Automated Protein Function Description for Novel Class Discovery

Meet Barot  
Center for Data Science, New York University  
meetbarot@nyu.edu

Vladimir Gligorijevic  
Prescient Design, Genentech  
gigorijevic.vladimir@gene.com

Richard Bonneau  
Department of Biology, New York University  
Prescient Design, Genentech  
bonneau.richard@gene.com

Kyunghyun Cho  
Center for Data Science, New York University  
Prescient Design, Genentech  
kyunghyun.cho@nyu.edu

Abstract

Knowledge of protein function is necessary for understanding biological systems, but the discovery of new sequences from high-throughput sequencing technologies far outpaces their functional characterization. Beyond the problem of assigning newly sequenced proteins to known functions, a more challenging issue is discovering novel protein functions. The space of possible functions becomes unlimited when considering designed proteins. Protein function prediction, as it is framed in the case of Gene Ontology term prediction, is a multilabel classification problem with a hierarchical label space. However, this framing does not provide guiding principles for discovering completely novel functions. Here we propose a neural machine translation model in order to generate descriptions of protein functions in natural language. In this way, instead of making predictions in a limited label space, our model generates descriptions in the language space, and thus is capable of composing novel functions. Given the novelty of our approach, we design metrics to evaluate the performance of our model: correctness, specificity and robustness. We provide results of our model in the zero-shot classification setting, scoring functional descriptions that the model has not seen before for proteins that have limited homology to those in the training set. Finally, we show generated function descriptions compared to ground truth descriptions for qualitative evaluation.¹

1 Introduction

Determining the function of proteins is a fundamental problem in biology. Accurately identifying these functions through wetlab experimentation is costly, so computational approaches to predict protein function have been necessary to reduce the functional search space for experimentalists. However, many existing approaches to protein function prediction are only able to predict known functional categories, leaving out the possibility of classifying proteins into new categories.

In this work, we propose a framing of the protein function prediction problem that does not rely on discrete categories. Instead, we directly predict the common functional description of a group of proteins in natural language, modeling the problem as a neural machine translation task. We train our model on about 300k protein sequences from the Swiss-Prot database [1].

¹Code available at https://github.com/nowittynamesleft/protein_function_description
annotated with functional descriptions from the Gene Ontology (GO) [Ashburner et al., 2000]. We show that the model is capable of generating accurate function descriptions of proteins that are less than 30% identical to sequences in the training set and that have functions not present in the training set. We also propose three metrics to evaluate the correctness, specificity, and robustness of any model that can assign probabilities to a given sequence set and description.

2 Related Work

2.1 Protein Function Prediction

Many methods have been proposed for protein function prediction, though most do not consider the problem of discovering novel functions or generating their descriptions. As observed by Friedberg [2006], this has mainly been because of inherent difficulties of the flexibility of natural language, such as synonymous terms and ambiguity. These same difficulties were what led to the development of controlled and well-defined vocabularies of protein function, such as the Enzyme Commission Classification [Webb et al., 1992] and the Gene Ontology. As a result, the protein function prediction problem is generally framed as a supervised or semi-supervised multilabel classification problem with a structured output defined by these vocabularies, where the predicted labels are assumed to have some example in the training set [Bonetta and Valentino, 2020]. Much focus has been placed on this framing. The Critical Assessment of Functional Annotation (CAFA) [Zhou et al., 2019] serves as the main community benchmark for protein function prediction, and drives the field to improve upon previous methods. The CAFA evaluation considers proteins that can be described by existing categories. Yet many unlabeled proteins, especially in understudied organisms, are likely to perform functions that have not been seen before. The supervised approach does not address this possibility, and so new methods must be proposed for function discovery.

2.2 Clustering

Flat clustering-based approaches, by themselves, are not able to give much information about the new functional categories that they predict. They can only predict that a protein may belong to a category that has not been studied. One could compute average distances to clusters that contain known proteins, but beyond this, there is no testable hypothesis that the model can give about their function. NeXO [Dutkowski et al., 2013] and CliXO [Kramer et al., 2014] are both methods that generate an ontology of protein functions given relationships between proteins using hierarchical clustering. They aim at discovering novel functions. However, information about those new functions still rely on comparing the groupings to existing ontologies such as GO. Wang et al. [2018] describe a method that creates a concept hierarchy from phrases automatically extracted from scientific literature. This concept hierarchy is then aligned with the CliXO ontology in order to annotate proteins. However, this approach is still less flexible than generating free-form natural language.

