2023 IEEE 3rd International Conference on Computer Communication and Artificial Intelligence (CCAI) | 979-8-3503-3526-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCAI57533.2023.10201310

Deep Learning Models for Cancer Classification from Microarray Gene Expression Profiles

Aiguo Wang*
School of Electronic Information Engineering
Foshan University
Foshan, China
wangaiguo2546@163.com

Qinghao Hu
School of Economics and Management
Southeast University
Nanjing, China
hqhseu@163.com

***Abstract*—Gene expression profiles measured by microarray technology enables accurate identification of disease genes, prediction of cancers, and distinguishing tumor subtypes at the molecular level. However, these profiles are characterized by a small sample size and high dimensionality, which would inevitably degrade the performance of analysis models. In this study, we proposed a deep learning-based model to improve the prediction accuracy. Specifically, we first use the minimum redundancy maximum relevancy feature selector to discard irrelevant and noisy features. Then, we utilize a deep autoencoder to learn complex and nonlinear relationships among data. Finally, a predictor is trained on the latent representation to classify cancer. We conduct extensive experiments on four publicly available microarray datasets and compare the proposed model with six commonly used feature selectors using naïve bayes and decision tree in terms of accuracy and F1. Results demonstrate the superiority of the proposed model over its competitors.**

Keywords—microarray data, cancer prediction, deep learning, autoencoder, feature learning

I. INTRODUCTION

In the post-genome era, microarray technology has greatly facilitated us in conducting biological analysis tasks at the molecular level such as the prediction of cancers, identification of disease genes, and classification of tumor subtypes [1]. Gene expression profiles, however, are typically characterized by a small sample size (as few as tens of samples) and high dimensionality (as many as thousands of features), which inevitably results in degraded performance of machine learning models or statistical analysis models [2]. A classifier, for example, trained on the original feature space can easily suffer from overfitting [3]. One widely used solution is to reduce the dimensionality by filtering out irrelevant and noisy features [4].

The primary goal of feature selection, also known as variable selection or gene selection in the context of microarray data analysis, is to discard noisy and irrelevant features while keeping informative features from original feature space. Accordingly, researchers have developed numerous feature selection methods to pursue enhanced performance, and we can classify them from various perspectives according to the general feature selection framework [4]. Firstly, we can divide feature selection methods into filter, wrapper, embedded, and hybrid methods based on whether they utilize a classification model to evaluate the quality of candidate features [5]. Wrapper methods use performance metrics (such as accuracy, error rates, and area under the curve) of a classification model to measure the quality of candidate features during the feature selection process. A certain search strategy (such as forward search, backward search, floating search, and random search) is employed in wrapper methods to generate candidate feature subsets. Embedded methods are essentially wrapper methods that output the finally selected features after training the predictor. Lasso algorithm, decision tree, and random forest are three common representatives of embedded methods. In contrast, filter methods use metrics other than classification performance (such as distance, dependency, consistency, and information theory-based metrics) to measure the quality of candidate features. Compared to wrapper methods, filter methods have a lower computational cost. Hybrid methods combine the wrapper and filter methods in order to achieve high classification accuracy and fast computation. For example, one can use a filter method to eliminate noisy and irrelevant features and then use a wrapper method to further optimize the feature space. Alternatively, one can integrate a filter into a wrapper method to reduce the search space. Secondly, we can divide existing feature selection methods into feature ranking methods and feature subset methods based on the output of the feature selector, where the former generates a ranked list of features and requires a further step to determine the final selected features.

