**Review of Machine Learning models for protein sequence-function mapping**

**Abstract**

With advances in modern deep learning, research at the interface between computer science and biology has seen unprecedented improvements. Especially, in the field of protein biology, new methods are developed to map the protein sequence-structure-function relationships. Understanding the relationship between proteins’ primary sequence and its function helps to design proteins and enzymes with desired properties in fields such as biochemistry and biotechnology. In contrast to traditional rule-based methods, machine learning (ML) methods can automatically discover complex patterns and relationships in the data, and they can handle large and complex datasets more effectively than traditional methods. In addition, ML methods can be updated and improved as new data becomes available, whereas traditional methods are typically fixed and cannot be easily modified. These advantages make ML methods particularly well-suited for protein sequence to function mapping, where the data is often large and complex, and the relationships between protein sequence and function are not well understood. Here, we review ML methods that can predict protein functions when provided with sequential information. We categorize these methods into two broad categories, supervised, and unsupervised methods. These methods are differentiated by the type of training data used for training these models. Along with recent progress, we discuss their strength and weaknesses.

**Introduction**

Creating proteins with desired functional properties has been an important problem in biology and drug discovery as these proteins can be used for biochemistry, biotechnology and pharmaceuticals. As new proteins are created using molecular evolution, deciphering sequence to function (S-F) mapping leads to the acceleration of proteins and enzymes design with desired functional properties by identifying useful variants. While the function of a protein depends on the biochemical process it is a part of, and hence, is hard to measure, protein’s fitness values such as growth rate can be used as a measure of protein’s function, as fitness depends on protein’s shape and the presence of any mutation. Additionally, S-F models can be useful for studying the evolution of proteins and for predicting the effects of genetic variations on protein function. Through high-throughput sequencing methods such as deep mutational scanning (DMS), it is possible to create a variant library of proteins of interest and use molecular evolution to measure the function of interest [1]. Iteratively discarding bad mutations and selecting high-fitness mutations can lead to proteins with the desired function [2]. But, since the sequence space for proteins is very high, for example, there are 5,700 single-amino-acid substitutions for a 300-amino-acid protein and 32,381,700 possibilities to create just two substitutions with the 20 canonical amino acids, high throughput methods are limited by their capacity to map the complete protein sequence space to their fitness [2]. With all the possible permutations/possibilities this creates a disadvantage in exploring the complete protein sequence-function landscape.

Also, DMS experiments always require at least one protein with the required protein function as a starting point through which other variants can be generated [2]. This means that using DMS alone can give a very low sampled view of the high-dimensional protein sequence landscape [2]. Low order models such as potts-model and additive models can be used to map sequence-functions to fill in the gaps created by DMS [3]. But since these low order models cannot model epistatic interactions beyond point-wise correlations, their accuracy is limited [4]. Machine learning (ML) methods try to mitigate low sampling issues. ML methods can be trained on data from high-throughput experiments and be further used to infer functional properties of proteins which occur in nature or are artificial. ML methods also generalize well to sequences that were not sampled through DMS experiments [2]. Standard ML algorithms work well when the data is limited in numbers, but deep learning algorithms are used to learn the relationship between amino acids in a protein sequence and use this relationship to create a protein sequence landscape. Deep learning methods also do not always require labeled data, this means that instead of using sequence-function data generated by DMS, sequence data alone can be used to predict pseudo-fitness values, which are highly correlated to actual fitness values of the proteins [4]. While deep learning models do not always require expensive labelled data, there is currently no consensus on whether deep learning models outperform simpler machine learning approaches when it comes to molecular property prediction [32-34].

This review covers various machine learning methods used to learn sequence-function maps in proteins. The paper is divided into two broad categories – supervised and unsupervised learning methods. These methods are described in terms of the data used for training, the training process, and their advantages and disadvantages.

