Check for updates

RESEARCH ARTICLE

Translate gene sequence into gene ontology terms based on statistical machine translation [version 1; peer review: 1 approved with reservations, 2 not approved]

Wang Liang¹, Zhao Kai Yong²¹Tencent Tech, Beijing, 100080, China²Department of Computer Science, Hong Kong Baptist University, Hong Kong, 999077, China

V1 First published: 01 Nov 2013, 2:231
<https://doi.org/10.12688/f1000research.2-231.v1>
 Latest published: 01 Nov 2013, 2:231
<https://doi.org/10.12688/f1000research.2-231.v1>

Abstract

This paper presents a novel method to predict the functions of amino acid sequences, based on statistical machine translation programs. To build the translation model, we use the “parallel corpus” concept. For instance, an English sentence “I love apples” and its corresponding French sentence “j’adore les pommes” are examples of a parallel corpus. Here we regard an amino acid sequence like “MTMDKSELVQKA” as one language, and treat its functional description as “0005737 0006605 0019904 (Gene Ontology terms)” as a sentence of another language. We select amino acid sequences and their corresponding functional descriptions in Gene Ontology terms to build the parallel corpus. Then we use a phrase-based translation model to build the “amino acid sequence” to “protein function” translation model. The Bilingual Evaluation Understudy (BLEU) score, an algorithm for measuring the quality of machine-translated text, of the proposed method reaches about 0.6 when neglecting the order of Gene Ontology words. Although its functional prediction performance is still not as accurate as search-based methods, it was able to give the function of amino acid sequences directly and was more efficient.

Keywords

GFP, gene function prediction, machine translation

Open Peer Review

Approval Status ? X X

	1	2	3
version 1	?	X	X
01 Nov 2013	view	view	view

1. **Paul Pavlidis**, University of British Columbia, Vancouver, BC, Canada
2. **Karin M. Verspoor** , University of Melbourne, Melbourne, Australia
3. **Juliana Bernardes**, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Wang Liang (wangliang.f@gmail.com)

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2013 Liang W and Kai Yong Z. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

How to cite this article: Liang W and Kai Yong Z. **Translate gene sequence into gene ontology terms based on statistical machine translation [version 1; peer review: 1 approved with reservations, 2 not approved]** F1000Research 2013, 2:231 <https://doi.org/10.12688/f1000research.2-231.v1>

First published: 01 Nov 2013, 2:231 <https://doi.org/10.12688/f1000research.2-231.v1>

Introduction

Determining the functions of proteins is a central problem in biology. There are many databases like RefSeq (<http://www.ncbi.nlm.nih.gov/refseq/>) that store amino acid sequences and their corresponding functions. However, almost all protein functional prediction methods rely on the identification, characterization, or quantification of sequence similarities between the available proteins of interest¹. Even sequences that are similar do not necessarily have identical function. A sequence may be similar to many other sequences, so it can be difficult to choose the most appropriate one. In addition, there is no way to deduce function if there are no similar sequences in any available database.

At present, many machine learning research methods are being applied to deal with the above mentioned problems. Two examples are, Support Vector Machines (SVM); a supervised learning model which is mainly used for classification and regression analysis, and network-based methods such as protein interactome networks^{2,3}. In addition, the sequence features, structures, evolution of amino acid sequences, and other characteristics are also utilized for functional prediction^{4,5}.

The same sequence may have different meanings in different contexts. Different sequences may also represent the same meaning. These problems are normally called “lexically/structurally ambiguous” in the language translation field. Current machine translation research could effectively avoid these problems by building a translation model between two ‘languages’.

Since many amino acid sequences and their corresponding functional description are already publically available, it is possible to build their translation relationships by mature machine translation technology. We address this task in this paper. When presented with a new amino acid sequence we aimed to provide a function-based translation.

To build the model, we need to construct the “parallel corpus” of amino acid sequences and their functional descriptions. The term “parallel corpus” is typically used in linguistic areas to refer to texts that are translations of each other. A parallel corpus contains the data for two languages.

For example, a parallel corpus from English to French translation:

English sentence: *I love apples*

French sentence: *J'adore les pommes*

Here, we selected amino acid sequences and their functional descriptions in Gene Ontology form to build the corpus⁶. The reason for using the Gene Ontology is that it gives the unified descriptions for protein functions of all species. The Gene Ontology data can be found on the website www.geneontology.org (<http://www.geneontology.org/GO.downloads.annotations.shtml>). The corresponding amino acid sequence data can be downloaded from the website www.uniprot.org (<http://www.uniprot.org/downloads>), which is a central repository of amino acid sequence. We use the database ID to identify a sequence and find its correspondences in above two data sources.

