Hernández-Lobato, 2020].

Another approach is Level Set Estimation (LSE), an algorithm to find the threshold level that is implicitly defined as a percentage of the (unknown) maximum of the target function space [Gotovos, 2013]. Gotovos et al. used LSE to identify a threshold of chlorophyll concentration on a vertical transect plane of Lake Zurich to monitor the lake environment for algal bloom. They modeled the chlorophyll concentration data as an unknown function modeled as a sample from a GP. The algorithm TRUVAR [Bogunovic et al., 2016] was presented later to treats BO and level-set estimation (LSE) with GPs in a unified fashion.

3 Research question and approach

The research consists of two parts: firstly, to employ GPR and SVR for predicting the functions of designed protein sequences and filtering sequences with an expected function score greater than 0.5. Secondly, to utilize BO to query the latent space with the GPR model, and sample latent variables with potential of generating functional sequences.

The latent space model to be used is InfoVAE trained on 5299 natural SH3 protein sequences. Both the model and the dataset are described in [Lian et al., 2022]. The dataset for function prediction and BO, which is a subset of our whole dataset, is composed of 5090 natural + 2855 InfoVAE = 7945 sequences with measured function score (relative enrichment). The dataset is available in the GitHub repository [Lian, 2022]. Here, our observation is the latent variable z of the sequences and their normalized functional score y ranging approximately from -0.5 to 1.5. Measurement and calculation of y are described in our paper.

3.1 Supervised-learning the latent space with GP and SVR

The SingleTaskGP model is implemented in gpytorch [Gardner et al., 2018]. Codes are shown in Listing 1. Prior of the likelyhood is a simple Gaussian prior $\mathcal{N}(0, 1)$ with constraints greater than 10^{-4}. The covariance module is with $\nu = 0.5$ and a Gamma lengthscare prior $\Gamma(2, 0.5)$ and constraints greater than 10^{-3}, and subjected to a output scare prior $\Gamma(2, 0.1)$ SVR is implemented in scikit-learn [Pedregosa et al., 2011] using default settings with the RBF kernel. Both models was trained on 80% of natural sequences with 3 folds of cross validation.

The matrix for evaluation is the F1 score where positive is defined as $y > 0.5$.

3.2 Address the optima in the latent space with BO

The goal is to optimize the sampling efficiency for proteins sequences with a targeted function (e.g. osmosensing) from the VAE latent space incoorperating supervised approaches or BO. The query strategy is described in Algorithm 1. The training data, which is the latent variables z, are splitted into $k = 25$ parts. 1 BO sample $(\hat{z}_i, \hat{y}_i)$ is queried by optimizing the Upper Confidence Bound (UCB) acquisition function with $\beta = 0.2$. The latent variable $\hat{z}_i$ is accepted if the predicted $\hat{y}_i > 0.5$. If accepted, $\hat{z}_i$ is decoded into amino acid sequences (assumed for wet-lab experiments). As a practice, I used AlphaFold [Jumper et al., 2021] in the ColabFold [Mirdita et al., 2022] notebook to predict whether corresponding decoded proteins can fold to the structure to bind with the target ligand, and to compare with the published protein structure. For ColabFold prediction, I used the default settings.

Algorithm 1 BO sampling in the InfoVAE latent space

Inputs: $z \in R^{N \times d}$, $y \in R^N$, k
$D^{new} = \emptyset$
for $i = 1...k$ **do**
    $D_i = \{(z_{(i-1)N/k}, y_{(i-1)N/k}), ...(z_{iN/k}, y_{iN/k})\}$
    Fit SingleTaskGP function f on D_i
    $\hat{z}_i = \arg\max_x UCB(x|f, D_k)$
    $\hat{y}_i = f(\hat{z}_i)$
    if $\hat{y}_i > 0.5$: $D^{new} \cup (\hat{z}_i, \hat{y}_i)$
    end for.
Output : D^{new}

Predictor	F1 (natural)	F1 (design)
GP	0.905 $\pm$0.005	0.806 $\pm$0.001
SVR	0.847 $\pm$0.009	0.819 $\pm$0.004

Table 1: F1 scores for GP and SVR prediction

4 Experiments and results

4.1 GPR and SVR prediction

The F1 score for GPR and SVR on the natural and designed library are shown in Table 1. Prediction of one trial is visualized in Fig. 1. GP performs better than SVR on the natural dataset, while on the designed dataset which was mot used for training, SVR was slightly better. Both models was robust for cross validation.