**Supervised Learning**

A supervised machine learning paradigm refers to a training method that uses labeled data to train a ML model. A labeled dataset contains data samples with their corresponding values or class labels. For protein sequence-function mapping, this means that protein sequences with known functions are used to train ML models. In this paradigm, data is divided into three sets, namely, training data, validation data, and test data in the ratio of 70-10-20 [2]. ML models require hyperparameters to be defined. In machine learning, hyperparameters are parameters that are not learned from the data but are set by the practitioner. These parameters can have a significant impact on the performance of a machine learning model, and so they must be carefully chosen in order to obtain the best possible results. Examples of hyperparameters include the learning rate and the regularization coefficient in a neural network, the number of trees in a random forest, and the kernel function and regularization parameter in a support vector machine. Hyperparameter optimization is run on validation data so that the model training can converge at the highest possible accuracy. Optimizing hyperparameters is an important step in the process of developing a machine learning model, and it can be done using a variety of techniques, such as grid search, random search, and Bayesian optimization. These methods allow the practitioner to explore the space of possible hyperparameter values and identify the settings that result in the best performance on the task at hand. Once hyperparameters for the model are set, the model is trained on training data. After the training process, the ML model is tested on the test data. The accuracy calculated on the test data is called the model’s accuracy. In the case of proteins, this data generally comes from high-throughput experiments that iteratively find protein sequence-function mapping (fig 1) [5]. Due to the high cost and low efficiency of the generated data, these ML methods are trained on limited classes of proteins [2]. Further, these methods are hard to generalize across protein families due to the data-generating process and as such, necessitate a large set of sequences for training [4].

A picture containing application

Description automatically generated

Fig 1 – A typical Deep Mutational Scanning workflow (From Hanning et al, 2022 [5])

Since all the ML algorithms are numerical functions, the input sequence needs to be transformed from string to numbers before any training can take place. A protein sequence is a string of characters of length L. Each of these characters represents one of the 21 amino acids. Since ML models are trained on numerical data, there is a need to transform the discrete protein sequence into a continuous value. The process of representing strings as numbers is called vectorization [6]. One of the simplest ways to vectorize a protein sequence is to assign an arbitrary number to each of the amino acids. But since ML models are sensitive to the magnitude of input features, assigning arbitrary number to amino acids creates a constraint on the ML models. To overcome the constraint, amino acids are represented by one-hot encoded numbers, wherein each amino acid is converted into a binary vector of length 21. Although one hot encoding is a good baseline vectorization method, they are sparse and high-dimensional. Instead of using amino acid sequences directly, proteins can also be represented based on their physical properties such as charge, volume, and hydrophobicity. AAIndex [7] and ProFET [8] contain multiple descriptors of amino acids, which can be used to substitute amino acid strings with numbers. Although these methods can convert strings into vectors, there is no pre-supposition that similar protein sequences will have a low Euclidian distance as compared to completely unrelated proteins. With advances in Natural Language Processing (NLP) methods, techniques called word2vec [9] are used to convert protein sequences to embedding vectors, which maintains Euclidean distance between proteins proportional to the similarity between them. These representations are learned from the data as opposed to methods such as physical descriptors. Learned representations are generated by training models to encode the sequence of proteins in an unsupervised fashion. These representations have the potential to capture more information than one-hot encoded representations or physical properties of amino acids. Further, these representations capture contextual information. With these embeddings, generating the sequence-function map is a downstream task, which can be learned using the supervised learning paradigm [4]. Training of these embeddings is covered under the unsupervised learning heading.

Once sequences are converted to vectors, ML methods can be used to map sequences and functions. For these methods, the training aims to generate a model that can be used to predict protein function values for proteins that were not present in the training set, i.e., the model should be generalizable to previously unseen proteins. A good starting point to learn the relation between sequence and function is to use a linear model. Linear models are good baseline methods to model the relationship as linear models provide a simple, interpretable baseline. But since linear models do not consider any epistatic interactions unless these interactions are pre-modeled in vector embeddings, other non-linear models are applied to further improve the accuracy and model interactions between amino acids in the sequence [4]. Regression trees [10] can give deterministic, computationally effective, and interpretable models. These are especially useful when the number of training samples < 104. To further increase the accuracy at the cost of interpretability, techniques called boosting and bagging are used where multiple models are trained in a sequential or paralleled fashion, and final predictions are obtained by combining predictions of each of the individual models. The process of using multiple models, also called ensembles, increase the accuracy by reducing variance of the ML model. Ensembles of trees are implemented in random forest algorithm or boosted trees [11]. Ensemble methods have been used to predict physical properties such as enzyme thermostability. Kernel methods such as Support Vector Machines (SVM) [12] can be also used to predict protein function from a sequence. Kernel methods implicitly find the relationship between data points and project them into a high-dimension implicit feature space using the kernel trick, without explicitly calculating the coordinates in the new feature space. Gaussian processes (GP) [13] combine the kernel method with Bayesian learning to provide probabilistic predictions while providing estimates of uncertainties. SVM and GP have been used to predict thermostability [14], fluorescence [15], and membrane protein expression and localization [16]. If data samples are more than 104, deep learning methods can be used to map sequence-function relationships. Deep learning methods stack linear layers with non-linear activation functions one after the other, which helps these methods to learn non-linear relationships 2]. They have been used to predict protein-nucleic acid binding [17], protein-ligand binding [18], and protein thermostability [19]. Supervised learning methods are limited in their accuracy and generalizability since experiments such as DMS only can generate limited data. These limitations are mitigated by unsupervised learning methods, which do not need a set of labeled data to be trained on.