An example of a parallel corpus from “amino acid sequence” to “function” translation:

Amino acid sequence (amino acid):

MTMDKSELVQKAKLAEQAERYDDMAAAMKAVTE-QGHELSNEERNLLSVAYKNVV

Function description (Gene ontology term ID):

0005737 0006605 0019904 0035308 0042470 0042826 0043085

The basic idea of machine translation is very simple. It tries to infer “rules” from the parallel corpus and build a translation model. The “rules” are mainly the words or phrase correspondences between two languages. Similar sequences will get near translation results. Although the application of translation technology in protein functional prediction seems reasonably simple, as far as we know, there is no existing program that applies machine translation to protein sequence function analysis.

Parallel corpus

Here we build three groups of gene parallel corpora. Group 1 only contains human amino acid data. It has 45,538 amino acid sequences and related Gene Ontology descriptions. Group 2 is a mixed dataset of several model species. It contains about 100,000 amino acid sequences. Group 3 is the largest dataset. It contains more than 2,000,000 amino acid sequences. All of the corpora can be downloaded from www.geneontology.org (Gene Ontology data) and www.uniprot.org (amino acid sequence data). We also provide these Gene ontology and amino acid sequence data in ‘data’ directory of [supplementary material](#) with this paper.

First, we need to “clean” the corpus. Data cleaning is the process of detecting and correcting (or removing) corrupted or inaccurate records from a dataset. For machine translation, sequences with significant differences in length between the sources and targets should be removed. The empty and overlong sequences should also be deleted. For amino acid data, we deleted long sequences (amino acid sequence length>1000 (amino acids), Gene ontology terms number >100) and empty sequences.

After data cleaning, we had about 14,000 pieces of data for 45,538 human amino acids. Data of groups 2 and 3 are also reduced to about 1/4 of the original data size.

Amino acid sequences are not naturally segmented. So we still need to “segment” them.

Text segmentation means dividing a string of written language into its component words. In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word delimiter. But for some East-Asia languages like Chinese, there are no natural delimiters. To process these languages, we need to segment sentences into “words” sequences first. For example:

A Chinese sentence:

我爱苹果(*Iloveapple*)

We need its word sequence form:

我 爱 苹果(*I love apple*)

Segmenting methods on processing East-Asia languages are applied into amino acid sequence segmentation. Here we segment/tokenise

amino acid sequences by an unsupervised segmentation method⁹. Its main idea is to evaluate the probabilities of all possible words by Expectation Maximization (EM), an iterative method for finding maximum likelihood estimates of parameters in statistical models¹⁰, or other statistical methods like the n-grams language method¹¹. Then for a sequence, we select the segmenting form with maximum probability as the segmentation of this sequence.

An example of parallel corpus about “segmented amino acid sequence” to “function”:

Amino acid sequence (amino acid, segmented):

MTMDKS ELVQKA KLA EQA ERYDDM AAA MKAVTE QGH ELSNEE RNLLSV AYKNVV

Function description (Gene ontology terms ID):

0005737 0006605 0019904 0035308 0042470 0042826 0043085

After segmenting, we also delete the obviously mis-aligned sequences (‘word’ number ratio>9. For above example, there are 10 amino acid words and 7 Gene ontology word. So its word number ratio is 10/7) in corpus.

Translation model

There have been many open source translation systems. Normally, the Moses system is treated as the baseline system¹². Moses is a statistical machine translation system that allows you to automatically train translation models for any language pair. All you need

is a collection of translated texts (parallel corpus). Once you have a trained model, an efficient search algorithm quickly finds the highest probability translation among the exponential number of choices. Here we also use the Moses system to build the translation model for amino acid sequence to Gene ontology terms. Moses offers two types of translation models: phrase-based and tree-based. We selected the phrase-based model.

We used 95% of the data to train the model, and the remaining 5% of the data for testing.

The input for Moses is the parallel corpus. All parameters are set by default values (more details reference in the “data” and “code” sections of the [Supplementary materials](#)). The language model for the Gene ontology sequence is set as 2-grams (in the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sequence of text. For more information, see Xiaping Ge, Wanda Prat, and Padhratic Smyth¹⁰). Then we could get an amino acid sequence to Gene ontology translation model. The detailed process is provided in the ‘code’ directory of [Supplementary materials](#). Its main steps are described as in [Figure 1](#).

We used the Bilingual Evaluation Understudy (BLEU) score to judge the performance of the translation system¹³. BLEU is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. The main idea of BLEU score can be described by formula 1.