Considering the objective of our prediction: to identify sequences with $y_{pred} > 0.5$ for the expensive wet lab experiments, the recall score of the designed etst set is also a matrix of evaluating the two methods. The recall score of GPR is 0.984 ± 0.003, while for SVR is 0.926 ± 0.012. Consisting with Fig. 1, GPR is expected to miss less functional designed sequences for experiments than SVR in this case.

4.2 BO querying latent space

21 queried latent points ($\hat{z}_i$) were accepted from a single experiment and were decoded to protein sequences, with an acceptation rate of 84%. Fig. 2 shows the location of accepted sampled latent variables are consist with reported functional sequences in our paper.

4.3 AlphaFold prediction of sampled sequences

I picked two BO designed sequences with highest novelty (lowest closest identity to the training set) for AlphaFold prediction. The predicted structures compared to the published structure of 2VKN (the wild-type functional SH3 sequence) are shown in Fig. 3.

The designed sequences exhibit moderate ($\sim$ 80%) identity to the closest natural alleles, while showing low ($\sim$ 50%) identity to 2VKN. The loop of the ligand binding site of sequence A appears slightly diverged compared to 2VKN and B. However, it is noteworthy that the loop of both sequences is in close proximity to the peptide ligand, indicating a functional ligand-binding capability. Overall, this preliminary analysis suggests that the designed sequences

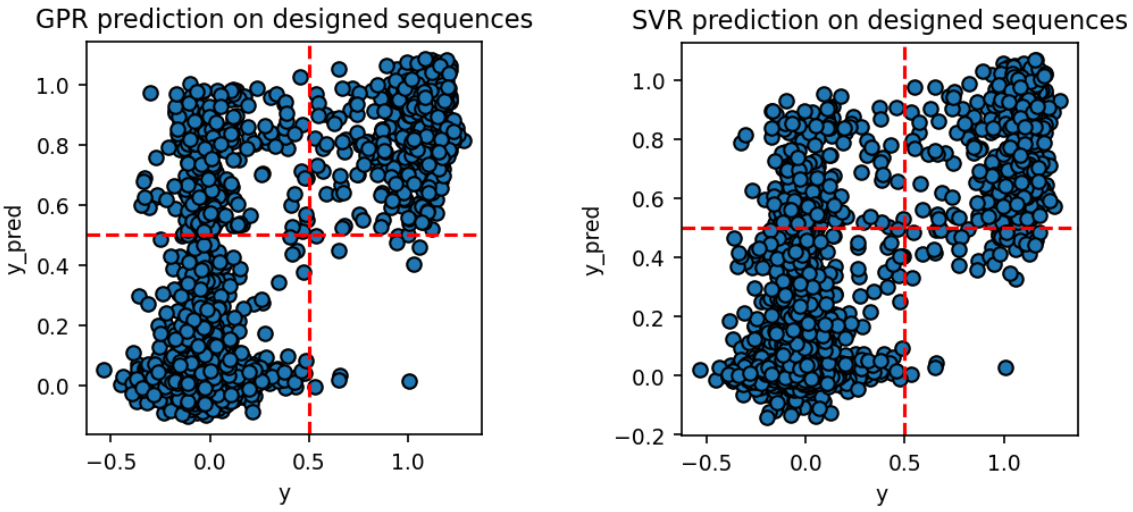

Figure 1: Visualization of one trial of GPR and SVR prediction on designed sequences. Models are trained on 80% of natural sequences. Red dash lines indicate the grid of 0.5.

