Recurrent Neural Network Encoding Decoding Translator based Prediction Protein Function and Functional Annotation using Recurrent Neural Network

1Md.Nazmul Hossain, 2Md. Khaled Ben Islam, 3Md. Monirul Islam*
1Pabna University of Science and Technology, Rajapur, Dhaka - Pabna Highway, Pabna 6600, Bangladesh
2 Pabna University of Science and Technology, Rajapur, Dhaka - Pabna Highway, Pabna 6600, Bangladesh
3,4,5Dhaka University of Engineering and Technology, Gazipur, Gazipur-1700, Bangladesh
Email: 1 nazmul.cse48@gmail.com,2 mdkhaledben@gmail.com, 3monir.duet.cse@gmail.com*

Abstract— **Protein sequences are symbols generally different characters representing the 20 amino acids used in human proteins those sequences can range from the very sort to the very long. Around 375 amino acids make up the normal human protein. There are many proteins database for the sequences are known but the function and functional annotation is not. The knowledge gap between us and what we don't know is widening. A major field of bioinformatics is to predict the function of the protein from its sequence or structure at the same time how can be judged how well these function prediction algorithms are performed. This paper proposed a novel approach for converting the protein function problem into a language translation problem by encoding the protein function in a new protein sequence language decoded and build a recurrent neural machine encoding decoding translator (RNNEDT) based on the recurrent neural networks model. The excellent performance on training testing datasets demonstrates the proposed system is an improving direction for protein function prediction. In summary, the proposed method converts the protein function prediction problem to a language translation problem and applies a recurrent neural network machine translation model for protein function prediction, and visualizes the annotation of molecular function, biological process, and cellular component.**

Keywords— protein function; annotation; recurrent neural network; encoding decoding translator.

I. Introduction

Proteins are the major biological mechanisms that make life possible. There are around 54 million protein sequences and members chose are exemplars of around 1.4 million protein sequences [1]. Protein function prediction usually a multi-label classification problem. The proposed method is using the modern machine learning (ML) method to better understand and predict the function of biological proteins and functional annotation in the field of biological BP stands for biological mechanism, MF stands for molecular function, and cellular component stands for cellular component (CC). The design space of protein is much larger than what we observe in the real world. To address this challenge, we are interested in computational and experimental work to modify and optimize proteins for a variety of uses in the field of biological process, molecular function, and cellular component. Bioinformatics researchers use protein function prediction methods to assign biological or biochemical functions to proteins. The term “protein function” refers to the molecular functions of a protein, such as a gene regulation, transport of materials, and catalysis of biochemical reactions (enzymes), among others. These are normal proteins that haven't been thoroughly studied or predicted using genomic sequence data. Data-intensive statistical procedures are often used to make these predictions. Nucleic acid sequence homology, gene expression profiles, protein domain structures, text mining of papers, phylogenetic profiles, phenotypic profiles, and protein-protein interaction are all sources of information. Protein function is a broad concept that encompasses anything from biochemical reaction catalysis to transport and signal transduction, and a single protein may play multiple roles in multiple processes or cellular pathways. "Something that happens to or through a protein" is a general definition of the function.

We focus on this We focus here on to Recurrent Neural Network method. The genome of an organism may consist of hundreds to tens of thousands of genes, this encodes for hundreds of thousands of different protein sequences. Gene and protein sequences can be determined quickly and cheaply thanks to the relatively low cost of genome sequencing. While thousands of organisms have been sequenced to date, many of the proteins are still poorly understood. Experimenting to determine a protein's function in the cell is a costly and time-consuming operation. Furthermore, functional assays are unlikely to provide a complete picture of protein activity, even though they are performed. As a result, using analytical methods to functionally annotate proteins has become critical. There are a number of statistical methods for predicting protein function that can be used to infer protein function from a variety of biological and evolutionary data, but there is still a lot of space for improvement. Protein function prediction that is accurate may have long-term consequences in biomedical and pharmaceutical science. The CAFA experiment's goal is to provide an impartial evaluation of computational methods, to encourage computational function prediction research, and to provide insight into the current state of the art in function prediction.

