# CNNs vs. HMMs: A High-Speed Showdown for Protein Binding

**Julia Holz (jholz)**
**John Cost (jcost)**
**Yucheng Shao (yshao3)**

## Abstract

Proteins are an essential element of biology, performing most functions within the cell. Understanding their structure and function is paramount in various fields such as drug discovery, structural biology, and beyond. Central to this understanding is the identification of protein domains, which are the functional units of proteins responsible for their behavior. Traditionally, protein domain recognition has relied on methods that suffer from drawbacks in either efficiency or accuracy. This paper presents a novel approach to protein domain recognition (PDR) applying Convolutional Neural Networks to protein sequence data.

## 1 Introduction

There is potential for a method of identifying Protein Binding Domains that is both fast and accurate. Identifying protein domains involves mapping domain signatures onto protein sequences to determine the locations of discrete functional units. Seemingly applicable methods such as sequence alignment fall short due to their inability to penalize errors differently based on their position in the sequence, which is a key feature of an effective protein domain detection method because we need to penalize errors in regions that we expect to be highly conserved more harshly. Moreover, existing tools such as HMMER, while accurate, suffer from inherent computational inefficiency, particularly when dealing with large datasets. In particular, HMMER, as its name implies, uses a hidden Markov model to detect protein domains, and this necessitates using the Viterbi decoding alogorithm, which is quite slow. In forming the dataset for training our model, we ran HMMER to search for one domain amongst about 107,000 sequences, and found that it took about one third of a second per sequence. While this may sound relatively fast, this limitation becomes increasingly problematic given the vast amount of protein sequence data available, with databases like UniProt containing tens of millions of protein sequences.

### 1.1 Inspiration from Object Recognition

Recognizing the parallels between protein sequences and image data, this paper assesses the viability of techniques from the field of object recognition to the problem of protein domain identification. Object detection has seen significant advancements in recent years, particularly with the widespread adoption of CNNs. By applying CNNs to protein sequences, the latest advancements in image recognition techniques may potentially be leveraged for the problem of PDR. In particular, we would like to acknowledge that we took inspiration in designing our CNN model from the object detection method proposed in the paper "Objects as Points" (Zhou et. al., 2019).

### 1.2 Spatial vs. Sequential Data

Fundamentally, images represent spatial data, and amino acids represent sequential data (amino acid sequences). However, convolutions are still an applicable technique for protein domain detection,

because in image recognition we want to extract local features from a larger image to identify objects and in domain detection we want to extract local features from a protein sequence to identify specific domains.

### 1.3 CNNs

The fundamental goal remains the same: distinguishing distinct motifs or regions within the data. Using CNNs for protein domain detection offers several potential advantages over traditional methods and existing tools. The time complexity of CNN convolution is linear in the input size. This represents a potential time complexity advantage relative to HMM approaches which make use of the Viterbi decoding algorithm (or another similar algorithm). Thus, a CNN approach could significantly enhance the speed of PDR, but, given enough training data, it also holds the promise of improving accuracy by capturing nuanced patterns within protein sequences missed by competing approaches. However, one key point that must be acknowledged is that, practically, for HMM models to be faster, they must have relatively small convolutional kernels (to reduce the size of the matrix multiplications being performed) and a small number of layers, since HMMER has been highly optimized, and beating it in terms of speed requires a lightweight model.

### 1.4 Structural Alignment

Structural alignment has been used for motif recognition, where the alignment algorithm itself can be done in polynomial time (Singh and Saha). For any conserved motif, its structure should be relatively conserved for a common / similar function. Therefore, even if the amino acid sequence is not well conserved, the overall structure should still be relatively conserved to be considered the same motif (Illergård et al., 2009). In particular, it is expected that two proteins (or their sub-sequences) will have highly conserved structures if they share 70 percent sequence identity. However, if the sequence identity falls below 30 percent, the structural conservation is no longer guaranteed (Ding and Dokholyan, 2006). Therefore, it might be reasonable to use structural alignment for samples that other methods (like CNN) failed to produce confident predictions for, and those which have poor amino acid alignments (sequence identity). This process is expected to give highly accurate prediction of both the existence and the location of the motif based on RMSD (root mean squared deviation, the standard metric for structural alignment, Singh and Saha) and location aligned.