2.3 Zero-shot learning approaches

Zero-shot learning approaches attempt to address the unseen class problem directly. DeepGOZero [Kulmanov and Hoehndorf, 2022] is a method that uses ontology axioms to predict for classes with no examples in the training set. However, the classes that are able to be predicted must be defined with ontological relations to seen classes. A similar limitation applies to clusDCA [Wang et al., 2015], which uses ontology relations to embed GO terms into a low dimensional space to perform zero-shot classification.

This constraint both restricts the possible novel functions that can be discovered and may not give sufficient information to design an experiment to test for the novel function.

2.4 Text generation and neural machine translation

Neural network-based text generation approaches have made significant progress in generating fluent and meaningful text [Fatima et al., 2022]. Further, deep learning-based techniques have shown promising results in image captioning methods [Hossain et al., 2019] and zero-shot classification of images [Radford et al., 2021]. Given enough data, deep learning methods have been shown to be capable of mapping between a range of input modalities and natural language. So far, there have been
Figure 1: High-level diagram of the proposed transformer encoder-decoder model. The model is trained to produce the most specific common function of the input protein sequences.

a few attempts to apply these methods to the protein function prediction domain. [Zhang et al. (2020)] use a graph-based generative model to generate Gene Ontology term names. However, the generation is limited to short phrases and relies on text descriptions from the GeneCards database [Safran et al., 2021] for the input.

Neural machine translation (NMT) is the automatic translation of written text from one natural language to another directly using neural networks [Cho et al., 2014]. NMT models have been widely deployed in production translation systems and show promise in domains other than natural language. Recently, a method called ProTranslator [Xu and Wang, 2022] has been proposed, which uses sequence, network and text description information concatenated into a 1-D feature vector in order to perform zero-shot classification on Gene Ontology terms. The authors also show that they are able to generate accurate and detailed descriptions for a set of proteins using a separate transformer model with this feature representation. Compared to ProTranslator, our method does not use any additional information to produce descriptions besides a set of protein sequences, and our model is trained directly to generate descriptions without pooling and losing positional information over the input sequences.

3 Methods

The following subsections give the motivation and formulations of the components of our method. Figure 1 contains a high-level overview.

3.1 Protein sets to describe

Biologists describe and categorize functions as abstractions of the common activity of a group of proteins, so we want our model to be able to perform this abstraction in a similar way. Formulating the problem as finding a single functional description for a single protein at a time is ill-defined, since a protein may have more than one distinct function [Jeffery (2018)]. Our task, then, is to find a description of the most specific function, $d_S$, for a set of sequences, $S = \{s_1, s_2, \ldots, s_n\}$, of usually different lengths, $|s_1| = l_1, \ldots, |s_n| = l_n$, that is common to all protein sequences $s_i \in S$. There is still a possibility that there is more than one specific common function among the set, but it is less likely with larger sets, e.g., $|S| = 32$.

3.2 Transformer encoder-decoder model with length transform

We use a transformer encoder-decoder model [Vaswani et al., 2017] with a length transform [Shu et al., 2020] to handle differing sequence lengths in order to average sequence features from the encoder. As a result of defining the learning task as a many-to-one problem, it was necessary to find a way to represent the common features of the set of sequences. The sequences’ representations should ideally be combined in some way that preserves amino acid ordering information, so we use the length transform in order to stretch the representations to the same shape in order to be averaged. This kind of length transform has been used previously in non-autoregressive neural machine translation.
problems Shu et al. [2020] and in protein design for changing protein sequence representations and
generating sequences of variable lengths Gligorijevic et al. [2021]. For each sequence \( s \in S \), we use
a transformer model with positional encoding and self-attention to obtain a representation \( h_s \) which
consists of \( |s| \) continuous-valued vectors. As described in Shu et al. [2020], the length transform takes
the input \( h_s \) of length \( |s| \) and transforms the sequence with a monotonic location-based attention into
\( h^l_s \) where \( l \) is the chosen output length so that \( |h^l_s| = l \). We choose \( l = \max_{s \in S} |s| \).