In addition to selecting informative features, mining their latent representations also plays a crucial role in determining the performance of a classifier. In recent years, deep learning models have achieved significant success and revolutionized many areas such as computer vision, text mining, natural language processing, and bioinformatics [6]. For instance, Fakoor et al. trained a model based on principal component analysis (PCA) and autoencoder for cancer type classification [7]. Basavegowda et al. proposed a model that uses PCA and deep feed-forward networks for cancer classification [8]. However, PCA is a feature extractor and reduces interpretability to a certain extent. Furthermore, some studies have conducted data preprocessing on the entire dataset (i.e., performing feature selection on the union of training and test sets), which would lead to biased results. To address these limitations, we propose a deep learning-based classification model to automatically learn high-level complex relationships among genes and to pursue better prediction accuracy. We first use the minimum redundancy maximum relevancy algorithm to discard noisy and irrelevant features and to mitigate the high-dimensional issue and then apply deep autoencoders to learn latent representations for better generalization. The main contributions of this study

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation (No. 2020A1515011499).

are as follows: (1) We present a deep learning-based model for cancer prediction from gene expression profiles and demonstrate the power of autoencoder in learning nonlinear relationships among data. (2) We explore two different ways of building the prediction model, one combined with a feature selector and the other directly working on the original feature space. (3) We conduct comparative experiments against six competitors in terms of accuracy and F1 on four publicly available microarray datasets, covering binary and multi-class cases, and show the effectiveness of our proposed model.

The rest of this paper is organized as follows. Section II details the proposed deep learning-based cancer classification model and its building blocks. Experimental setup and results are illustrated in section III, followed by the conclusion section.

II. Deep Learning Model for Cancer Classification

A. Autoencoder

An autoencoder, typically consisting of one input layer, one hidden layer, and one output layer, aims to reconstruct input in the output layer. That is, an autoencoder first transforms the n-dimensional input x into $h(x)$ in a k dimensional space using (1).

$$h(x) = f(W^{(1)}x + b^{(1)}) \tag{1}$$

where $W^{(1)} \in \mathbb{R}^{k*n}$ stores the weight matrix between the input layer and hidden layer, $b^{(1)} \in \mathbb{R}^{k*1}$ is the bias of hidden units, and $f(\cdot)$ is the activation function. Sigmoid function is among the commonly used one, as shown in (2):

$$f(Q) = 1/(1+\exp(-Q)) \tag{2}$$

Afterwards, we try to recover x from $h(x)$ by minimizing the difference $g(x, y)$ between x and y.

$$\begin{aligned} &\min g(x,y) \\ &\text{s.t. } y = f(W^{(2)}h(x) + b^{(2)}) \end{aligned} \tag{3}$$

where $W^{(2)} \in \mathbb{R}^{n*k}$ is the weight between the hidden layer and output layer, and $b^{(2)} \in \mathbb{R}^{n*1}$ is the bias of output units.

After the above process, we get a latent representation $h(x)$ of x. Furthermore, we can stack a collection of encoders to get a hierarchical architecture. In stacked autoencoder (SAE), the hidden layer of an autoencoder is the input to the adjacent layer, and the aim is to reconstruct the input with the last autoencoder. Given a SAE with P layers and the first layer is the input layer, for the p-th autoencoder, $W^{(p)}$ are the weight matrix and $b^{(p)}$ is the bias. The training procedure iterates with the greedy layer-wise scheme:

$$\begin{aligned} a^{(p)} &= f(Z^{(p)}) \\ Z^{(p+1)} &= W^{(p)}a^{(p)} + b^{(p)} \end{aligned} \tag{4}$$

where $Z^{(p)}$ is the input of the p-th layer. This helps us to learn nonlinear and complex relationships among data.

B. The Proposed Model

Considering that microarray data have noisy and irrelevant features, we could use a feature selector to pre-select a subset of informative features. We in this study use the minimum redundancy maximum relevance algorithm (MRMR) to choose the top ranked r features for the purpose of interpretability rather than using a feature extractor such as PCA [9]. Next, we utilize autoencoders on the reduced data to learn latent representations. Finally, we train a classifier on the representation of the last layer for cancer prediction. Fig. 1 presents the corresponding framework.