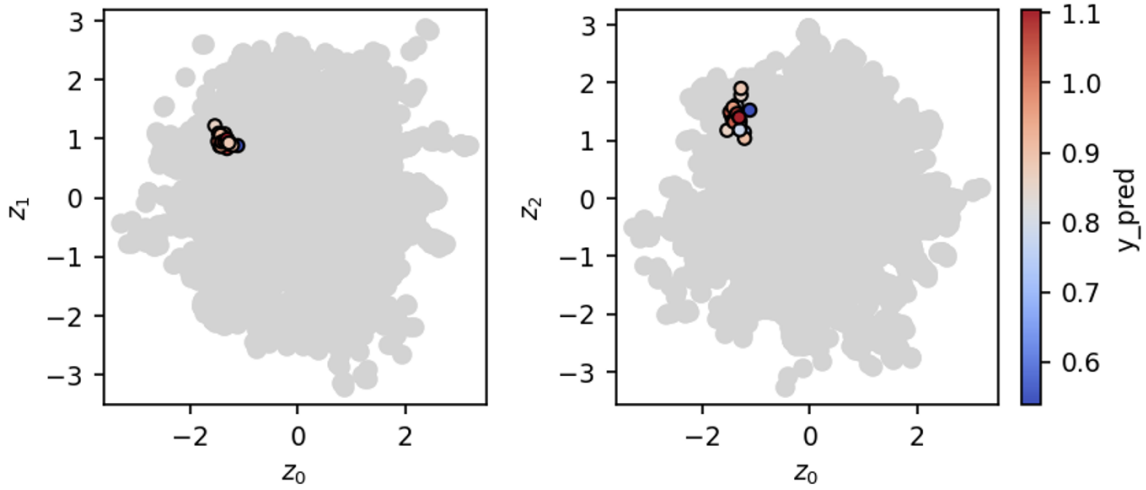

Figure 2: Distribution of BO sampled latent variables with y_{pred}

	Sequence	closest ID	ID 2VKN
A	EYPYRAKALYSYQANPDDPNEISFTKGEVLDISDNSGRWWQARKADGETGIAPSNYLQLL	83%	56%
B	EYPYRAKALYAYQADDDDPNEISFAKGEVLDISDNSGKWWQARKADGETGIAPSNYLQLI	85%	59%

Table 2: Sequences with closest and 2VKN (the wild-type functional protein in the yeast host) identities of the two BO designed protein candidates for AlphaFold prediction.

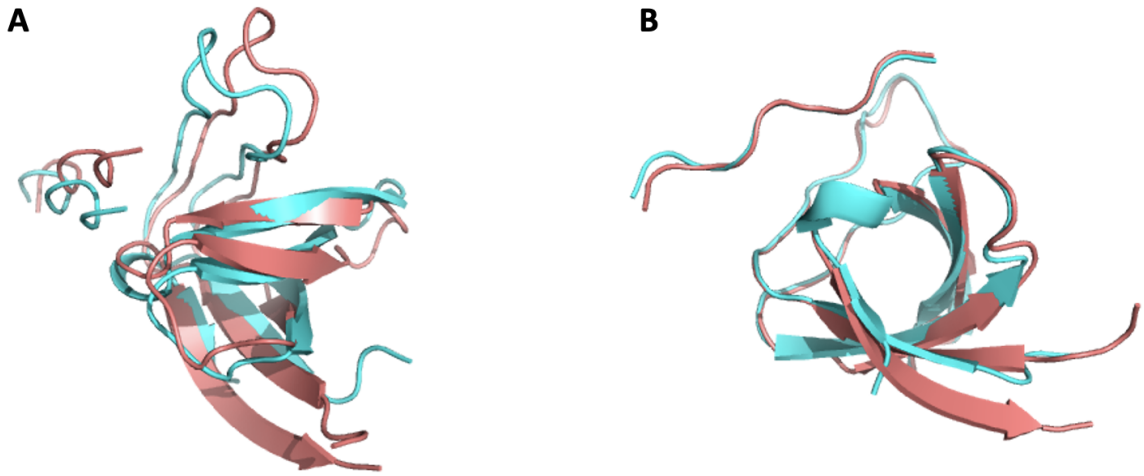

Figure 3: Comparison of the AlphaFold-predicted structures of designed sequences by BO sample in the latent space (cyan) with the published structure of 2VKN (red).

fold into a comparable structure with the wild-type. This is a known property of a protein homologous family, where members can conserve the structure while exhibiting sequence diversity.