### 3.3 Autoregressive generation of descriptions

It is desirable to represent protein function in a compositional way, so that the model has the ability
to describe any given set of proteins without having to rely on examples of proteins with that specific
function. To do this, we generate protein function descriptions in natural language, which gives
the model the capability to compose a new function. We predict the tokens autoregressively, which
is a standard practice in the NMT literature of top performing methods. With the \( |S| \) sequence
representations \( h^l_s \) having all the same length after the length transform, we are able to take the
average of these abstract representations, giving us \( h_S \), the representation of the whole sequence
set. We use this representation in the transformer decoder in order to predict the next token of the
description \( d \) given all the previous tokens.

### 3.4 Zero-shot Classification setting

Fundamentally, our model assigns probabilities to pairs of protein sets and descriptions. In order to
evaluate the method, we use the zero-shot classification setting, where we wish to classify proteins
into unseen categories. We develop three metrics in the Evaluation section to evaluate the conditional
probability distribution \( P(d_S|S) \) learned by the model in this classification setting.

### 3.5 Generation (beam search)

Generation of descriptions is a search problem through the set of all possible output token sequences,
where the goal is to find the sequence with the largest probability. Generation given an autoregressive
model is a highly studied problem in the natural language processing literature. We use beam search
Graves [2012] in the current implementation in order to find reasonable generated descriptions. We
use a beam width of 10 with a length penalty of 2.0. Direct evaluation of these descriptions is an
unsolved problem: currently, manual inspection by expert human evaluators is the best method we
have.

### 3.6 Architecture and hyperparameters

The model used in our experiments has 2 layers in the encoder and 2 layers in the decoder, with 4
attention heads and 512 dimensions per layer. The sequence set size to train with was chosen to be
32, and the maximum sequence length was 1000 amino acids. We apply a dropout of 0.25 to the
encoder and decoder layers during training, use a length transform parameter fixed to be \( \sigma = 0.5 \),
and use a learning rate of 0.0001 with the Adam optimizer. We use a byte-pair encoding scheme for
our vocabulary with 1000 operations, resulting in a vocabulary size of 1084 subwords.

### 4 Evaluation

In this section, we define three metrics that can be computed using known functional descriptions in
order to evaluate our models’ learned probability distributions.

Generated descriptions are shown in the Results section for qualitative analysis. Quantitative analysis
of the generated descriptions requires data from human evaluators with expertise in protein function
in order to determine the accuracy of generated descriptions. A framework for performing that
analysis with expert curators is explored in the Discussion section.
4.1 Attribute 1: Annotation correctness.

Given a sequence set for which the model is assigning scores to function descriptions, descriptions of GO terms that annotate the entire sequence set should be scored higher than terms that do not annotate the entire sequence set.

Let \( D_S \) be the GO term descriptions associated with sequence set \( S \).

\[
P(d \in D_S|S) > P(d \notin D_S|S)
\]

A way to measure this attribute would be to calculate:

\[
\frac{1}{|D_S| \times |D_S^c|} \sum_{d_i \in D_S, d_j \notin D_S} 1(P(d_i|S) > P(d_j|S))
\]

where \( D_S^c \) is the complement of \( D_S \) and \( 1 \) is the indicator function.

4.2 Attribute 2: Specificity preference.

Among terms that do annotate the whole set, the model should score child terms higher than their ancestor terms. Let \( A(d) \) denote the description of a direct parent of the GO term described by \( d \).

\[
P(d \in D_S|S) > P(A(d) \in D_S|S)
\]

Note: any protein set that is annotated with \( d \) would always be annotated with \( A(d) \), \( A(A((d))) \) and so on.

A way to measure this attribute would be to calculate:

\[
\frac{1}{|D_S|} \sum_{d_i \in D_S} 1(P(d_i|S) > P(A(d_i)|S))
\]

4.3 Attribute 3: Annotation robustness.

Any set of sequences that have the same exact set of GO descriptions in common should be scored with the same rankings for those GO descriptions.

