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 sequence 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. The average human protein comes in at around 375 amino acids. There are many proteins database for the sequences are known but the function and functional annotation is not. The gap between what we know and what we do not know is growing. A major field of bio informatics is to predict the function of protein from it sequence or structure at the same time how can be judged how well these function prediction algorithms are performed. This paper proposed a novel method to convert protein function problem into a language translation problem by a new proposed protein sequence language encoded to the protein function language decoded and build a recurrent neural machine encoding decoding translator (RNNEDT) based on 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 visualize 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 mechanics that make life possible. There are around 54 million protein sequences and members chosen are exemplars around 1.4 million protein sequences [1]. Protein function prediction usually a multi-label classification problem. The proposed method is using modern machine learning (ML) method to better understand and predict the function of biological proteins and functional annotation in the field of biological process (BP), molecular function (MF) and 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 interesting 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. Protein function prediction methods are techniques that bioinformatics researchers use to assign biological or biochemical roles to proteins. The term “protein function” refers to the molecular functions of a protein, such as: gene regulation, transport of materials, and catalysis of biochemical reactions (enzymes), among others. These proteins are usually ones that are poorly studied or predicted based on genomic sequence data. These predictions are often driven by data-intensive computational procedures. Information may come from nucleic acid sequence homology, gene expression profiles, protein domain structures, text mining of publications, phylogenetic profiles, phenotypic profiles, and protein-protein interaction. Protein function is a broad term: the roles of proteins range from catalysis of biochemical reactions to transport to signal transduction, and a single protein may play a role in multiple processes or cellular pathways. Generally, function can be thought of as, "anything that happens to or through a protein”.

We focus in this We focus here to Recurrent Neural Network method. The genome of an organism may consist of hundreds to tens of thousands of genes, this encode for hundreds of thousands of different protein sequences. Due to the relatively low cost of genome sequencing, determining gene and protein sequences is fast and inexpensive. Thousands of species have been sequenced so far, yet many of the proteins are not well characterized. The process of experimentally determining the role of a protein in the cell, is an expensive and time-consuming task. Further, even when functional assays are performed, they are unlikely to provide complete insight into protein function. Therefore, it has become important to use computational tools in order to functionally annotate proteins. There are several computational methods of protein function prediction that can infer protein function using a variety of biological and evolutionary data, but there is significant room for improvement. Accurate prediction of protein function can have longstanding implications on biomedical and pharmaceutical research. The CAFA experiment is designed to provide unbiased assessment of computational methods, to stimulate research in computational function prediction, and provide insights into the overall state-of-the-art in function prediction.

We first time propose a method converts the protein function prediction problem to a language translation problem and applies a neural machine translation model for protein function prediction. Classification of protein’s amino acid sequence to one of the protein family accession based on “uniprot_sprot_20160120.fasta”. The task is given the amino acid sequence of the protein domain, predict the class it belongs to. Predict the functional annotation of MF, BP, CC.

II. Literature review

The annotation of functional protein is an important matter for in vivo and in silico biology. Several computational methods have been proposed that make use of a wide range of features such as motifs, domains, homology, structure and physicochemical properties. There is no single method that performs best in all functional classification problems because information obtained using any of these features depends on the function to be assigned to the protein. In this study, we portray a novel approach that combines different methods to better represent protein function. First, we formulated the function prediction problem as a language translation problem then formulate 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 tired different computational methods in the last few decades for prediction protein function problem. Usually the following methods are used for protein function prediction.

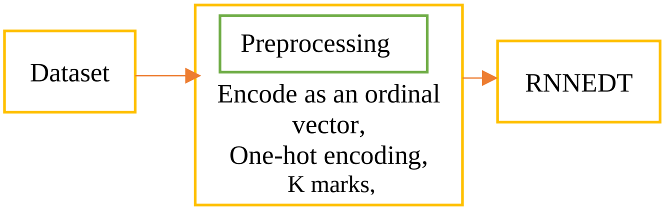
Firstly, and the most widely used method for the problem is Basic Local Aligned Search Tool (BLAST) used to search query sequence against the existing protein databases, that contain experimentally determined protein function information and then use these homologous protein function information for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

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

Thirdly other 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

We have been provided with 5 features; they are as follows:

sequence: These are usually the input features to the model. Amino acid sequence for this domain. There are 20 very common amino acids (frequency > 1,000,000), and 4 amino acids that are quite uncommon: X, U, B, O, Z.

falily_accession: “uniprot_sprot_20160120.fasta” has contained 550302 proteins sequences and 21601 Species Idification 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 sequence from the multiple sequence alignment with the rest of the members of the family in seed, with gaps retained.

family_id: We use SequenceVersion as family id

A. Data preprocessing for ML with protein sequence data

In this part, we used three preprocessing techniques including (i) Encode the sequence information as an ordinal vector and work with that directly, (ii) one-hot encode the sequence letters and use the resulting array, (iii) treat the protein sequence as a language (text) and use various language processing methods.