Figure 1. Amino acid function translation system.

$$P = \frac{m}{w_i} \quad (1)$$

In this formula m is the number of words from the candidate translation that are found in the reference translation, and w_i is the total number of words in the candidate translation.

For example, a candidate translation is “I air apple”. The reference translation is “I love apple”. There are 3 words in the candidate translation. Two words, “I” and “apple”, appear in the reference translation. So the BLEU score of this translation is $2/3=0.67$.

The improved version of BLEU normally considers the n-grams match. Since we don’t care about the order of Gene ontology words, we only needed to use the unigram form of BLEU.

Because different amino acid sequence segmentation methods will produce completely different “word” sequences, and the segment methods mainly rely on the selection of maximal word length, we tried different maximal word lengths. The relation of maximal word length and the BLEU score are shown in Figure 2.

Figure 2 shows maximal word length of 7 or 8 letters is a good choice for amino acid translation. In theory, we need 20^n letters corpus to train an n word length segmenting model. For $n=8$, we need about 26 Gigabytes amino acid sequence data. For $n=9$, we

Figure 2. The relationship between maximal word length in segmentation and the BLEU score. The red line corresponds to group 1, human dataset. Blue line is group 2, mixed dataset and green is group 3, the big dataset.

need 512 Gigabytes of data. Now there are only about 20 Gigabytes in annotated amino acid sequence data in Gene ontology databases (<http://www.geneontology.org/GO.downloads.annotations.shtml>). So a practical word length selection could be 7.

If we don’t segment the amino acid sequence and treat every amino acid letter as a “word”, 20+ amino acid letters could only express 20+ functions. But from Figure 2, we deduct that we could still get a translation model if we have sufficient corpus. This was mainly because the training system could combine some consecutive letters into “phrases” in the alignment process. So such “phrases” could represent more things.

In Figure 2, we could also find the increasing of the size of corpus could obviously improve the performance of translation. So far now, we tried a 5,000,000 amino acid sequences data set. Its BLEU score could reach 0.6. We also provide the method to build 40,000,000 amino acid sequences corpus data in ‘get_gene_data’ directory of supplied materials.

For a usable translation system, its unigram BLEU score should be more than 0.5. Our amino acid sequence to Gene ontology translation model has reached this level. So we could compare it with conventional search based method.

Comparing with search-based method

The sections above show that the “amino acid sequence” to “protein function” translation is workable. Next, we compared it with current protein functional prediction methods.

We designed the comparison as follows:

- (1) Dataset: we divided the parallel corpus into 95% training corpus and 5% test corpus.
- (2) For the translation-based method: we trained a “amino acid sequence” to “Gene ontology” translation model based on the training corpus and then predicted the functions of a test amino acid sequence based on this translation model.
- (3) For the search-based method: we indexed the amino acid sequences in the training corpus by BLAST to build a training sequences database. To predict the function of the test sequences, we searched the test amino acid sequence in database of training sequences respectively. The Gene ontology description of the best match sequence was regarded as the function of this test sequence.
- (4) Comparing method: we compared the predicted functions of two methods by the BLEU score. The results are shown in Table 1.

From Table 1, we found that the functional prediction performance of the translation method was still not as good as BLAST, but there were about 1%–5% sequences whose functions the BLAST method could not predict. On the other hand, the translation method could

Table 1. BLEU score of search based method and translation method.

	Human data (group 1)	Mixed data (group 2)	Big data (group 3)
Search method	0.41	0.54	0.83
Translation method	0.35	0.37	0.50

predict the functions of all amino acid sequences. Here, we mainly want to show the feasibility of the new method. More datasets and translation methods should be tested in the future.

Summary

Based on statistical machine translation technology, we present a novel method to predict the function of amino acid sequences. Although its performance remains to be improved, it shows that the application of machine translation in protein functional prediction promising. For statistical translation methods, the more corpora we generate, the better results we would obtain. Google's translation system has proven that vast amounts of corpus with a few rules, or even without any rules, can produce excellent translation results¹⁴. Since the protein functional prediction problem is so close to 'language translation', translation based protein functional prediction is likely to be the most

promising approach. Source codes are permanently accessible from [10.5281/zenodo.7506](https://doi.org/10.5281/zenodo.7506).

Author contributions

Wang Liang designed the experiments, wrote the codes and trained the translation model. Zhao Kaiyong prepared the experimental data and provided the biology knowledge supports.

Competing interests

No competing interests were disclosed.

Grant information

The author(s) declared that no grants were involved in supporting this work.

Supplementary materials

There are 5 directories in Supplementary materials.