**Unsupervised Learning**

Unsupervised learning is a machine learning paradigm to train models without any existing labeled data. Multiple databases consist of protein sequence information for various organisms. Although these sequences are majorly unlabeled, protein databases are a huge untapped resource for learning about proteins and their distribution across organisms. For protein biology, there are two mainstream methods to learn from unlabeled data: 1) modeling of sequence distribution, and 2) extracting representations.

The first method, modeling of sequence distribution, involves finding Multiple Sequence Alignments (MSA) of the protein of interest and probabilistically modeling the sequence distribution. A multiple sequence alignment contains evolutionary information about the protein and provides a set of weak positive labels for all the proteins in the alignment. MSA also intrinsically represents the conserved domains of a protein since during evolution, a protein is functionally conserved [4]. The domain of a protein that maintains its function has a higher alignment score with the rest of the proteins. These domains represent the protein core or functional sites which were conserved during evolution. Hence, a previously unseen protein that consists of a conserved domain has a high probability of retaining the functionality of its domain [4]. The earliest method that models the probability distribution from an MSA is called the Position Weight Matrix, or Position-Specific Scoring Matrix (PSSM), which expresses motifs (patterns) in biological sequences [20]. PSSM is generated by using MSA to calculate the probability by giving higher weight to domains that were conserved as compared to variable domains. A protein of interest is aligned to a database such as Uniclust [21], Uniref [22], or SwissProt [23] using an MSA program such as JackHMMer [24], which generates sequence alignments of the protein of interest with the rest of the proteins in the database. Constraints such as percentage similarity between sequences are used to find protein homologs. The homologs are assumed to be enriched for the same property as that of the query sequence. Once the MSA is generated, a probability density model of this set of proteins is calculated [4]. For example, in SIFT [25], the MSA is used to calculate the probability of amino acid substitutions and maps these probabilities to protein function. But methods based on PSSM presume the independence assumption. These models assume that the amino acids mutate independently, and their evolution is not impacted by the rest of the amino acids present in the protein. This limitation is mitigated by the Potts model [3] where the log probability depends on individual amino acid’s frequency, plus, a pairwise term. The pairwise term depends on the joint probability of two amino acids occurring together. Pairwise term models the epistatic interactions between amino acids. EVMutation [26] is a potts equation-based model that takes into consideration the epistatic interactions between amino acids. The log-likelihood of mutation with wild-type is calculated, which correlates to the protein fitness and represents the affinity of a mutation towards the protein’s fitness. Since the potts model can only model pairwise interactions [4], there is a need for models that can capture higher-order epistasis. Extension of pairwise model to third or fourth-order interaction is infeasible as even third-order interaction models for a protein of 100 amino acids will have approximately 1 billion parameters, which makes this extension computationally infeasible. Since deep learning models can learn approximate functions from available data, deep learning methods are used to infer higher-order epistatic interactions. Deepsequence [27], an MSA-based Variational Autoencoder neural network model approximates high-order epistatic interaction as a non-linear latent hidden variable, which it learns from the MSA input. The major drawback of MSA-based models is that they do not predict the exact protein function, rather, they can only rank the protein and its homologs based on their functional order. Further, the decision of which protein sequences to include in an MSA is ad-hoc, which limits the quality of the probability distribution.