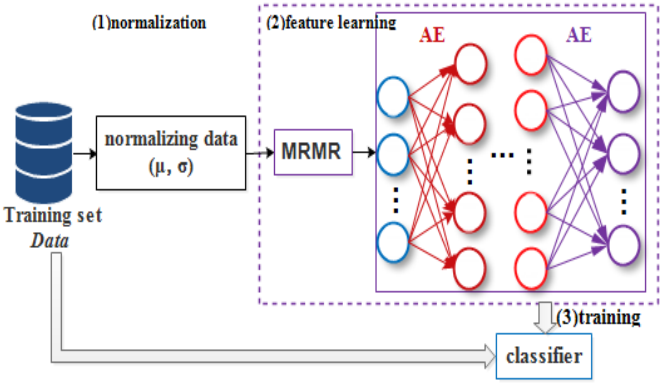

Fig. 1. The proposed classification model.

III. Experimental Setup and Results

A. Experimental Dataset

Comparative experiments are conducted on four microarray datasets that cover binary and multi-classes cases to evaluate the proposed method. Table I presents their descriptions, where we observe a high ratio (i.e., the last column) of the number of genes to sample size.

1) *BLADDER*: It has 5724 genes and 40 samples (with 11 in T1 stage, 10 in T2-T4 stage, and 19 in Ta stage). The goal is to distinguish the three subtypes.

2) *COLON*: There are 62 samples encoded by 2000 genes. It aims to build a classifier for colon cancer prediction.

3) *DLBCL*: Diffuse Large-B-cell Lymphoma (DLBCL) dataset contains 77 samples and there are 7219 genes.

4) *LEUKEMIA*: It consists of 25 AML samples, 9 T-cell ALL samples, and 38 B-cell ALL samples. There are collected from 5327 genes. The task is to distinguish the three subtypes.

TABLE I. Experimental Datasets

Dataset	#Classes	#Samples	#Genes	#SGR
BLADDER	3	40 (10/19/11)	5724	0.007
COLON	2	62 (40/22)	2000	0.031
DLBCL	2	77 (58/19)	7129	0.011
LEUKEMIA	3	72 (38/9/25)	5327	0.014

B. Experimental Setup

For the proposed model, we first empirically use MRMR to pre-select 25 features and then apply the autoencoder on the reduced data to learn latent representations. In this study, we only use a one-hidden-layer autoencoder (AE) and a two-hidden-layer stacked autoencoder (SAE) rather than fully explore a large number of architectures. We note corresponding methods as MRMR-AE and MRMR-SAE, respectively. Afterwards, we train classifiers on the learnt features. Besides, we can directly take as the input of an autoencoder the original features and we note them as all-AE and all-SAE for the purpose of comparison. Table II shows the hyperparameter setting for autoencoder training.

To demonstrate the efficacy of our proposed method, we include a comparison with six commonly used feature selection methods (i.e., reliefF, Mutual Information Maximization (MIM), Joint Mutual Information (JMI), Conditional Mutual Information Maximization (CMIM), Minimum Redundancy Maximum Relevance (MRMR), and Fast Correlation-Based Filter (FCBF)) [9]. Among these methods, FCBF returns a subset of features and is classified as a feature subset selection method, whereas the other five are feature ranking methods. For our experiments, we choose the top 25 ranked genes for the feature ranking methods. After selecting features, we train our cancer diagnosis classification model using two different classifiers with different metrics, namely Naive Bayes (NB) and Decision Tree (DT).

TABLE II. HYPER-PARAMETER SETTING

Model	Architecture	Parameter and Values
AE	*A*	weight regularization: 0.004, sparsity: 0.15, activation: sigm, optimizer: CG, hidden unit: 25
SAE	*A-A*	weight regularization: 0.004, sparsity: 0.15, activation: sigm, optimizer: CG, hidden unit: 25-25

To avoid the selection bias issue, a stratified ten-fold cross validation is used to generate independent training sets and test sets [10, 11], where one dataset is partitioned into ten equal-sized folds. Each fold is used as a test set to evaluate of power of a feature selection method and the trained classifier, and the remaining nine folds form a training set. We report the average of the ten results. Notably, feature selection and classifier training are only conducted on the training set. Besides, we transform the training set to zero mean and unit standard deviation and use its mean and standard deviation to normalize the test set. Fig. 2 is the overall framework, where the upper part is the training stage and the lower part is the test stage. Accuracy (*Acc*) and F1 are taken as the performance metrics.