Let \( S_i \) and \( S_j \) be different sequence sets such that \( D_{S_i} = D_{S_j} \) and \( S_i \neq S_j \), and let \( R(X) \) be a ranking function that gives the ranks of entries in \( X \), in descending order.

\[
R_d(P(d \in D_{S_i}|S_i)) = R_d(P(d \in D_{S_j}|S_j))
\]

A way to measure this attribute would be to calculate the average Spearman’s rank correlation of the rankings for all sequence sets’ correct descriptions. Let \( R_{S_i} = R(P(D_{S_i}|S_i)) \):

\[
\frac{1}{N(N-1)} \sum_{S_i, S_j} \frac{\text{cov}(R_{S_i}, R_{S_j})}{\sigma_{R_{S_i}} \sigma_{R_{S_j}}}
\]

where \( N \) is the total number of sequence sets that have the exact set of GO descriptions \( D_{S_i} \). In reality, this number may be too large to actually sum (especially if \( |D_{S_i}| \) is small), so we approximate this measure by subsampling \( n < N \) sequence sets to average over instead. The sum is only calculated over non-identical pairs of sequence sets.

5 Data

We take sequences and annotations from the Uniprot-KB Swiss-Prot database, which is manually annotated and reviewed, in order to create our training and evaluation sets of proteins and function descriptions. This database had 566,996 proteins total. To show that our model can generalize to non-homologous proteins, we clustered the proteins into groupings with less than 30% sequence
Table 1: Number of proteins and GO terms in training and test sets.

<table>
<thead>
<tr>
<th></th>
<th>Train P&amp;F</th>
<th>Train P, Test F</th>
<th>Test P, Train F</th>
<th>Test P&amp;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proteins</td>
<td>316k</td>
<td>181k</td>
<td>20k</td>
<td>20k</td>
</tr>
<tr>
<td>GO terms</td>
<td>9k</td>
<td>2k</td>
<td>879</td>
<td>1.5k</td>
</tr>
</tbody>
</table>

Table 2: Model Performances

<table>
<thead>
<tr>
<th>Metric</th>
<th>Train P, Test F</th>
<th>Test P, Train F</th>
<th>Test P&amp;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation Correctness</td>
<td>0.8844</td>
<td>0.8014</td>
<td>0.7157</td>
</tr>
<tr>
<td>Specificity Preference</td>
<td>0.5765</td>
<td>0.5526</td>
<td>0.5701</td>
</tr>
<tr>
<td>Annotation Robustness</td>
<td>0.4020</td>
<td>0.1977</td>
<td>0.2362</td>
</tr>
</tbody>
</table>

identity using cd-hit [Li and Godzik, 2006], and separated these into training and test sets. To focus on the functions that were both specific enough and had a sufficient number of examples in our evaluation sets, we restricted the maximum number of proteins per GO term to 1280, and minimum number of proteins to 32. Hyperparameters chosen were tuned on the training set proteins with training function descriptions. The number of proteins and GO terms that were used after these restrictions in our training set and evaluation sets are listed in Table 1.

6 Results

We show model performances in Table 2. The table suggests that the model is able to rank unseen functions for protein sets that it has been exposed to in training, with the model’s rankings of identically annotated sets being in moderate agreement. For test proteins that have less than 30% sequence identity to the training set, the model is still able to assign rankings of 1000 randomly selected functions from the training set with a correctness 30% above random assignment (0.5). For the low-similarity test proteins that have functions that are not seen in the training set, the model is still able to rank 21% better than random rankings.

We are mainly focused on using the model for generation, and these metrics are meant mostly as guides for model design. The loss function used is not optimizing for classification accuracy; it is optimizing the model’s probability distribution to assign high probability to descriptions assigned to a sequence set.