The second method, extracting representations, involves working with only a single sequence and hence these methods do not suffer from the disadvantages that MSA based methods go through. In representations-based method, vector embeddings of a protein sequence are calculated. The methods to generate these embeddings are largely inspired by NLP methods such as word2vec [6]. The vector embeddings should be biologically significant. For example, when embeddings generated by ProtTrans [35] are dimensionally reduced using t-SNE, it was found that the embeddings captured biologically significant amino acid properties such as charge, polarity and hydrophobicity. The procedure to generate embeddings is called pre-training. In presence of labeled data, the pre-trained model can be further fine-tuned on a downstream task such as protein function prediction. Pre-trained models can also generate embeddings which can be further used as input for training supervised learning models using the process of fine-tuning. Representation generating methods take into consideration the context around each amino acid i.e., the embeddings model the epistatic interactions between amino acids in the sequence. In absence of labeled data, these models act as zero-shot predictors [28]. Under the zero-shot paradigm, pre-training allows protein language models to learn the information needed to solve a task, and then they can be applied effectively to new instances of the task without specialization. This means that a single general-purpose model can be trained once and then applied to a wide range of applications. There are three ways to learn embeddings using unsupervised learning (fig 2): 1) Word2Vec 2) Neural autoregressive language models and 3) Neural masked language model

Graphical user interface, diagram

Description automatically generated

Fig 2 – Embedding generation from protein sequence [31]

*Hidden states from the following three methods are extracted to generate embedding vector - 1) Word2vec is a method for generating continuous-valued word embeddings, which are vector representations of words that capture their semantic meaning and relationship to other words in a corpus. 2) A neural autoregressive language model is a type of language model that uses a neural network to predict the next word in a sequence based on the previous words in the sequence, using an autoregressive approach to modeling the dependencies between the words.  
 3) A neural masked language model is a type of language model that uses a neural network to predict the correct word in a sentence given the surrounding context, where some of the words in the sentence have been "masked" and replaced with a placeholder token.*

All these three ways provide embeddings for a single protein sequence. One of the first methods specific to protein embedding generation was Unirep[29]. The Unirep model was trained on 24M uniref50 amino acid sequences. The model was trained to perform the next amino acid prediction task in left-to-right order. This pretraining task helps the model to learn the association between previously seen amino acids in a sequence, and then use these amino acids to predict the next amino acid. Unirep was based on a Long Short-Term Memory (LSTM) deep learning architecture. The embeddings for downstream tasks are collected from the hidden state of the LSTM. But LSTM models contain many parameters, are non-interpretable and non-parallelable to train. These shortcomings are mitigated by transformer based neural networks [4]. With the rise in transformer architecture of deep learning, pre-training tasks such as masked amino acid prediction are used to generate embeddings of protein sequences using an attention model [30]. Unlike Unirep, transformer-based encoders approach masked amino acid prediction as a bidirectional problem. Trained on large, unannotated databases of protein sequences such as UniClust[21], protein language models aim at reconstructing a sequence from a corrupted version with ~10-20% of the residues masked or randomly mutated. The attention maps generated by transformers during pre-training correspond to structural properties such as protein contact maps, i.e., pretrained transformers can recapitulate the biophysical properties of a protein. Transformer based ESM1-b [28] model was pretrained on Uniref50 and then further finetuned on assay-labeled data.

**Which is better? Supervised learning or unsupervised learning?**

It is difficult to say which type of learning – supervised or unsupervised – is better for protein function prediction, as the best approach will depend on the specific details of the problem at hand.

However, given the nature of training data used for supervised learning, which is generated through DMS, supervised algorithms have several disadvantages such as: 1) high cost and time requirements – since DMS experiments involve synthesizing large number of mutant proteins and measuring their fitness, supervised algorithms require expensive dataset to be trained on. 2) limited scope – since supervised learning algorithms can only predict fitness values for protein family they are trained on, supervised learning algorithms are not generalizable. 3) incomplete coverage – DMS experiments typically introduce mutations at a limited number of positions in the protein sequence, and they may not cover all possible mutations at each position, hence, the data generated by DMS experiments may not be complete, and it may not be possible to predict the effects of all possible mutations using the data from these experiments. 4) limited accuracy: Deep mutational scanning experiments are typically performed *in vitro*, which means that they do not consider the effects of the mutations on the protein's function in a cellular environment which limits the accuracy of the predictions made using the data from these experiments, as the effects of mutations on protein function may be different *in vivo* compared to *in vitro*.