C. Classification Performance

Tables III-IV show the classification accuracy and F1 of the proposed method and its competitors when NB and DT are used, respectively. The column "w/o" corresponds to the case of without using feature selection and the best F1 on each dataset is shown in bold. The row gives the average results. First, we observe that the use of feature selection method generally improves classification accuracy in the majority of cases. Second, we observe that MRMR obtains comparable accuracy to other feature selectors, which indicates its effectiveness. Third, we observe that the use of MRMR to first pre-select a subset of features enhances the performance of the autoencoder. For example, when using NB on *LEUKEMIA*, MRMR-AE obtains 95.83% accuracy compared to the 70.83% accuracy of all-AE and MRMR-SAE improves the accuracy from 68.06% of all-SAE to 93.06%. For decision tree, all-AE and all-SAE obtain 75.00% and 77.78% accuracy, respectively, compared to the 83.33% accuracy of MRMR-AE and 95.83% accuracy of MRMR-SAE. This is mainly because MRMR discards irrelevant and noisy features and helps an autoencoder to better learn inherent representations. Fourth, we see that the number of hidden layers has an impact on the performance of autoencoders. In the study, MRMR-AE performs better than MRMR-SAE with NB, while MRME-SAE performs better with DT. This indicates that the choice of the number of hidden layers should consider the used classification models.

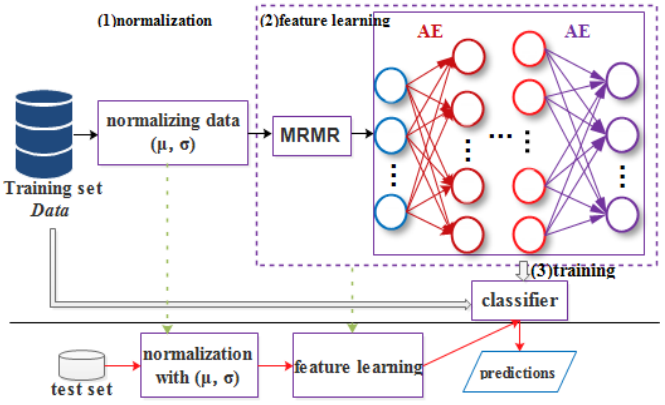

Fig. 2. Flowchart of the model training and prediction.

TABLE III. ACCURACY AND F1 COMPARISONS OF DIFFERENT METHODS USING NAÏVE BAYES

Dataset	w/o		ReliefF		MIM		CMIM		JMI		FCBF		MRMR		MRMR-AE		MRMR-SAE		all-AE		all-SAE	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
BLADDER	70.00	67.44	67.50	69.00	82.50	79.02	82.50	83.19	85.00	83.96	87.50	87.74	87.50	87.75	92.50	**90.61**	80.00	77.73	75.00	73.57	85.00	82.49
COLON	56.45	59.87	83.87	82.10	83.87	83.39	83.87	84.09	83.87	82.82	77.42	76.56	83.87	84.09	87.10	**86.28**	85.48	83.85	77.42	75.90	61.29	63.33
DLBCL	79.22	70.18	89.61	85.61	87.01	83.99	90.91	87.33	90.91	88.59	90.91	87.33	92.21	89.97	94.80	**93.35**	94.80	93.01	83.12	75.05	68.83	70.80
LEUKEMIA	97.22	**96.61**	94.44	91.25	93.06	89.27	95.83	92.68	95.83	94.34	95.83	92.90	95.83	92.90	95.83	93.91	93.06	92.10	70.83	69.13	68.06	59.94
average	75.72	73.53	83.86	81.99	86.61	83.92	88.28	86.82	88.90	87.43	87.92	86.13	89.85	88.68	92.56	91.04	88.34	86.67	76.59	73.41	70.80	69.14