## 2 Methods

### 2.1 Training Data

Our approach made use of the slow-but-accurate tool HMMER to generate training data in order to train a convolutional neural network to detect the locations of domains within the protein sequence. Our initial set of protein sequences comes from the PANTHER database, a curated database of gene and protein families (Mi et. al., 2005). It consists of approximately 107,000 partial and full-length protein sequences, drawn from the family, PTHR45527, of nonribosomal peptide synthetases. The two domains we decided to search for were domains that are known to be included in many nonribosomal peptide synthetases, PF00668, a condensation domain, which catalyzes formation of the peptide bond during nonribosomal peptide synthesis, and PF00501, which is an AMP binding domain, where the synthetase binds AMP, which is bound to the amino acid substrate during the synthesis process. Both of these domains, and their seed alignments, which we fed into HMMER, come from the PFAM protein family database (Punta et. al., 2012).

Once we had our two protein domains and our nonribosomal peptide synthetase sequences, we used HMMER to generate our training dataset. We used HMMER with the default parameters to find all hits for each domain in the protein sequences, and we made use of the pyHMMER API Pipeline7 with our two domain seed alignments to perform our HMMER queries (Laralde and Zeller, 2023). Then, we tried two different methods for identifying domain keypoints in the protein sequence based on HMMER's output. The first method was to simply label the center of the domain (halfway between

the predicted envelope start and envelope end given by HMMER) as a keypoint, and a second method, which we tried after realizing that insertions and deletions could cause the "center" of the domain to be shifted to different points of the sequence and that the center of the domain is not guaranteed to be a well-conserved region.

For the second method, we used the consensus sequence from our HMM, and for each domain found the length-10 window that contained the most "highly conserved" residues as classified by HMMER. Highly conserved residues are considered to be those with an emission probability of at least 50 percent by HMMER, and we selected a window size of 10 amino acids, since we found that increasing the window size beyond 10 did not increase the number of highly conserved residues within the window beyond what was found with 10. The number of conserved residues in the window was 5 for domain 00668 and 10 for 000501, and remained at 5 for 00668, and only increased to 11 for 00501 when we increased window size to 30, so we decided to use the 10 window size most-conserved regions as keypoints for each motif. Then, using the local multiple sequence alignment yielded by the hmmsearch, for each hit's protein sequence, we mapped the location that was aligned to the center of this size-10 window to its index in the original protein sequence to give us the location of the conserved window in the protein domain.

### 2.2 Model Training

Once we had our set of sequences with hits for each domain, and locations of keypoints within those domains, we needed to convert these sequences and keypoint locations to inputs and targets for our model. Converting the sequences to inputs was easy enough. We encoded sequences as vectors both by trying a one-hot encoding, and by assigning each amino acid a number in the range 1 to 20. We turned the keypoint locations into targets for our model using a method inspired by the method used in the image object detection paper, "Objects as Points". As they did for objects in the paper, we "splatted" the keypoint for the domain over the vector using a gaussian kernel, adapting their method from two dimensions to one, using the following formula:

$$V(i) = e^{\frac{-(i-c)^2}{2\sigma^2}}$$

adapted from Zhou et. al., where $V$ is our target vector, $i$ is the index in that vector, and $c$ is the location of our center (Zhou et. al., 2019). We used this dataset of approximately 70,700 hits for domain PF00668 and 85,700 hits for domain PF00501 to train distinct models for each of the two domains, using an 80/20 train/test split.

Then, it came time to design and train our model. Based, once again, on the method used in the paper "Objects as Points", we used sigmoid focal loss to train our model. Focal loss is typically used by object detection methods because it helps deal with the issue of sparseness of objects/object centers in the image. This is applicable to the protein domain detection problem, because we have relatively few domains/domain centers as compared to non-domain points.

We tried several different architectures for the model, each with different numbers of convolutional layers, linear layers, and different Gaussian sigmas (higher sigmas "splatted" out the keypoints over a larger area of the target vector). In building our model, we made use of PyTorch's nn.Conv1D layers for our convolutions, and nn.Linear layers (Paszke et. al., 2017).