Directory list:

1. "data": parallel corpus for Gene ontology sequence and corresponding amino acid sequence.
2. "code": codes and directions for building translation model.
3. "get_gene_data": direction for downloading data from geneontology.org (Gene ontology data) and uniprot.org (amino acid sequence data)
4. "pre_process_gene_data": convert the format of original Gene ontology data and amino sequence data to the required format of Moses system.
5. "build_dict": direction for building amino acid dictionary.

1 Directory "/data", parallel corpus

human.pr : human protein sequence (amino acid form) for training, which has been segmented (set maximal word length as 7, same for other gene corpus)

human.go : human gene function description in Gene ontology terms, for training

human.pr.test : human amino acid sequence (amino acid form) for test, has been segmented

human.go.test : human gene function description in Gene ontology terms, for test

mixed.pr : mixed amino acid sequence (amino acid form) for training
mixed.go : mixed gene function description in Gene ontology terms, for training

mixed.pr.test : mixed amino acid sequence (amino acid form) for test, has been segmented

mixed.go.test : mixed gene function description in Gene ontology terms, for test

big.pr : big amino acid sequence (amino acid form) for training

big.go : big gene function description in Gene ontology terms, for

training

big.pr.test : big amino acid sequence (amino acid form) for test, has been segmented

big.go.test : big gene function description in Gene ontology terms, for test

2 Directory "/code"

It contains the source and example to train the gene translation model, we use the mixed data set to train the translation model, include:

readme.txt: direction about how to train the model

Content in readme:

2.1 System requirement

(1) You need install Moses system. <http://www.statmt.org/moses/?n=Development.GetStarted>

(2) You'd better run the baseline of Moses. <http://www.statmt.org/moses/?n=Moses.Baseline>

(3) Python 2.7.

Here we use `irstlm` to train language model, as this baseline suggest.

2.2 Parallel corpus

for train:

gene.6.pr : amino acid sequence, have been segmented, maximal word length 6

gene.6.go: gene ontology sequence

for test:

gene.6.pr.test: amino acid sequence

gene.6.go.test: gene ontology sequence

2.3 Train language model of target sequence, here we train gene ontology model

file: train_lm.sh

(attention: please revise the moses_path and irstml_path to your own path!)

run:
./train_lm.sh gene.6.go
You will get gene.6.go.blm.

2.4 Train translation model

File: train_ts.sh

(attention: please revise the moses_path and giza_path to your path!)

run:
./train_ts.sh gene.6 gene.6.go.blm
Wait about 4 hours (1 CPU. 10+G memory requirement)
you will get a “train” directory.
the train/model/moses.ini is just your translation model file

2.5 Test

File: get_bleu.sh

(please revise the moses_path)

run:
./get_bleu.sh gene.6.go.test gene.6.pr.test
it will translate gene.6.pr.test and then compare it with gene.6.go.test. If there is no error, you will get the result:

BLEU = 26.38, 47.4/30.0/21.4/15.9 (BP=1.000, ratio=1.571, hyp_len=48369, ref_len=30786)

Because we don't consider the order of go terms, 47.4 is just its BLEU score (unigram). Here we didn't run the tuning process. You could divide the test corpora into test part and tuning part, and then run the tuning process. It could bring some improvement for BLEU score, but it may take several days.

3 Directory “/get_gene_data”, get gene data

If you want build your own corpus from original database source (www.geneontology.org, Gene ontology data. www.uniprot.org, amino acid sequence data), see this section:

3.1 Get gene data

Here we will show how to get corpus from original data source. You need install gunzip.

Using human data:

File: get_gene_data.sh, its operation

- (1). get original gene ontology data. (ftp of www.geneontology.org).
- (2). get amino acid sequence data. (ftp of ftp.uniprot.org).
- (3). gunzip
- (4). convert amino acid sequence format. (get_gene.py)
- (5). convert gene ontology format. (get_go.py)
- (6). Get parallel corpus according to gene database id. (get_corpus.py)

just run ./get_gene_data.sh
We will get parallel corpus
human.pr.filter gene sequence
human.go.filter corresponding gene ontology sequence
This corpus contain about 1,8000 data.
You could also download the big corpus:

Gene ontology files (UniProt [multispecies]):

http://www.geneontology.org/gene-associations/submission/gene_association.goa_uniprot.gz

contain about 40 million data.

Related sequence data could be found at:

UniProtKB/Swiss-Prot,

ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fasta.gz

UniProtKB/TrEMBL,

ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_trembl.fasta.gz

You could download these two files then merge (normally Linux shell 'cat' command) them together. Then revise the ./get_gene_data.sh. (refer to get_big_gene_data.sh). Finally, you could get about 20 million parallel corpus.