While unsupervised learning methods based on MSA input and embedding generation do not require expensive labelled data to be trained on, these methods have a few disadvantages such as: 1) high computational complexity for training – unsupervised learning algorithms can be computationally demanding, as they must discover the underlying structure of the data without the guidance of correct outputs. This can make it difficult to apply unsupervised learning to large datasets, and it can limit the scalability of unsupervised learning methods. 2) high computational complexity for input generation – MSA involves aligning many protein sequences, which can be computationally demanding. This can make it difficult to apply MSA to large datasets, and it can limit the scalability of MSA-based methods for protein function prediction. 3) limited accuracy - MSA assumes that proteins with similar sequences are likely to have similar functions. However, this assumption is not always true, and it may not be possible to accurately predict the function of a protein based solely on its sequence similarity to other proteins. 4) limited interpretability – unsupervised learning algorithms, particularly those using deep learning, often produce complex and hard-to-interpret models. This can make it challenging to understand and interpret the results of unsupervised learning algorithms, which in turn can limit their usefulness for some applications.

Both supervised and unsupervised learning methods have their own strengths and weaknesses, and the most appropriate method will depend on the availability and quality of the training data, the nature of the task, and other factors such as generalizability.

**Conclusion**

In this review paper, we covered multiple machine learning paradigms used to learn protein sequence to function mapping. We discussed why protein sequence to function mapping is an important problem for solving protein design and exploring evolutionary trajectories of proteins. The high throughput sequence-function data generation process was discussed. Then, we explored methodologies for supervised and unsupervised learning. It was also explained how machine learning methods can be used to fill gaps in sequence-function maps that are generated through DMS experiments. The importance of modeling epistatic interactions for accurate sequence-function maps was highlighted using examples of low order models. Additionally, it was noted that models developed for natural language processing are being translated for use in protein biology, indicating that proteins can be viewed as a pseudo-language where grammar is represented by epistatic interactions. Several disadvantages of both the paradigms of learning we discussed. In conclusion, both supervised and unsupervised learning methods have their own strengths and weaknesses when it comes to protein function prediction. Supervised learning algorithms have the advantage of using high-quality training data generated through DMS experiments, but they have several disadvantages, including high cost and time requirements, limited scope and incomplete coverage, which can limit their accuracy and usefulness. Unsupervised learning methods based on MSA input and embedding generation do not require expensive labeled data to be trained on, but they have disadvantages such as high computational complexity and limited accuracy, which can make it difficult to apply these methods to large datasets and limit their usefulness for protein function prediction. Overall, the best approach for protein function prediction will depend on the specific details of the problem at hand, and it may be necessary to use a combination of different methods in order to achieve the best results.

**References:**

1. Fowler, D., Fields, S. Deep mutational scanning: a new style of protein science. *Nature Methods* **11**, 801–807 (2014). <https://doi.org/10.1038/nmeth.3027>
2. Yang, K. K., Wu, Z., & Arnold, F. H. (2019). Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, *16*(8), 687–694. <https://doi.org/10.1038/s41592-019-0496-6>
3. Levy RM, Haldane A, Flynn WF. Potts Hamiltonian models of protein co-variation, free energy landscapes, and evolutionary fitness. *Curr Opin Struct Biol*. 2017 Apr;43:55-62. doi: 10.1016/j.sbi.2016.11.004. Epub 2016 Nov 18. PMID: 27870991; PMCID: PMC5869684.
4. Hsu, C., Nisonoff, H., Fannjiang, C., & Listgarten, J. (2022). Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, *40*(7), 1114–1122. <https://doi.org/10.1038/s41587-021-01146-5>
5. Hanning KR, Minot M, Warrender AK, Kelton W, Reddy ST. Deep mutational scanning for therapeutic antibody engineering. *Trends Pharmacol Sci*. 2022 Feb;43(2):123-135. doi: 10.1016/j.tips.2021.11.010. Epub 2021 Dec 9. PMID: 34895944.
6. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: “Efficient Estimation of Word Representations in Vector Space”, (2013\_; [http://arxiv.org/abs/1301.3781 arXiv:1301.3781].
7. Shuichi Kawashima, Minoru Kanehisa, AAindex: Amino Acid index database, *Nucleic Acids Research*, Volume 28, Issue 1, 1 January (2000), Page 374, <https://doi.org/10.1093/nar/28.1.374>

Ofer, D. & Linial, M. ProFET: feature engineering captures high-level protein functions. *Bioinformatics* 31, 3429–3436 (2015)

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: “Efficient Estimation of Word Representations in Vector Space”, (2013); [http://arxiv.org/abs/1301.3781 arXiv:1301.3781].