5 Discussion and conclusion

In conclusion, this study aimed to compare the performance of GPR and SVR in identifying potential functional designed protein sequences based on latent variables and functional score (y) of the training set. Our results suggest that SVR is slightly better than GPR in terms of F1 score for predicting the test set (designed library). However, GPR demonstrates a recall score close to 1, which indicates that it can almost perfectly select all potential functional sequences. Since GPR itself does a good job, I did not consider using more sophisticated LSE. Additionally, we used BO to query the latent space and identified sequences that exhibit moderate diversity but are predicted to be functional by AlphaFold.

The findings of this study highlight the potential of incorporating supervised machine learning for protein design using deep-learning generative models. This approach can help identify potentially functional sequences or directly improve the sampling process. However, one limitation of this simple BO study is that the queried latent variables are conserved, only occupying a small location inside the functional cluster (Fig. 2). This could be caused by the large number of data points for BO due to the small proportion of functional sequences. While I tried the qUCB function in BorTorch to query multiple latent variables in one trial, the resulting acceptation rate was reduced a lot based on the $y_{pred} > 0.5$ criteria.

To address these limitations, an iterative design approach could be implemented by training

the model further with designed sequences that have been experimentally tested in the wet lab to have functional scores. Another way to improve the approach is to use a higher-dimensional latent space, which could increase the sparsity of the latent space and potentially enable finer-tuning, since the latent "volume" increases exponentially with the dimension. Additionally, advanced BO methods such as TurBO could be used to query the high-dimensional latent space. These enhancements could enable the identification of functional protein sequences with better diversity.

6 Supplementary Information

Listing 1: Codes to define the GP function

```
import gpytorch

noise_prior = gpytorch.priors.NormalPrior(0,1) # simple prior
lengthscale_prior = gpytorch.priors.GammaPrior(2, 0.5)
outputscale_prior = gpytorch.priors.GammaPrior(2, 0.1)

likelihood = gpytorch.likelihoods.GaussianLikelihood(
    noise_prior = noise_prior,
    noise_contraint = gpytorch.constraints.GreaterThan(1e-4)
    )

covar_module = gpytorch.kernels.MaternKernel(
    nu = 2.5,
    lengthscale_prior = lengthscale_prior,
    lengthscale_constraint = gpytorch.constraints.GreaterThan(1e-3)
    )

covar_module = gpytorch.kernels.ScaleKernel(
    covar_module,
    outputscale_prior = outputscale_prior
    )

gp = SingleTaskGP(
                  X_train,
                  Y_train,
                  likelihood,
                  covar_module = covar_module)
mll = gpytorch.mlls.ExactMarginalLogLikelihood(gp.likelihood, gp)
```

References

[Bogunovic et al., 2016] Bogunovic, I., Scarlett, J., Krause, A., and Cevher, V. (2016). Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation. *Advances in neural information processing systems*, 29.

[Ding et al., 2019] Ding, X., Zou, Z., and Brooks III, C. L. (2019). Deciphering protein evolution and fitness landscapes with latent space models. *Nature communications*, 10(1):5644.

[Gardner et al., 2018] Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31.

[Gotovos, 2013] Gotovos, A. (2013). Active learning for level set estimation. Master's thesis, Eidgenössische Technische Hochschule Zürich, Department of Computer Science,.

[Griffiths and Hernández-Lobato, 2020] Griffiths, R.-R. and Hernández-Lobato, J. M. (2020). Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2):577–586.

[Jumper et al., 2021] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

[Lian, 2022] Lian, X. (2022). Protein design mmdvae torch. https://github.com/Ferg-Lab/Protein_design_mmdVAE_torch.

[Lian et al., 2022] Lian, X., Praljak, N., Subramanian, S. K., Wasinger, S., Ranganathan, R., and Ferguson, A. L. (2022). Deep learning-enabled design of synthetic orthologs of a signaling protein. *bioRxiv*.

[Mirdita et al., 2022] Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., and Steinegger, M. (2022). Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Praljak and Ferguson, 2022] Praljak, N. and Ferguson, A. (2022). Auto-regressive wavenet variational autoencoders for alignment-free generative protein design and fitness prediction. In *ICLR2022 Machine Learning for Drug Discovery*.