We show sample test set descriptions in Table 3. The first row shows that the model describes verbatim a related term (GO:0001654, eye development) for the proteins selected, whereas the true term is appendage development (GO:0048736). Their common ancestor term is anatomical structure development (GO:0048856). This description is more specific than the actual term from which the proteins are sampled, but it is not accurate. The next generated description is more general than the actual description of the sampled set (modulates vs. activates), but is correct; it is the direct parent of the true term. The third generated description is related but ultimately different than the actual description of the protein set. The fourth generated description is more specific than both the true common GO description of the set (protein import, GO:0017038) and the generated description’s closest known GO term, protein exit from endoplasmic reticulum (GO:0032527). It is describing protein import into the nucleus from the endoplasmic reticulum, which is not currently a GO term, but if it was, it would be a descendant of both of these terms.

7 Discussion

In this work, we have proposed a novel method to generate protein function descriptions in order to discover new protein functions. We have demonstrated that our model can accurately rank unseen function descriptions for proteins not seen in the training set, and show promising results in generated function descriptions. Given that this model is trained using raw text descriptions of protein function, it is possible to extend this work to use descriptions from other databases besides the Gene Ontology, such as Pfam [Bateman et al., 2004], KEGG [Kanehisa et al., 2002], or Enzyme Commission classes. This increase in data could allow for higher quality descriptions, or the ability to query the model.
Table 3: Sample Test Set Description Generations

<table>
<thead>
<tr>
<th>True Common GO Term</th>
<th>True Common GO Description of Sequence Set</th>
<th>Model Generated Description of Sequence Set</th>
<th>Closest Known GO Term to Generated Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0048736, appendage development</td>
<td>&lt;SOS&gt; the process in which the anatomical structures of appendages are generated and organized. An appendage is an organ or part that is attached to the trunk of an organism. &lt;EOS&gt;</td>
<td>&lt;SOS&gt; the process whose specific outcome is the progression of the eye over time, from its formation to the mature structure. &lt;EOS&gt;</td>
<td>GO:0001654, eye development</td>
</tr>
<tr>
<td>GO:0045597, positive regulation of cell differentiation</td>
<td>&lt;SOS&gt; any process that activates or increases the frequency, rate or extent of cell differentiation. &lt;EOS&gt;</td>
<td>&lt;SOS&gt; any process that modulates the frequency, rate or extent of cell differentiation. &lt;EOS&gt;</td>
<td>GO:0045595, regulation of cell differentiation</td>
</tr>
<tr>
<td>GO:0071162, CMG complex</td>
<td>&lt;SOS&gt; a protein complex that contains the gins complex, cdc45p, and the heterohexameric mcm complex, and that is involved in unwinding dna during replication. &lt;EOS&gt;</td>
<td>&lt;SOS&gt; any process involved in forming the mature 3' end of a dna (mRNA) molecule. &lt;EOS&gt;</td>
<td>GO:0031124, mRNA 3'-end processing</td>
</tr>
<tr>
<td>GO:0017038, protein import</td>
<td>&lt;SOS&gt; the targeting and directed movement of proteins into a cell or organelle. Not all import involves an initial targeting event. &lt;EOS&gt;</td>
<td>&lt;SOS&gt; the directed movement of proteins from endoplasmic reticulum to the nucleus. &lt;EOS&gt;</td>
<td>GO:0032527, protein exit from endoplasmic reticulum</td>
</tr>
</tbody>
</table>

7.1 Future human-assisted evaluation of function discovery

As our scoring metrics for evaluation are automated, they can be used for optimizing the architecture and other hyperparameters of the model (either manually or with some search method). However, in the case of actual use on proteins that are not very well studied, it can be difficult to know whether a given description is accurate. Human-assisted evaluation will be needed for the descriptions generated for a given set of novel proteins. This feedback could be used to fine-tune the model to produce more accurate, fluid or generally desirable descriptions of proteins, as has been done for document summarization models [Ziegler et al., 2019, Stiennon et al., 2020].

One possible way of obtaining human feedback would be to ask an expert with knowledge of the Gene Ontology and familiarity with some families of proteins to choose between two descriptions for a given sequence set that is generated from a trained model. Doing this over a large enough dataset would allow us to train a reward estimation model that can then be used to fine-tune the original trained model using reinforcement learning. However, this would be expensive, as the task needs to be done by an expert. Richer information, such as ranking the similarities to an existing GO term, or suggesting changes to particular portions of the description, could be used to increase performance even with a small number of examples with human feedback.

References