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: 20. The dictionary is considered 20 amino acids with integer values in incremental order to be farther 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:

Above step results in vectors of uniform length and that is a requirement for feeding data to classification or regression algorithm. So with the above methods we have to resort to things like truncating sequences or padding with ‘n’ or ‘0’ to get vector or uniform length. Protein sequence can be viewed metaphorically as the language of life. The language encodes instructions as well as functions for the molecules that are found in all life forms. 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. It is the reason that the amazing work done in the natural language processing field should also apply 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 sentence. K= the length of word. Here K=4 is an arbitrary value. We have a vocabulary of size KL possible words. Word length can be turned to suit the particular situation. The word length and amount of overlap need to be determined empirically for any given application. 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 recurrent neural network encoder decoder translation natural language processing method on the sentences.

Recurrent Neural Network Encoding Decoding Translator (RNNEDT)

This is made possible by the simple but powerful idea of the sequence to sequence network, in two recurrent neural networks work together to transform one sequence to another. An encoder network condenses an input sequence into a vector, and a decoder network unfolds that vector into a new sequence. 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 continuing. The file is a tab separated list of translation pairs. Similar to the character encoding used in the character-level RNN, we will be representing each word in a language as a one-hot vector, or giant vector of zeros except for a single one (at the index of the word). Compared to the dozens of characters that might exist in a language, there are many many more words, so the encoding vector is much larger.

We now consider ProLang as word and GOALang as index. We’ll need a unique index per word to use as the inputs and targets of the networks later. To keep track of all this we will use a helper class called Lan has word → index (word2index) and index → word (index2word) dictionaries, as well as a count of each word word2count to use to later replace rare words. The files are all in Unicode, to simplify we will turn Unicode characters to ASCII, make everything. To read the data file we will split the file into lines, and then split lines into pairs. Figure 1 shows the RNNEDT model.

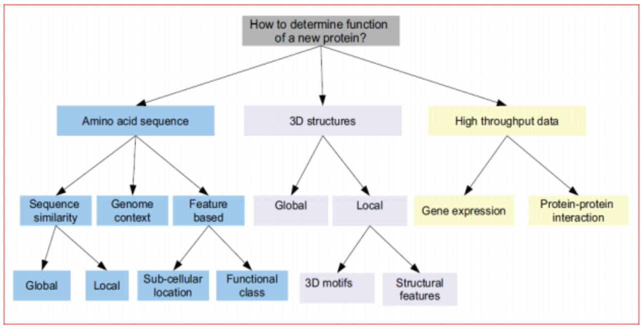
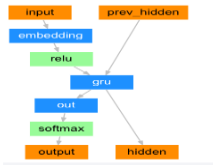
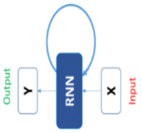


Figure 1. Overview of protein function prediction method





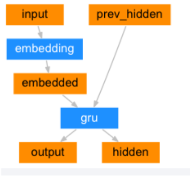


Figure 2. Decoder Figure 3. RNN Figure 4. Encoder

Seq2Seq Mode: A Recurrent Neural Network, or RNN, is a network that operates on a sequence and uses its own output as input for subsequent steps. A Sequence to Sequence network, or seq2seq network, or Encoder Decoder network, is a model consisting of two RNNs called the encoder and decoder. The encoder reads an input sequence and outputs a single vector, and the decoder reads that vector to produce an output sequence. With a seq2seq model the encoder creates a single vector, in the ideal case, encodes the “meaning” of the input sequence into a single vector — a single point in some N dimensional space of sentences.

Encoder: The encoder of a seq2seq network is a RNN that outputs some value for every word from the input sentence. For every input word the encoder outputs a vector and a hidden state, and uses the hidden state for the next input word.