### 2.3 Structure File and Alignment

To test the accuracy of structural alignment for predicting the existence of motifs, structural information is extracted from the UniProt database by searching both the PF00501 and PF00668 families and downloading the PDB files. Due to the limited availability of X-ray crystallography and NMR data, all PDB files used are AlphaFold prediction results. It is expected that the predicted structures can be used as inputs, otherwise, any structural alignment would require experimentally determined structures and would be completely impractical. From the HMM determined motifs, we chose one that had the minimal length and searched it against the PDB database. We chose the best match structure (6p1j) and used pymol to save a copy of the substructure based on sequence alignment (65

percent sequence identity, expected to have highly similar structure) to use as a reference structure. We used the pymol align function to carry out the alignment. This function first does a sequence alignment between the reference structure of the motif and the structure of the target protein, then continues with a structural alignment, returning the RMSD (Align).

# 3 Results

## 3.1 CNNs

The metric we decided to use to measure the performance of our model is the percentage of actual motif centers or keypoints (as given by our target vectors) within a threshold of amino acid distance from the center predicted by our model. Though it would be nice to use some sort of ground truth data for domain centers and keypoints, we didn't find many datasets of this sort available, and since our CNN method is aiming to perform similarly to HMMER at faster speeds, rather than explicitly trying to improve performance, we decided that comparing to the HMMER-predicted centers and keypoints was appropriate. Additionally, adding a distance threshold allowed us to examine whether the model's predictions were generally close to correct, or completely in the wrong region.

| | Condensation Domain | | | AMP Binding Domain | | |
|---|---|---|---|---|---|---|
| Accuracy Threshold (Within X AAs) | 10 | 50 | 100 | 10 | 50 | 100 |
| C=1, L=1, [31], $\sigma = 30$ | 1.15% | 5.27% | 10.6% | 1.38% | 5.67% | 10.9% |
| C=2, L=0, [5,3], $\sigma = 30$ | 2.75% | 14.9% | 29.2% | 4.38% | 13.4% | 29.4% |
| C=3, [11,11,11], $\sigma = 30$ | 1.62% | 16.4% | 29.9% | 3.02% | 10.6% | 22.3% |
| C=2, L=0, [5,3], $\sigma = 30$, One-Hot | 4.7% | 20.9% | 38.2% | 5.16% | 23.3% | 41.0% |
| C=2, L=1, [5,3], $\sigma = 30$, One-Hot | 0.95% | 4.79% | 9.57% | 1.06% | 4.91% | 9.7% |

Table 1: Center Prediction Results: Results using centers as keypoints for domains. Performance was generally poor, the best setting was found to be two convolutional layers, the first with a kernel size of 5, and the second with a kernel size of 3, and a gaussian sigma of 30, using a one hot encoding. In the lefthand column of the table we have the model parameters, with C being the number of convolutional layers, L being the number of fully connected linear layers, and sigma being the parameter used for the gaussian. The first three sets of results in the table were derived using the integer (1-20) encodings of the amino acid, while the last two sets of results were derived using the one-hot encoding.

| | Condensation Domain | | | AMP Binding Domain | | |
|---|---|---|---|---|---|---|
| Accuracy Threshold (Within X AAs) | 10 | 50 | 100 | 10 | 50 | 100 |
| C=2, L=1, [3,3], $\sigma = 30$ | 1.06% | 4.92% | 9.75% | 1.07% | 4.86% | 9.62% |
| C=2, L=0, [5,3], $\sigma = 5$, | 1.72% | 11.14% | 23.54% | 1.3% | 9.83% | 19.1% |
| C=1, L=0, [7], $\sigma = 5$, | .75% | 12.3% | 22.17% | 0.9% | 9.28% | 19.34% |

Table 2: Keypoint Prediction Results: Results using conserved locations as keypoints for domains. Performance was still generally poor, though the best model was once again the 5-width and 3-width kernel. All models shown here are trained using the one-hot encodings.

Overall, in spite of trying a variety of choices of hyperparameters, the results were pretty poor, and not much better than we would expect from random chance, both when predicting the center-points of the domains, and when predicting using keypoints. For both methods, the kernel size of five followed by a kernel size of three performed relatively well, but overall, performance was disappointing.

## 3.2 Structural Alignment

Each structure used was manually downloaded from the UniProt database, and as a result, only 20 structures were inspected (10 from each protein family). In general, the structural alignment did

very well in separating proteins that contained the real motif (PF0068 chosen) from ones that do not contain the motif (from PF00501).