We are the first to suggest a method for converting the protein function prediction problem into a language translation problem and using a neural machine translation model to predict protein function. Classification of protein’s amino acid sequence to one of the protein family accession based on “uniprot_sprot_20160120.fasta”. The task is to predict the protein domain's class based on its amino acid sequence. Predict the functional annotation of MF, BP, CC.

II. Literature Review

The annotation of functional protein is an essential matter for in vivo and in silico biology. Several computational methods have been proposed that make use of motifs, domains, homology, structure, and physicochemical properties, among other features. Since the knowledge obtained using any of these features depends on the role to be assigned to the protein, there is no single approach that works best in all functional classification problems. In this paper, we present a novel method for better-representing protein function by combining various approaches. First, we formulated the function prediction problem as a language translation problem then formulate a multiclass classification problem defined on different Gene Ontology terms from the fasta file format according to NCBI protein data 2016 “uniprot_sprot_20160120.fasta.

Researchers have tried different computational methods in the last few decades for predicting protein function problems. Usually, the following methods are used for protein function prediction.

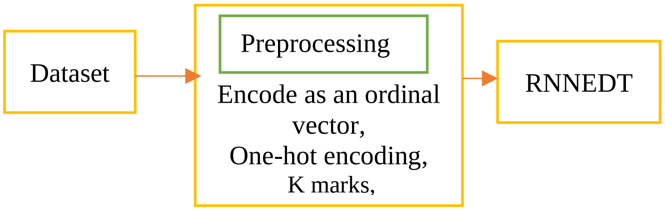
Firstly, and the most widely used method for the problem is the Basic Local Aligned Search Tool (BLAST) used to search query sequence against the existing protein databases, that contain experimentally determined protein function information and then use these homologous protein function information for the function prediction of the query sequence. For instance, Blastp, Gotcha, OntoBlast [2],[3]. Some methods the tool PSI-BLAST tool to find the remote homologous, such as the PEP method [4].

Secondly network-based methods. Various methods in this category use protein-protein interaction networks based on the assumption that interacted proteins share similar functions, gene-gene interaction networks, and domain co-occurrence networks[5].

Thirdly another information-based method. For example protein structure or microarray gene expression data PANNZER[11] and MS-KNN[12], SMISS[13],[14], SVMProt[6],[7].

The last state is machine learning method-dep learning neural network use multilayer data representation and abstraction. Which includes machine learning predicts new anti-crispr proteins. DeepPred[15][17], DeepGO[18] NMT Bidirectional LSTMProtCNN, ProtENN[19], Top pick HMM, ProLanGO[8]–[10]. RNNEDT method based on recurrent neural network encoder-decoder proposed method provides a new way predicting protein function and annotation of protein. Matric of the RNNEDT is multiclass log loss and accuracy on train, test, and validation data.

III. Proposed Methodology



Data collection

The utilized dataset is collected from National Center for Biotechnology Information [20]. It is a fasta type dataset.

Table 1 describes UniprotKB.

db	UniqueIdentifier	EntryName
OrganismName	OrganismIdentifier	ProteinExistence
ProteinSequences	ProteinName	SequenceVersion

The following are the five features that have been presented to us:

sequence: These are usually the model's input features. This domain's amino acid sequence. There are 20 very common amino acids (frequency > 1,000,000) and four very rare amino acids: X, U, B, O, and Z.

falily_accession: “uniprot_sprot_20160120.fasta” has contained 550302 protein sequences and 21601 Species Identification this is used here as family_accesion or class. These are usually the labels for the model.

sequence_name:The form "uniprot_accession_id/start_index-end_index". We extract EntryName from UniprotKB as sequence name.

aligned_sequence: Contains a single series from the multiple sequence alignment, with differences preserved, with the rest of the family members in the seed.

family_id: SequenceVersion is used as the family id.

A. Data preprocessing for ML with protein sequence data

We used three preprocessing techniques in this section, including (i) Encode the sequence information as an ordinal vector and work with that directly (ii)Use the resulting array to one-hot encode the sequence letters. (iii)Using different language processing methods to treat the protein sequence as a language (text).