4 Directory: “pre_process_gene_data”, segment amino acid sequence and clean the corpus

If you want to build your own corpus, you will need segment the amino acid sequence and clean the corpus.

4.1 Data files

- (1) mix.go original gene ontology sequence
- (2) mix.pr original amino acid sequence
(Here we give these data. You could also copy the human.pr.filter in previous to mix.go, and human.go.filter to mix.go to run your own experiment, also for the big data set)
- (3) protein.dict dict file

4.2 Preprocess the file, get the parallel corpus

File : pre_process.sh

just run:

./pre_process.h

Its operation:

- step 1: segment the amino acid sequence. (segment_gene_sequence.py)
 - step 2: clean the corpus. (clean_corpus.py)
 - step 3: divide the corpus into train and test parts. (divide_corpus.py)
- finally, you will get 4 files

gene.pr : amino acid sequence, have been segmented, maximal word length 6

gene.go: gene ontology sequence

gene.pr.test: amino acid sequence for test

gene.go.test: gene ontology sequence for test

These files could be used to train the translation model.

5 Directory “build_dict”, build amino acid dictionary

Moreover, if you want to build your own “protein.dict” for different maximal word length, we also give an example:

Step 1, install SRILM

A simple method to build dictionary is to use language model. Here use SRILM. You should install it first. (<http://www.speech.sri.com/projects/srilm/>)

Then cp the executable file “ngram-count” to this directory. Normally in your install dir. There have been an “ngram-count”, but it could only run in a special Linux version, so just overwrite it.

Step 2, you should make 2 file

- (1) In ./get_gene_word: run make
- (2) In ./get_gene_word_prob: you should set the srilm install path and MACHINE_TYPE in Makefile, then run make.

Step 3, uncompress the data file

Run: tar -xvzf protein.fa.tar.gz

Step 4, train n-gram language model, n=1–5

Run: ./build_all_lm_model.sh

(Attention, you must cp “ngram-count” to this directory. See step1)

Step 5, Build dictionary with different maximal word length

Run: ./build_dict_all.sh

Its operations:

- (1) Get all possible gene word, filter them by frequency. (get_word)
- (2) Get probability of all gene words. (get_prob)
- (3) Filter the gene words in dictionary by MI method. (mi_filter.py)

Finally, you will get protein.dict.*.mi, (* is 1,2,...,5) is the dictionary with maximal word length *. Then you could use these dictionary file to segment the amino acid sequence (in section 3).

If you have map/reduce cluster, you could use EM method to build the gene word. See our open source project (<https://code.google.com/p/dnasearchengine/>). To train an 8 maximal word length dictionary, you need at least 4G amino acid sequence data. More amino acid sequence could be found in “Get gene data” section or ftp of gene Refseq databases. <ftp://ftp.ncbi.nih.gov/refseq/>.

References

1. Hawkins T, Chitale M, Luban S, *et al.*: **PFP Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data.** *Proteins*. 2009; **74**(3): 566–82.
[PubMed Abstract](#) | [Publisher Full Text](#)
2. Guan Y, Myers CL, Hess DC, *et al.*: **Predicting gene function in a hierarchical context with an ensemble of classifiers.** *Genome Biol.* 2008; **9**(Suppl 1): S3.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
3. Pavlidis P, Gillis J: **Progress and challenges in the computational prediction of gene function using networks [v1; ref status: indexed, <http://f1000r.es/SqmJUM>].** *F1000Res*. 2012; **1**: 1–14.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Eisen JA: **Phylogenomics: Improving Functional Predictions for Uncharacterized Genes by Evolutionary Analysis.** *Genome Res.* 1998; **8**(3): 163–167.
[PubMed Abstract](#) | [Publisher Full Text](#)
5. Whisstock JC, Lesk AM: **Prediction of protein function from protein sequence and structure.** *Q Rev Biophys.* 2003; **36**(3): 307–40.
[PubMed Abstract](#)
6. Ashburner M, Ball CA, Blake JA: **Gene Ontology: tool for the unification of biology, The Gene Ontology Consortium.** *Nat Genet.* 2000; **25**(1): 25–29.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
7. Brown PF, Cocke J, Stephen A, *et al.*: **A Statistical Approach to Machine Translation.** *Computational Linguistics.* 1990; **16**(2): 79–85.
[Reference Source](#)
8. Chiang D: **Hierarchical phrase-based translation.** *Computational linguistics.* 2007; **33**(2): 201–228.
[Publisher Full Text](#)
9. Liang W, Kaiyong Z: **Segmenting DNA sequence into ‘words’.**
[Reference Source](#)
10. Ge X, Prat W, Smyth P: **Discovering Chinese words from unsegmented text.** *Proceedings on the 22 Annual International ACM SIGIR Conference On Research and Development in Information Retrieval. Berkeley CAUSA.* 1999; 271–272.
[Publisher Full Text](#)
11. Rosenfeld R: **Two decades of statistical language modeling: where do we go from here?** *Proceedings of the IEEE.* 2000; **88**(8): 1270–1278.
[Publisher Full Text](#)
12. Koehn P, Hoang H, Birch A, *et al.*: **Moses: Open Source Toolkit for Statistical Machine Translation.** *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic.* 2007; 177–180.
[Reference Source](#)
13. Papineni K, Roukos S, Ward T, *et al.*: **BLEU: a method for automatic evaluation of machine translation.** *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics.* 2002; 311–318.
[Publisher Full Text](#)
14. Mikolov T, Chen K, Corrado G, *et al.*: **Efficient Estimation of Word Representations in Vector Space.** *In Proceedings of Workshop at ICLR.* 2013.
[Reference Source](#)

