Proteins are the building blocks of humans. It plays an important role in all biological processes, for example the use of haemoglobin for oxygen carrying. The structural features and 3D shape of the protein determines the functions of these proteins. Knowing the functions of these proteins is useful for drug development because it allows the design of specific drugs to treat the patients. As different people will have different responses to drugs due to genetic difference, exploring the function of different proteins can help realize personalized medicine by giving specific treatment.

However, there are several challenges in predicting the functions of proteins. The first one is sequence similarity. Some proteins show similar functions but actually have larger difference in the amino acid sequences. In contrast, small differences in the sequence can actually lead to different functions of protein. Besides, some proteins do not have easily identified features for some type of functions. Lastly, many proteins can have more than one function, causing the prediction to be more complicated.

The above reasons limit the accuracy of the prediction and only less than 1 % of protein sequences have determined functions. It is also difficult to identify rare protein functions because these functions are underrepresented in existing datasets.

To solve this problem, Bioinformatics is a great tool for consideration. By utilizing computer science to handle large amounts of bio-related information. For instance, motif-based methods to predict protein binding sites or deep learning algorithms to train existing data and predict protein features. Gene ontology provides a structured vocabulary of terms for describing gene and protein functions.

[KA]

The first model we’ll be looking at is called DEEPred.

The researchers wanted to create a pipeline that can accept amino acid sequences and output GO terms.

But a raw sequence is too much information, right? And it’s not very useful information. So they want to extract the most important features from the sequence to use as the actual input for the model.

And they did that by incorporating a preprocessing step, where they created feature vectors, and they tried 3 methods.

The first one is called the conjoint triad feature, which is where each amino acid is assigned a class, 1 to 7, red to blue. And then we go through every triplet in the amino acid sequence and record their frequency.

The second method is called PACC, and it encodes the spatial information of the amino acid neighbours, the next neighbour, and the next-next neighbour etc.

The third one called sub-sequence profile mapping, which is to split up the sequence into sub-sequences and do hierarchical clustering on them.

They found that SP Map gave the best results, so they used that to generate the feature vectors for all inputs.

So now the next step is to feed it into a neural network.

The architecture of DEEPred is a simple feed-forward multi-task deep neural network. As you can see on the left, that’s our preprocessed feature vector as the input, and on the right it will predict the probability or confidence for 5 GO terms at the same time.

But there are a lot more than 5 GO terms in total,

so DEEPred is actually a stack of 1000 of this kind of DNN, each responsible for 5 terms. They chose 5 because using more gave worse performance.

But they didn’t decide randomly what each DNN should predict for. Because if the 5 terms have different broadness, which means if one of them can apply to many of the data points than the others,

then the model can just always output that broad term with a high confidence,

then it can achieve a high accuracy without learning at all.

But if every term has a similar broadness,

Then the same guessing approach doesn’t work, and we can force the model to learn.

So they divided the GO terms into different levels based on their broadness, and make sure each DNN is only responsible for predicting 5 terms in the same level so they can’t cheat.

They ended up only using 10 levels maximum because they found a diminishing return for using more levels.

For the results, the model achieved the highest F1 score in molecular function of 83%, but only around 55-65% in the other categories.

Compared to other models, DEEPred is among the top 3 in every category, and it also has the highest overall performance in the benchmark when combined.

The main advantages or reasons why DEEPred performs well are:

1. the authors actually did 100,000 automated hyper-parameter optimisation to find the best one,
2. the training data include both experimental low-noise data as well as electronic high-noise data, which makes the model noise-tolerant,
3. it is also scalable since training many small models to each predict only 5 terms in parallel is faster and more accurate than training one very very big model to predict every term. And we can also retrain each part independently to optimise the performance.

[KA]

Now onto the next approach, which is called DeepGraphGO.

It has a completely different architecture type than DEEPred.

But first of all, similar to DEEPred, there is also a processing step to generate feature vectors before feeding them into the network. But DeepGraphGO uses a much more complicated database called InterProScan to do this, which basically includes information about protein domains, protein families, functional sites, and motifs.

The bulk of the DeepGraphGO model architecture is something called a graph neural network, GNN, and it takes the feature vectors as an input and again output GO terms with confidence values.

If we look at the architecture diagram,

on the left once again we can see the processed InterProScan feature vectors as the input.

Then we have a fully connected layer that condenses the information into a low-dimensional representation,

which is transformed into a graph, and then we have two graph convolutional layers.

But what *IS* a graph, right?

This is an example of a graph, with 4 nodes, ABCD. In our case, they will represent protein A, protein B, protein C, protein D. And then there are edges, which are the lines between the nodes, and they represent some function or interaction relationship.

So mathematically, we can encode this into a matrix with 1 meaning connected and 0 meaning not connected.

But in reality, the edges are weighted, so they have a number instead of 1 and 0, like this.

And the edges are directed, so when you travel in one direction, it’s positive, and in the other direction, it’s negative.

So going back to our graph convolutional layers, they are just layers that turn one graph into another.

And finally on the right, there is the output layer with confidence for each GO term.

But the thing that makes DeepGraphGO special is that,

it actually has a special mechanism for transforming the low-dimensional representation into a graph.

And they used protein-protein interaction, or PPI, networks as the foundation for the graph layers. They used 7 types of PPI networks from a database called STRING.

And multiply everything by 17 species.