Encode the sequence information as an ordinal vector and work with that directly: This method encode each amino acid characters as an ordinal value. “PESRIRLSTRRDA” becomes [13,4,16,15,8,15,10,16,17,15,15,3,1]. We have to transform this textual data into the numerical form that the machine can process. I have used one hot encoding method for the same with considering 20 common amino acids as other uncommon aids are less in quantity. The dictionary {'A': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8, 'K': 9, 'L': 10, 'M': 11, 'N': 12, 'P': 13, 'Q': 14, 'R': 15, 'S': 16, 'T': 17, 'V': 18, 'W': 19, 'Y': 20}.

Dictionary Length is 20. The dictionary consists of 20 amino acids in incremental order with integer values to be used for integer encoding.

one-hot encode the sequence letters and use the resulting array: We use one-hot encoding to represent the protein sequence the ‘ATGC’ would become [0,0,0,1] ,[0,0,1,0], [0,1,0,0], [1,0,0,0]. The one codded vector can either be concatenated or turned into dimensional arrays.

Treat the protein sequence as a language (text) and use various language processing methods:

The above step results in vectors of uniform length and that is a requirement for feeding data to classification or regression algorithm. To get vector or uniform length with the above approaches, we have to resort to stuff like truncating sequences or padding with ‘n' or ‘0'. Protein sequence can be viewed metaphorically as the language of life. The language contains both instructions and functions for the molecules present in all living things. The sequence language analogy continuous with the genome subsequences (genes and genes families) are sentences k-mers and peptides (motifs) are words and amino acids are alphabet. This is why the groundbreaking work in the field of natural language processing should be applied to the natural language of protein sequences. The used method is simple and easy. In this method, I first take the long biological sequence and break it down into k-mar length overlapping ‘words. If word of length 4, ‘MAFSAE’ becomes: ‘MAFS’, ‘AFSA’, ‘FSAE’. The number of words will be L-K+1. Where L= the length of the sentence. K= the length of the word. Here K=4 is an arbitrary value. We have a vocabulary of size KL possible words. The length of the words can be adjusted to fit the situation. For any given application, the word length and amount of overlap must be calculated empirically. In computational biology, we refer to this type of manipulations as “k-mer counting” or counting the occurrences of each possible k-mer sequence. There are specialized tools for this work. But recurrent neural network natural language processing tools make it super easy. It returns a k-mers ‘words’. We join the word into a sentence then apply the recurrent neural network encoder-decoder translation natural language processing method to the sentences.

Recurrent Neural Network Encoding Decoding Translator (RNNEDT)

This is made possible by the sequence to sequence network, a basic but effective concept in which two recurrent neural networks collaborate to turn one sequence into another. An encoder network compresses an input sequence into a vector, which is then unfolded into a new sequence by a decoder network. First we parse the data from fasta file and save the id and sequence in a text file. Load the data as language. We consider the sequence as Protein Language and the id is gene ontology language. We consider the sequence as proLang and id as GOALang. The data for this project is a set of many thousands of ProLang to GOALang translation pairs.

Before going any further. The file contains a list of translation pairs separated by tabs. We will represent each word in a language as a one-hot vector or giant vector of zeros except for a single one, similar to the character encoding used in the character-level RNN (at the index of the word). There are much many more words than there are hundreds of characters in a script, so the encoding vector is much larger. ProLang is now treated as a title, and GOALang is treated as an index. We'll need a specific index for each term to use as the networks' inputs and goals later. We'll use a helper class named Lan to keep track of anything. It has word→ index (word2index) and index→ word (index2word) dictionaries, as well as a list of each word (word2count) that we can use to replace uncommon words later. Since the files are all in Unicode, we'll convert Unicode characters to ASCII and reassemble them. To read the data file, we'll divide it into lines, then pair the lines together. The RNNEDT model is depicted in Figure 1.