Open Peer Review

Current Peer Review Status: ? X X

Version 1

Reviewer Report 22 October 2014

<https://doi.org/10.5256/f1000research.2746.r6146>

© 2014 Bernardes J. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Juliana Bernardes

Programa de Engenharia de Sistemas e Computação, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

The paper present a new method to predict protein function based on machine translation concepts. In this scenario, protein sequences and their gene ontology terms are interpreted as words in different languages. The idea is to find the best translation for a protein sequence, that is, its most likely gene ontology terms. The method has two phase: training and inference. During the training, protein sequences are broken into several segments and then these segments are related to gene ontology terms. This mapping between protein segments and gene ontology terms constitutes the parallel corpus and it is learned during the training phase. Next, the inference is carried out by an statistical machine translation system that translates a new protein sequence into gene ontology terms by using the collection of translated pairs (parallel corpus) learned previously.

My main issue with this paper is the way how sequences are broken into segments. It is well known that some segments such as motifs and domains are essential to predict protein functions. The author must explore them to enrich their model.

A second point concerning the results, they are disappointing. The method did not outperform BLAST that is one of the simpler existing method for protein function prediction. The comparison was not properly done, what is the false positive rates for both methods? Authors must compare the methods by using classical measures such as precision, recall and F-measure and a ROC curve could be plotted. Moreover, the authors did not compare their approach with state-of-art methods such as SVM and HMMs. The discussion of results is really poor, the authors did not defend their method.

The training and testing strategy is unusual. 95% of the data to train and just 5% to test, the author have performed a cross validation? If yes how many folds? how many times it was repeated? It is not clear in the text.

The paper is not well written and there are several incompressible parts. Just to cite some of them:

- "Then we use a phrase-based translation model to build the "amino acid sequence" to "protein function" translation model. (I think it is "to map" instead of "to build")
- Here we build three groups of gene parallel corpora. Group 1 only contains human amino acid data. (the expression "human amino acid data" is so unusual in bioinformatics community).

The author wrote in the conclusion "Since the protein functional prediction problem is so close to 'language translation', translation based protein functional prediction is likely to be the most promising approach."

I do not agree that protein functional prediction is close to language translation problem. There are a lot additional points, for example evolution. In my opinion, we cannot interpret proteins as words. Anyway, the results have not showed that it is the most promising approach.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Reviewer Report 20 October 2014

<https://doi.org/10.5256/f1000research.2746.r6149>

© 2014 Verspoor K. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Karin M. Verspoor

Department of Computing and Information Systems, University of Melbourne, Melbourne, VIC, Australia

This article provides an intriguing approach to protein function prediction -- to treat the task as a word translation process. However, the results that the authors provide do not justify the statement that the approach is likely to be "the most promising approach" and are not appropriately compared with state of the art methods. Furthermore, no biological intuition appears to have informed the approach; this results in an approach that, while novel, doesn't capture some fundamental characteristics of the data.

Protein sequence - function relationship: The fundamental assumption embodied by the method is that there is a strong relationship between protein (sub-) sequences and functions, as represented by sequences of GO terms. While this is likely true for some sub-sequences, it won't be true for all, as the authors themselves point out.

- Does the method allow for some sequence "words" to be effectively silent in the prediction phase? Figure 1 shows each word being included in some phrase which is "translated". Can some words remain untranslated? How often does this happen for this task?