The major advantages are,

1. it is able to predict protein functions in many species since it uses multi-species information in the model, compared to other GNNs where every species need to train its own model.
2. it is easily expandable by transfer learning onto a larger PPI network,
3. it has a lot more context using the PPI network than only the amino acid sequence so it can infer unseen proteins from similar ones to achieve better performance.

And in fact, from the results, we can see that DeepGraphGO managed to achieve the best scores among other models. And it is able to comprehensively give all GO terms without any missing while other models missed the more difficult terms.

And even for difficult proteins and multispecies prediction, DeepGraphGO scores the highest due to its ability to infer function from similar data points and its multi-species network.

Despite this, there is also an inherent limitation of GNNs, which is they are very slow to train.

[LOUIS]

Now, let’s move on to the final approach which is DeepFRI. It’s a graph convolution network for predicting protein functions by extracting features from sequences and protein structure. DeepFRI covers two stages. The first stage is a self-supervised language model with long short-term memory. The language model is first pre-trained on a set of protein domain sequences form protein database. The model acts as a sequence feature extractor for the second stage of DeepFRI. The second stage is a graph convolution network to propagate the residue-level features and construct final protein-level feature representations.

We can compare DeepFRI with other methods including sequence-based methods and deep learning methods to investigate its performance.

From figure a, we can see that DeepFRI has a higher protein-centric F score which outperforms other methods in predicting molecular function and biological process of proteins.

From figure B, we can see that DeepFRI robustly predicts MF-GO terms of proteins with low sequence identity to the training set, meaning DeepFRI learns structure-function relationships more robustly than other methods.

Then, we explore the performance of DeepFRI on training different models. From figure c, we can see that DeepFRI has higher performance for native structures, DMPFold models and Rosetta models in terms of the F score. Besides, DeepFRi exhibits significant denoising capability because of a high correlation between graph convolution network features extracted from contact maps.

By comparing the MF-GO terms with different methods as shown on figure d, we can see that DeepFRI outperforms other methods for more MF-GO terms with fewer training samples.

Let’s see an example of mapping function prediction to sites on protein structure. Figure e shows the identified residues for a calcium ion binding of a rat protein. The two highest peaks are the calcium-binding residues in the structure of the protein.

Although DeepFRI can identify some site-specific predictions, it still has some limitations. The below figure from supplementary material shows us that it has lower performance for unseen protein models, which limits the accuracy of rare protein functions. Another limitation is that DeepFRI cannot fully capture long-distance structural correlations within proteins which restrict the depth of the networks.

Through the mention of these three approaches, we understand the significant role of deep learning in protein function prediction. It can lead to faster, scalable and higher accuracy of prediction. Another example is proposed by Professor Yu Li in CUHK. The paper, which is Fast, sensitive detection of protein homologs using deep dense retrieval was published on *Nature biotechnology* in 2024. In simple words, the first step of the method is to convert protein sequences into a special "vector" using a protein language model. This vector summarizes important information about the protein, like its structure and function. To find proteins that are related, the method compares the vectors of the protein which you're searching for with the vectors of proteins in a database. If two vectors are similar, it means the proteins are likely homologs. This method skips alignment and just compares the vector representations. This makes it much faster and able to find distant homologs. The model is under contrastive learning to make related proteins have similar vectors and unrelated ones have very different vectors. This helps improve the accuracy.

To conclude, we have identified some problems in protein function prediction and explain some advanced methods that try to solve the problem. We can see that there is a trend in deep learning method which have been shown to perform better than sequenced-based method.

Q & A:

1. *What is the main advantage of using deep learning frameworks like DeepFRI over traditional sequence-based methods such as BLAST?*

Answer:

Higher accuracy: They can predict functions for proteins with low sequence identity (<30%) to the training set, which traditional methods struggle with.

Structural insights: They incorporate both sequence and structural data, allowing for better prediction of molecular functions and biological processes.

Scalability: They can handle large datasets and complex protein functions more efficiently.

2. *How does the integration of contact maps in DeepFRI improve protein function prediction accuracy?*

Answer:

Contact maps provide structural information about the spatial relationships between amino acid residues in a protein. By integrating these maps, DeepFRI can:

1. Capture **structural dependencies** beyond just sequence information.
2. Identify **functional sites** more accurately, as these are often related to specific 3D arrangements of residues.
3. Enhance prediction performance for proteins with similar structures but low sequence similarity.

3. *What role does contrastive learning play in improving protein function prediction, and how is it different from traditional methods?*

This approach is particularly useful for proteins with low sequence similarity but similar functions.

1. **Encoding protein sequences into high-dimensional vectors** that preserve functional and structural similarities.
2. **Skipping alignment**: Unlike traditional methods like BLAST, it compares vector representations directly, making it faster and more scalable.
3. Enhancing accuracy by focusing on **learning meaningful protein representations** instead of relying solely on sequence alignment.

4. *Why is protein function prediction more difficult for multi-functional proteins, and how might future methods address this?*

Answer:

Multi-functional proteins perform several distinct functions, often depending on context (e.g., cellular environment or interacting partners). This complexity makes it challenging to assign a single function. Future approaches could address this by:

1. Using **context-aware models** to predict functions based on environmental or interaction data.
2. Incorporating **multi-label classification** techniques to assign multiple functions with confidence levels.
3. Leveraging **dynamic structural simulations** to understand how proteins behave in different functional states.