TABLE IV. ACCURACY AND F1 COMPARISONS OF DIFFERENT METHODS USING DECISION TREE

Dataset	w/o		ReliefF		MIM		CMIM		JMI		FCBF		MRMR		MRMR-AE		MRMR-SAE		all-AE		all-SAE	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
BLADDER	65.00	58.59	70.00	66.31	62.50	57.27	57.50	50.09	62.50	56.91	57.50	50.47	62.50	58.17	72.50	72.50	87.50	**86.94**	80.00	77.55	82.50	79.23
COLON	75.81	73.32	72.58	70.37	77.42	75.90	80.64	78.50	77.42	76.56	75.81	73.86	79.03	77.35	80.64	78.86	85.48	**84.80**	70.97	68.98	54.84	53.61
DLBCL	81.82	73.98	85.71	80.46	88.31	84.02	89.61	85.39	85.71	79.91	81.82	74.72	88.31	84.57	89.61	86.59	90.91	**87.58**	88.31	83.62	74.03	62.57
LEUKEMIA	86.11	83.31	84.72	76.84	84.72	79.59	84.72	80.41	84.72	82.57	84.72	80.41	84.72	82.57	83.33	78.20	95.83	**95.50**	75.00	69.09	77.78	65.41
average	77.19	72.30	78.25	73.50	78.24	74.20	78.12	73.60	77.59	73.99	74.96	69.87	78.64	75.67	81.52	79.04	89.93	88.71	78.57	74.81	72.29	65.21

TABLE V. ACCURACY AND F1 COMPARISONS BETWEEN NAÏVE BAYES AND SOFTMAX

Dataset	MRMR-AE		MRMR-AE-S		MRMR-SAE		MRMR-SAE-S		all-AE		all-AE-S		all-SAE		all-SAE-S	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
BLADDER	92.50	90.61	92.50	91.13	80.00	77.73	87.50	84.62	75.00	73.57	85.00	83.60	85.00	82.49	85.00	82.15
COLON	87.10	86.28	79.03	77.93	85.48	83.85	83.87	82.39	77.42	75.90	80.64	78.86	61.29	63.33	70.97	68.98
DLBCL	94.80	93.35	94.80	93.35	94.80	93.01	93.51	91.92	83.12	75.05	81.82	75.54	68.83	70.80	76.62	65.33
LEUKEMIA	95.83	93.91	94.44	95.58	93.06	92.10	94.44	94.43	70.83	69.13	91.67	90.80	68.06	59.94	81.94	81.86
average	92.56	91.04	90.19	89.50	88.34	86.67	89.83	88.34	76.59	73.41	84.78	82.20	70.80	69.14	78.63	74.58

TABLE VI. ACCURACY AND F1 COMPARISONS BETWEEN DECISION TREE AND SOFTMAX

Dataset	MRMR-AE		MRMR-AE-S		MRMR-SAE		MRMR-SAE-S		all-AE		all-AE-S		all-SAE		all-SAE-S	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
BLADDER	72.50	72.50	92.50	91.13	87.50	86.94	87.50	84.62	80.00	77.55	85.00	83.60	82.50	79.23	85.00	82.15
COLON	80.64	78.86	79.03	77.93	85.48	84.80	83.87	82.39	70.97	68.98	80.64	78.86	54.84	53.61	70.97	68.98
DLBCL	89.61	86.59	94.80	93.35	90.91	87.58	93.51	91.92	88.31	83.62	81.82	75.54	74.03	62.57	76.62	65.33
LEUKEMIA	83.33	78.20	94.44	95.58	95.83	95.50	94.44	94.43	75.00	69.09	91.67	90.80	77.78	65.41	81.94	81.86
average	81.52	79.04	90.19	89.50	89.93	88.71	89.83	88.34	78.57	74.81	84.78	82.20	72.29	65.21	78.63	74.58

Besides the two above classification models, we investigate the performance of the Softmax classifier that is widely used in deep learning models. Tables V-VI show its comparison to NB and DT, respectively. The corresponding results are indicated by MRMR-AE-S, MRMR-SAE-S, all-AE-S, and all-SAE-S. From Tables V-VI, we observe the mixed results. For example, MRMR-AE using DT performs better than MRMR-AE-S on *COLON*, but achieves lower accuracy on *BLADDER*, *DLBCL*, and *LEUKEMIA*. Second, we can also observe that the use of MRMR to pre-select a subset of features tends to obtain better performance. For example, MRMR-AE-S gets 92.5% accuracy on *BLADDER* compared to the 85.00% accuracy of all-AE-S, and MRMR-SAE-S improves the 85.00% accuracy of all-SAE-S to 87.50%. This also motivates us to optimize a classification model on the reduced feature space, that is, it would be better to first optimize the feature space before conducting down-stream analysis tasks.