- What about sequence words that should correspond to multiple GO terms?
- Is the sequence of words biologically important? The "Best BLAST" method that the authors compare with (NB: I wouldn't call that approach "search-based" as that name is extremely underspecified; rather I would suggest "sequence similarity-based") uses *local* sequence similarity; some regions of the sequence will not be considered.
- It certainly isn't obvious to me that linear order should matter at all on the output side; it appears that the GO sequences are listed in numeric order by ID and a "2-gram language model" over GO sequences should not carry any more information than a 1-gram model would, as the IDs themselves are arbitrary. This representation is not well-justified.

Segmentation: The authors use a EM-based approach to drive segmentation, which essentially captures the statistical properties of particular subsequences. There should be some relationship to work on motif finding here; the authors should identify some relevant work on that topic (e.g., Tompa *et al.* ¹, although there are likely many more reviews available; as this is not my area specifically I suggest the authors do their own search). The authors should also compare to a far simpler segmentation approach: just splitting the sequence into words of fixed length. This is a character-level n-gram representation. Perhaps all the work of running EM (and the associated large number of data points) is completely unnecessary. (The authors would of course need to try a few different lengths for this to be meaningful.) Essentially, the strategy needs to be justified, and the impact on the broader function prediction task must be quantified. What is the relationship between the maximal word length and the number of output terms included in the output GO sequence? Is there a length bias on the *output* side that is also reflected in Figure 2? Again, this hinges on the appropriateness of characterising this task in terms of the translation model.

Data: The authors do not clearly state whether they consider all GO annotations available for a protein, or if non-experimental evidence codes are excluded. If "Inferred by Electronic Annotation" codes are included, this means that "Best BLAST" has likely already been used to produce the annotations, which would explain at least to some extent why the performance of that method shows such strong results.

Experiment details: The authors do not specify what e-value threshold they assume for BLAST.

Filtering: The authors incorporate various filters along the way, and it is unclear how strongly this biases the training/test data that they utilise. The authors delete long sequences and proteins with many associated GO terms. The cleaning reduces the data set substantially, but the rationale for doing this cleaning appears to have more to do with the limitations of the approach that the authors have adopted than because of some biologically valid reason. The exclusion of such a large number of {sequence, GO annotations} pairs will certainly undermine the findings. This also applies to the deletion of the "obviously mis-aligned" sequences. How many instances does this filter? Why are these obvious mis-alignments? The number of GO terms associated with a sequence has more to do with (a) annotation bias [some sequences are far more heavily studied/annotated than others] and (b) overall progress on curating protein function information (see Baumgartner Jr. *et al.* ²).

Evaluation: Recently, the CAFA challenge explored protein function prediction from sequence ³, and in association with that there has been detailed consideration of evaluation of this task ^{4 5} -- the key point being that the structure of the GO should be taken into consideration in the evaluation. The authors present their results in terms of BLEU, but in their use of the unigram

form, I think that BLEU simply collapses to Precision. This, then, ignores Recall completely. The evaluation presented should certainly measure Recall, and the authors should provide results that are comparable to the results of systems that participated in CAFA. They should then compare their results to the state-of-the-art methods.

Summary

In summary, the applicability of the method to this task should be more strongly justified and the experimental set-up (including the extensive filtering of data points and the evaluation metrics) should be more carefully considered. The authors' conclusions have not been adequately justified on the basis of their results.

References

1. Tompa P, Davey NE, Gibson TJ, Babu MM: A million peptide motifs for the molecular biologist. *Mol Cell*.2014; **55** (2): 161-169 [PubMed Abstract](#) | [Publisher Full Text](#)
2. Baumgartner Jr WA, Cohen KB, Fox LM, Acquaah-Mensah G, et al.: Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics*.2007; **23** (13): i41-i48 [PubMed Abstract](#) | [Free Full Text](#) | [Publisher Full Text](#) | [Reference Source](#)
3. Radivojac P, Clark WT, Oron TR, Schnoes AM, et al.: A large-scale evaluation of computational protein function prediction. *Nat Methods*.2013; **10** (3): 221-227 [PubMed Abstract](#) | [Free Full Text](#) | [Publisher Full Text](#)
4. Clark WT, Radivojac P: Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics*.2013; **29** (13): i53-i51 [PubMed Abstract](#) | [Free Full Text](#) | [Publisher Full Text](#)
5. Verspoor K, Cohn J, Mniszewski S, Joslyn C: A categorization approach to automated ontological function annotation. *Protein Sci*.2006; **15** (6): 1544-1549 [PubMed Abstract](#) | [Free Full Text](#) | [Publisher Full Text](#)

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Reviewer Report 04 November 2013

<https://doi.org/10.5256/f1000research.2746.r2296>

© 2013 Pavlidis P. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Paul Pavlidis

Centre for High-Throughput Biology and Department of Psychiatry, University of British Columbia, Vancouver, BC, Canada

This paper describes a method for predicting gene function based on amino acid sequence, using

a method inspired by automatic translation of human language. The idea is to break the sequence into segments, and each segment is assigned a GO term. During training the association between segments and GO terms is learned, and then this learned model is applied to new sequences. In a benchmark against BLAST performance it is much worse at identifying the GO terms associated with a protein.