Decoder: The decoder is another RNN that takes the encoder output vector(s) and outputs a sequence of words to create the translation. In the simplest seq2seq decoder we use only last output of the encoder. This last output is sometimes called the context vector as it encodes context from the entire sequence. This context vector is used as the initial hidden state of the decoder. At every step of decoding, the decoder is given an input token and hidden state.

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 actual target values with those predicted by the machine learning model. 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. High precision relates to the low false positive rate. It is a measure of 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: It is the weighted average of Precision and Recall. F1 is usually more useful than accuracy, when there is uneven class distribution in the data set. It is shown mathematically in equation 4.

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

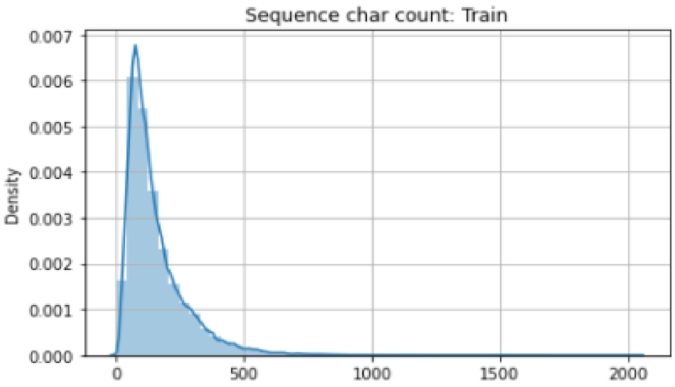


Figure 5: Sequence Character Count description

Figure 5. Shows the data and count the number of amino acid in each unaligned sequences. Most of the unaligned amino acid sequences have character counts in the range of 50-300.

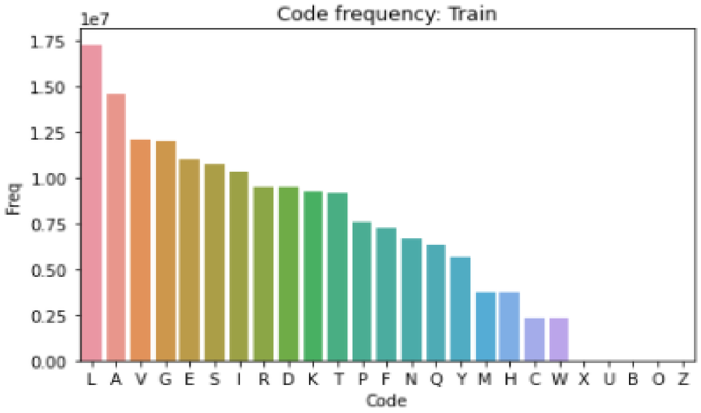


Figure 6: Amino acid frequency

Figure 6. According curve the most frequent amino acid code is Leucine (L) followed by Alanine (A), Valine (V) and Glycine (G). As we can see, that the uncommon amino acids (i.e., X, U, B, O, Z) are present in very less quantity. Therefore we can consider only 20 common natural amino acids for sequence encoding at the preprocessing.

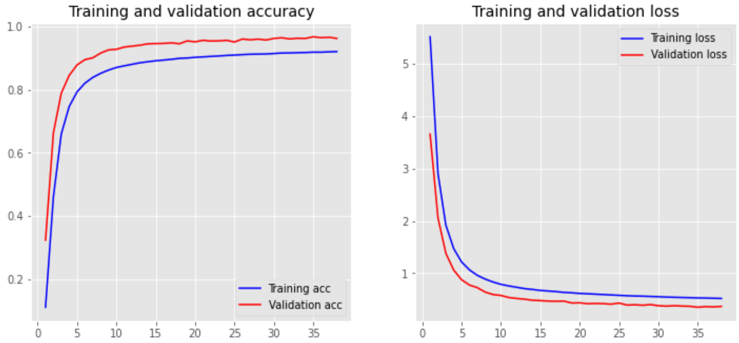


Figure 7: Training validation accuracy - loss

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 protein function prediction problem. We consider the gene ontology term as GOALang and protein sequence as ProLang.The method added regularization, Dropout to prevent model over-fitting. We evaluate this method comparing input output and target variable. 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 need the information after a small time it may be reproducible, but once a lot of words are fed in, this information gets lost somewhere.

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/