Breiman, L. Classification and Regression Trees (Routledge, 2017)

Breiman, L. Random forests. Mach. Learn. 45, 5–32 (2001)

Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn*. 20, 273–297 (1995)

Rasmussen, C. E. & Williams, C. K. I. Gaussian Processes for Machine Learning (MIT Press, 2006)

Romero, P. A., Krause, A. & Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl Acad. Sci.* USA 110, E193–E201 (2013)

Saito, Y. et al. Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS Synth. Biol.* 7, 2014–2022 (2018)

Bedbrook, C. N., Yang, K. K., Rice, A. J., Gradinaru, V. & Arnold, F. H. Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLoS Comput. Biol.* 13, e1005786 (2017)

Zeng, H., Edwards, M. D., Liu, G. & Gifford, D. K. Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics* 32, i121–i127 (2016)

Gomes, J., Ramsundar, B., Feinberg, E. N. & Pande, V. S. Atomic convolutional networks for predicting protein-ligand binding affinity. Preprint at https://arxiv.org/abs/1703.10603 (2017)

Dehouck, Y. et al. Fast and accurate predictions of protein stability changes upon mutations using statistical potentials and neural networks: PoPMuSiC-2.0. *Bioinformatics* 25, 2537–2543 (2009)

Mount, D. W. (2004). *Bioinformatics: Sequence and Genome Analysis* (2nd ed.). Cold Spring Harbor Laboratory Press

Milot Mirdita, Lars von den Driesch, Clovis Galiez, Maria J. Martin, Johannes Söding, Martin Steinegger, Uniclust databases of clustered and deeply annotated protein sequences and alignments, *Nucleic Acids Research*, Volume 45, Issue D1, January (2017), Pages D170–D176, <https://doi.org/10.1093/nar/gkw1081>

Baris E. Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, Cathy H. Wu, UniRef: comprehensive and non-redundant UniProt reference clusters, Bioinformatics, Volume 23, Issue 10, 15 May (2007), Pages 1282–1288, <https://doi.org/10.1093/bioinformatics/btm098>

Bairoch A, Apweiler R. The SWISS-PROT protein sequence database: its relevance to human molecular medical research. *J Mol Med* (Berl). (1997) May;75(5):312-6. PMID: 9181472.

Johnson, L.S., Eddy, S.R. & Portugaly, E. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics* **11**, 431 (2010). <https://doi.org/10.1186/1471-2105-11-431>

Sim NL, Kumar P, Hu J, Henikoff S, Schneider G, Ng PC. SIFT web server: predicting effects of amino acid substitutions on proteins. *Nucleic Acids Res.* 2012 Jul;40(Web Server issue):W452-7. doi: 10.1093/nar/gks539. Epub (2012) Jun 11. PMID: 22689647; PMCID: PMC3394338.

Hopf TA, Ingraham JB, Poelwijk FJ, Schärfe CP, Springer M, Sander C, Marks DS. Mutation effects predicted from sequence co-variation. *Nature Biotechnol*. (2017) Feb;35(2):128-135. doi: 10.1038/nbt.3769. Epub (2017) Jan 16. PMID: 28092658; PMCID: PMC5383098.

Riesselman, A.J., Ingraham, J.B. & Marks, D.S. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods* **15**, 816–822 (2018). <https://doi.org/10.1038/s41592-018-0138-4>

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. In Proceedings of the National Academy of Sciences (Vol. 118, Issue 15). Proceedings of the National Academy of Sciences. <https://doi.org/10.1073/pnas.2016239118>

1. Alley, E.C., Khimulya, G., Biswas, S. *et al.* Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods* **16**, 1315–1322 (2019). <https://doi.org/10.1038/s41592-019-0598-1>
2. Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., & Linial, M. (2021). ProteinBERT: A universal deep-learning model of protein sequence and function. Cold Spring Harbor Laboratory. <https://doi.org/10.1101/2021.05.24.445464>
3. Dept. Biomedical Informatics Columbia University. “Seminar: Multimodal deep learning for protein engineering (Kevin Yang).” *YouTube*, 11 Oct. (2022), <https://youtu.be/lICSBRNdNiU>
4. Jiang, D. et al. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J. Cheminform*. 13, 12 (2021).
5. Yang, K. et al. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model*. 59, 3370–3388 (2019).
6. Valsecchi, C. et al. Predicting molecular activity on nuclear receptors by multitask neural networks. *J. Chemom*. (2020).
7. Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2022). ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE transactions on pattern analysis and machine intelligence*, *44*(10), 7112–7127. https://doi.org/10.1109/TPAMI.2021.3095381