The defence offered that the work shows the feasibility of the approach falls a bit flat for me. It is feasible in the sense that *"it can be attempted"*. I do not find the results encouraging. I disagree with the claim that protein function prediction is *"so close"* to language translation, except in some rarefied technical sense. Indeed, using translation seems to be obscuring the actual task, which (in this context) is to find sequences that are similar and transfer GO annotations from one to the other. At the heart of it, that is all that is happening here.

I know I am not supposed to address novelty, but I have to point out that many feature extraction methods have been proposed for sequence analysis (for example, string kernels combined with SVMs¹). Many methods for sequence scoring are based on methods also used for analyzing human language (e.g. HMMs²), but without the baggage of *"translation"*. The authors do not make these connections in the introduction.

In addition:

The method is not described in enough detail, especially for bioinformatics researchers who may not be familiar with the translation methods (including myself). The provision of source code, while laudable, is insufficient. For example, for the segmentation method, the reader is directed to a non-peer reviewed manuscript on DNA segmentation. Why does each segment get at most one GO term? If that is necessary, it seems a big problem, since GO does not work that way. BLAST also does not have the limitations that are exposed by the requirements for cleaning described here. What makes a protein sequence *"corrupted"* or *"inaccurate"*? I do not get what the authors mean by *"obviously misaligned"* in comparing GO terms to protein sequences; it makes little sense and the strained relationship of the task to translation is exposed. Then authors seem to be requiring exact matches; this is also not very biological. It may be that these are fundamental limitations of the approach; if not, they must be addressed. (I did not take the time to review the source code).

The results are not described in enough detail and the evaluation is weak. By optimizing the word length on the same corpus used for testing, the authors have probably inflated the performance scores, not that it helps much. The authors claim that an advantage of the method is that it always gives a prediction, even when BLAST does not. Is it really reasonable to trade specificity for sensitivity in this way? Can the authors show any cases where this yields a useful answer? Showing some cases where a correct functional annotation was found by this method, but not by BLAST, and providing some insight into why that happened, is necessary. The authors mention in the introduction that sequences that are similar may not have the same function; are they claiming that this new method addresses this? I do not think so, so it seems irrelevant.

Given all these issues, I nearly checked the *"not approved"* box. But on consideration I think the work could be fixed up, from a technical standpoint.

There are also quite a few English language errors.

References

1. Leslie CS, Eskin E, Cohen A, Weston J, et al.: Mismatch string kernels for discriminative protein classification. *Bioinformatics*. 2004; **20** (4): 467-476 [PubMed Abstract](#) | [Publisher Full Text](#)
2. Eddy SR: Hidden Markov models. *Current Opinion in Structural Biology*. 1996; **6** (3): 361-365 [PubMed Abstract](#) | [Publisher Full Text](#)

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 05 Nov 2013

亮王, Tencent Tech, Beijing, China

We thank the referee for their suggestions.

In next version of this paper:

1:

We will give a more in depth introduction to machine translation and segment technologies. We will also add detail as to why we applied translation to protein functional prediction. Although these technologies are well known in the area of natural language processing, it is completely new for most researchers in bioinformatics.

I have written a short PPT for sequence segmentation:

<http://www.slideshare.net/maris205/segmenting-dna-sequence-into-words>

We will also build an online demo for this work and its codes will also be uploaded.

<http://www.dnasearchengine.com/trans>

Most current machine translation researches are based on the work of IBM in 1990:

Brown *et al* [Statistical Approach to Machine Translation](#), *Computational Linguistics*. 16(2), 79-85(1990).

This translation model (IBM model 5) mainly contains 5 steps: word alignment, phrase extraction, language model training, decoder training, and evaluation.

The translation model also uses the string features but it builds the phrase corresponding model (one phrase corresponds to a phrase list with probability, not one to one), but not the classification model like SVM. Besides from language translation, translation models are also applied in query rewrite, multi-label classification and many other areas. Because protein functional prediction could be regarded as a kind of multi-label classification problem (one

sequence to several GO terms), we argue that one could use translation methods in protein functional prediction.

2:

The referee requests more evaluations - There still seems to be no standard data set and evaluation methods for protein functional predictions. We will find a proper data set and try different evaluation metrics. More available function prediction methods will also be tried.

Competing Interests: No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research