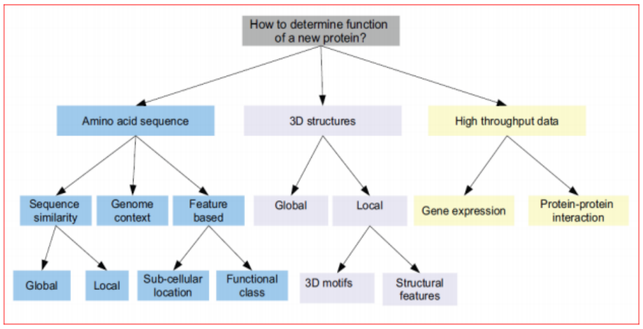


Figure 1. Overview of protein function prediction method

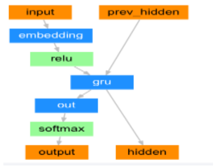


Figure 2. Decoder

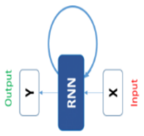


Figure 3. RNN

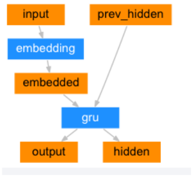


Figure 4. Encoder

Seq2Seq Mode: A Recurrent Neural Network (RNN) is a network that operates on a series and feeds its own output into subsequent steps. A seq2seq network, also known as an encoder-decoder network, is a model that consists of two RNNs called the encoder and decoder. The encoder reads an input sequence and produces a single vector, which is then read by the decoder to generate an output sequence. In the ideal case, a seq2seq model encodes the "value" of the input sequence into a single vector — a single point in some N-dimensional space of sentences.

Encoder: A seq2seq network's encoder is an RNN that outputs a value for each word in the input sentence. The encoder outputs a vector and a hidden state for each input word, and the hidden state is used for the next input word.

Decoder: The decoder is a second RNN that takes the encoder's output vectors and generates a string of words to construct the translation. Only the encoder's last output is used in the simplest seq2seq decoder. The decoder is a second RNN that uses the encoder's output vectors to produce a string of words from which the translation is constructed. In the simplest seq2seq decoder, only the encoder's last output is included.

IV. RESULT AND ANALYSIS

In the output step, we get the final result performance of all executed machine learning models. By coding our proposed model, we can exam the results. In this part, we can observe the detailed accuracy of running all stated models. We measured accuracy, precision, recall, and f1-score. Moreover, we also measured the weighted and macro average of the scores for each classifier. All of these terms are obtained from the confusion matrix is an N × N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the real goal values to the machine learning model's predictions. This gives us a view of how our classification model is performing and what kinds of errors it is making. A general sight of the confusion matrix is shown in table 1.

Table 2. Confusion Matrix

	Predicted Yes	**Predicted No**
Actual Yes	TP	FN
Actual No	FP	TN

- True Positive (TP): The number of positive events or cases of a data set that are correctly predicted is denoted as TP.
- True Negative (TN): The number of negative events or cases of a data set that are correctly predicted is denoted as TN.
- False Negative (FP): FP is the measure of positive cases that are predicted incorrectly as negative.
- False Positive (FN): FN is the measure of negative cases that are predicted incorrectly as positive.
- Accuracy: Accuracy is the measure of correctly classified cases of a data set. It is expressed mathematically in equation 1.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (1)$$

- Precision (P): P is the ratio of correctly predicted positive cases to the total positive cases. A low false-positive rate correlates with high precision. It is a measure of the exactness of a classifier. It is defined mathematically in equation 2.

$$P = \frac{TP}{TP + FP} \qquad (2)$$

- Recall (R): R is the ratio of correctly predicted positive cases to the all predicted positive cases of a classifier. It is a measure of completeness of a classifier. R is defined mathematically in equation 3.

$$R = \frac{TP}{TP + FN} \qquad (3)$$

- F1-Score: Precision and Recall are weighted averages. F1 is usually more useful than accuracy, when there is uneven class distribution in the data set. It is shown mathematically in equation 4.

$$F1 - score = \frac{2 \times (P \times R)}{P + R} \qquad (4)$$

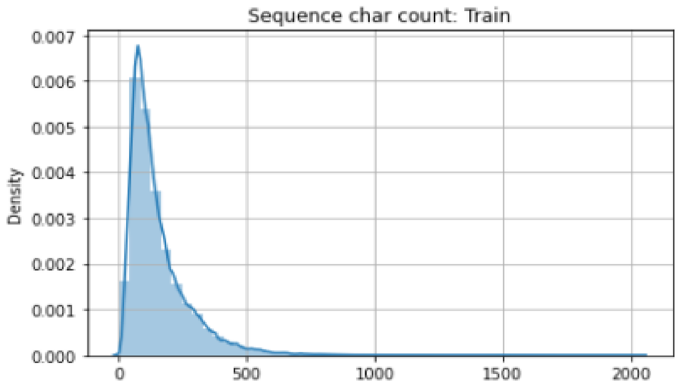


Figure 5: Sequence Character Count description

Figure 5. Shows the data and count the number of amino acid in each unaligned sequences The majority of unaligned amino acid sequences have between 50 and 300 characters.