IV. CONCLUSION

Microarray gene expression profiles offer us an objective means of classifying cancers, identifying tumors, and locating disease genes at the molecular level. The small sample size and high dimension, however, poses a great challenge. To this end, we propose a deep learning-based model towards better cancer classification performance. Specifically, we first use MRMR to pre-select a small subset of features to discard irrelevant and noisy features, and then utilize autoencoders to learn complex and nonlinear relationships among data. Extensive comparative experiments are conducted on four publicly available microarray datasets again six commonly used feature selectors in terms of accuracy and F1, where two different classification models are used. Results indicate the effectiveness of the proposed method. Besides, we evaluate the impact of the number of hidden layers on prediction accuracy. For the further work, we plan to consider other application fields such as protein-protein interaction [12] and other deep learning models [13, 14].

REFERENCES

[1] A. Wang, H. Liu, J. Yang, and G. Chen, "Ensemble feature selection for stable biomarker identification and cancer classification from microarray expression data," Computers in Biology and Medicine, 2022, p. 105208.

[2] A. Negi, A. Shukla, A. Jaiswar, J. Shrinet, and R. S. Jasrotia, "Applications and challenges of microarray and RNA-sequencing," Bioinformatics, 2022, pp. 91-103.

[3] A. Wang, N. An, G. Chen, L. Li, and G. Alterovitz, "Improving PLS–RFE based gene selection for microarray data classification," Computers in Biology and Medicine, 2015, vol. 62, pp. 14-24.

[4] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," ACM Computing Surveys, 2017, vol. 50, no. 6, pp. 1-45.

[5] A. Wang, N. An, G. Chen, L. Liu, and G. Alterovitz, "Subtype dependent biomarker identification and tumor classification from gene expression profiles," Knowledge-Based Systems, 2018, vol. 146, pp. 104-117.

[6] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," Briefings in Bioinformatics, 2017, vol. 18, no. 5, pp. 851-869.

[7] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber, "Using deep learning to enhance cancer diagnosis and classification," Proceedings of the International Conference on Machine Learning, 2013, vol. 28, pp. 3937-3949.

[8] H. S. Basavegowda and G. Dagnew, "Deep learning approach for microarray cancer data classification," CAAI Transactions on Intelligence Technology, 2020, vol. 5, no. 1, pp. 22-33.

[9] G. Brown, A. Pocock, M. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," The Journal of Machine Learning Research, 2012, vol. 13, pp. 27-66.

[10] A. Wang, N. An, J. Yang, G. Chen, L. Li, and G. Alterovitz, "Wrapper-based gene selection with Markov blanket," Computers in Biology and Medicine, 2017, vol. 81, pp. 11-23.

[11] A. Wang, H. Liu, and G. Chen, "Chaotic harmony search based multi-objective feature selection for classification of gene expression profiles," 2021 IEEE 9th International Conference on Bioinformatics and Computational Biology, 2021, pp. 107-112.

[12] Q. Yuan, J. Chen, H. Zhao, Y. Zhou, and Y. Yang, "Structure-aware protein–protein interaction site prediction using deep graph convolutional network," Bioinformatics, 2022, vol. 38, no. 1, pp. 125-132.

[13] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," Advances in Neural Information Processing Systems, 2021, vol. 34, pp. 15908-15919.

[14] T. Sanderson, M. L. Bileschi, D. Belanger, and L. J. Colwell, "ProteInfer, Deep neural networks for protein functional inference," Elife, 2023, vol. 12, p. e80942.