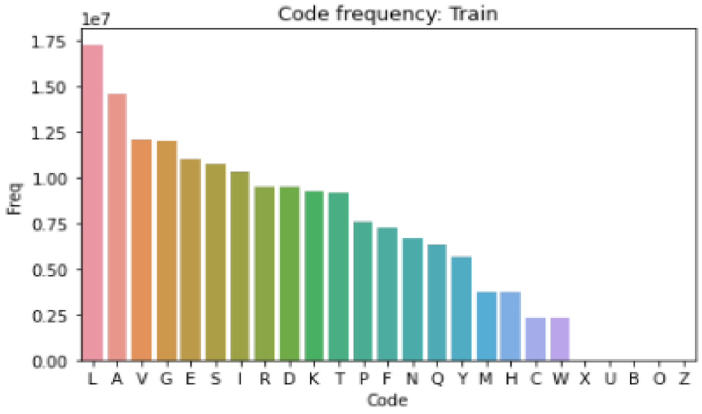


Figure 6: Amino acid frequency

Figure 6. According to the curve, the most frequent amino acid code is Leucine (L) followed by Alanine (A), Valine (V), and Glycine (G). As can be shown, the rare amino acids (X, U, B, O, and Z) are found in very small amounts. As a result, only 20 popular natural amino acids can be used for sequence encoding during preprocessing.

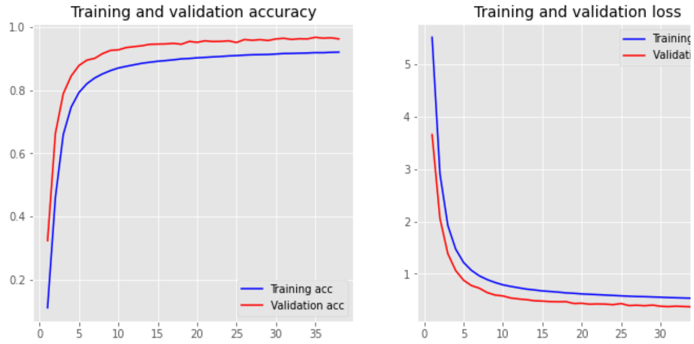


Figure 7: Training validation accuracy - loss

Figure 7. Shows trained with 10 epochs, batch_size of 256 and validated on the validation data.

Table 3. Confusion Matrix Score

	Precision	Recall	F1-Scrore
RNNEDT	94	86	90

Table 3. Show the average Precision, Recall and F1-Score all classes of protein sequences.

Table 4: Comparison training, validation, test accuracy among different model

Model	Train Accuracy	Validation Accuracy	Test Accuracy
RNNEDT	0.968	0.962	0.961
Bidirectional LSTM	0.94	0.86	0.90
ProtCNN	0.99	0.98	0.98

Table 3. Show the comparison among RNNEDT, Bidirectional LSTM and ProtCNN Train, Validation and Test Accuracy.

V. Conclusion

This paper proposed a novel method for the protein function prediction problem. We consider the gene ontology term as GOALang and protein sequence as ProLang. The method added regularization, Dropouts to prevent model over-fitting. We evaluate this method by comparing input-output and the target variables. In the result section, the paper calculates RNNEDT Accuracy, Precision, F1-score. Sometimes fails our proposed method for vanishing gradient. The method remembers things for just small durations of time, i.e. if we only use the information for a short period of time, it may be reproducible, but once a large number of words are entered, the information is lost.

References

[1] T. Bepler and B. Berger, 'Learning protein sequence embeddings using information from structure,' in 7th International Conference on Learning Representations, ICLR 2019, 2019.

[2] Z. Lv, C. Ao, and Q. Zou, 'Protein Function Prediction: From Traditional Classifier to Deep Learning,' Proteomics. 2019, doi: 10.1002/pmic.201900119.

[3] M. Kulmanov, M. A. Khan, and R. Hoehndorf, 'DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier,' Bioinformatics, 2018, doi: 10.1093/bioinformatics/btx624.

[4] W. Liu, B. Schmidt, and W. Müller-Wittig, 'CUDA-BLASTP: Accelerating BLASTP on CUDA-enabled graphics hardware,' IEEE/ACM Trans. Comput. Biol. Bioinforma., 2011, doi: 10.1109/TCBB.2011.33.

[5] S. Wan, Y. Duan, and Q. Zou, 'HPSLPred: An Ensemble Multi-Label Classifier for Human Protein Subcellular Location Prediction with Imbalanced Source,' Proteomics, 2017, doi: 10.1002/pmic.201700262.

[6] C. Z. Cai, L. Y. Han, Z. L. Ji, X. Chen, and Y. Z. Chen, 'SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence,' Nucleic Acids Res., 2003, doi: 10.1093/nar/gkg600.

[7] L. Y. Han, C. Z. Cai, S. L. Lo, M. C. M. Chung, and Y. Z. Chen, 'Prediction of RNA-binding proteins from primary sequence by a support vector machine approach,' RNA, 2004, doi: 10.1261/rna.5890304.

[8] Ö. S. Saraç, V. Atalay, and R. Cetin-Atalay, 'GOPred: GO molecular function prediction by combined classifiers,' PLoS One, 2010, doi: 10.1371/journal.pone.0012382.

[9] R. Cao, C. Freitas, L. Chan, M. Sun, H. Jiang, and Z. Chen, 'ProLanGO: Protein function prediction using neural machine translation based on a recurrent neural network,' Molecules, 2017, doi: 10.3390/molecules22101732.

[10] M. L. Bileschi et al., 'Using Deep Learning to Annotate the Protein Universe,' bioRxiv, 2019, doi: 10.1101/626507.

[11] P. Koskinen, P. Törönen, J. Nokso-Koivisto, and L. Holm, 'PANNZER: High-throughput functional annotation of uncharacterized proteins in an error-prone environment,' Bioinformatics, 2015, doi: 10.1093/bioinformatics/btu851.

[12] E. Lavezzo, M. Falda, P. Fontana, L. Bianco, and S. Toppo, 'Enhancing protein function prediction with taxonomic constraints - The Argot2.5 web server,' Methods, 2016, doi: 10.1016/j.ymeth.2015.08.021. [13]M. L. Bileschi et al., 'Using Deep Learning to Annotate the Protein Universe,' bioRxiv, 2019, doi: 10.1101/626507.

[13] J. S. Cottrell, 'Protein identification using MS/MS data,' Journal of Proteomics. 2011, doi: 10.1016/j.jprot.2011.05.014.

[14] A. Sureyya Rifaioglu, T. Doğan, M. Jesus Martin, R. Cetin-Atalay, and V. Atalay, 'DEEPred: Automated Protein Function Prediction with Multi-task Feed-forward Deep Neural Networks,' Sci. Rep., 2019, doi: 10.1038/s41598- 019-43708-3.

[15] Z. Cang and G. Wei, 'TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions,' PLoS Comput. Biol., 2017, doi: 10.1371/journal.pcbi.1005690.

[16] R. Fa, D. Cozzetto, C. Wan, and D. T. Jones, 'Predicting human protein function with multitask deep neural networks,' PLoS One, 2018, doi: 10.1371/journal.pone.0198216..

[17] Y. Liu, P. Palmedo, Q. Ye, B. Berger, and J. Peng, 'Enhancing Evolutionary Couplings with Deep Convolutional Neural Networks,' Cell Syst., 2018, doi: 10.1016/j.cels.2017.11.014.

[18] M. Kulmanov, M. A. Khan, and R. Hoehndorf, 'DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-awareclassifier,' Bioinformatics, 2018, doi: 10.1093/bioinformatics/btx624.

[19] V. Gligorijevic et al., 'Structure-Based Function Prediction using Graph Convolutional Networks,' bioRxiv, 2019, doi: 10.1101/786236.

[20] https://www.ncbi.nlm.nih.gov/