Table 3: Structural Alignment Results

| contain motif | | no motif | |
|---|---|---|---|
| RMSD | Match | RMSD | Match |
| 1.519 | 36.5 | 14.606 | 32 |
| 1.156 | 64 | 15.254 | 25.5 |
| 1.504 | 37.5 | 6.475 | 28.5 |
| 1.132 | 64 | 12.039 | 31.5 |
| 1.156 | 59 | 9.074 | 23 |
| 1.570 | 39.5 | 10.744 | 26 |
| 1.074 | 59 | 9.231 | 35.5 |
| 1.528 | 38.5 | 6.409 | 38.5 |
| 1.160 | 61 | 4.672 | 40 |
| 1.463 | 36.5 | 6.126 | 27 |

Table 4: The RMSD and sequence match score: As shown above, difference in RMSD separated proteins that contain motif from ones that do not. Which, the sequence matching 30-40 percent fails to distinguish between the two groups.

As shown in Table 4, when the amino acid sequence is aligned to the reference substructure amino acid sequence, a 30 to 40 percent matching score cannot decide if the protein contains the motif PF 00668. However, the true motifs when aligned with the reference structure return much lower RMSD (less than 2) when compared to false ones (only matching the sequences, RMSD > 4).

One AlphaFold prediction was done on >A0A073JYF7 (containtrue motif) which took 1.5 hours for 377 amino acids. The resulting predicted structure aligned to the reference structure with RMSD = 0.998

# 4 Conclusions

This paper explored various convolution-based architectures for predicting the locations of the Condensation Domain (PF00668) and AMP Binding Domain (PF00501) in protein sequences. Different combinations of convolutional and linear layers, kernel sizes, encoding methods (one-hot and integer), and regularization techniques (dropout and batch normalization) were experimented with. However, models generally struggled to achieve high accuracy, with the best architecture performing only around 5% accuracy within a range of 50 amino acids of the correct domain location for the condensation domain and around 23% for the AMP binding domain.

A potential explanation for the poor performance lies in the nature of the domain sequences themselves. By examining multiple sequence alignments of these domains, while the domains have conserved motifs, we recognized that there are some residues that are highly conserved, but even fairly conserved residues can have multiple common amino acids at that spot, which complicates domain identification. Another potential explanation is that there is some flaw in our model design, in the way we adapted methods from object recognition spaces, the way we generated our training data, the way we designed our model, or the way we set our hyperparameters.

The structural alignment performed well with the limited samples tested. However, to conclude its accuracy or efficiency, many more samples are needed. This points to a major problem with this approach. The structure alignment requires structural data (PDB) to operate, which turns out to be very hard to find. Most of the sequences do not have a resolved or predicted structure (not even a structure with enough sequence similarity) readily available, causing the need for another prediction of structure (AlphaFold 2). This prediction can take very long (15 min for approximately 300 amino acids, or hours / days with the 10k+ amino acids for many of the inputs used). This is contradictory with our goal of finding a more efficient algorithm, though structural alignment may still be a useful technique for specific cases where our predictions using other methods have low confidence.

Moving forward, several key strategies could potentially improve performance for the CNN model. First, training data could incorporate locations of multiple conserved locations in the domain sequence as key points. By leveraging these key points as anchors, the models may be better equipped to navigate the sequence variability and localize the domains more accurately, though it would require development to interpolate between them. Additionally, should you want to expand this model out from a proof of concept to a more useful tool, negative examples would need to be introduced into training data to prevent the model from predicting that there is a relevant domain within every inputted protein sequence. Additionally, for this model to be useful, it would likely need to be trained to identify a wide variety of motifs, rather than a single motif, as training an individual model for every possible motif does not seem very feasible. Another way to potentially improve performance would be to use more informative embeddings to represent amino acids. The one-hot and integer based encodings we used for amino acids are not very informative, and so, using more informative embeddings, which take into account the chemical features of the amino acids might allow the model to learn more readily.

Overall, developing new machine learning methods, or even just applying existing methods to a new domain, is more of an art than a science, and it appears that our CNN method would require some more examination and improvement to become a reasonable option for protein domain detection. However, we still believe that applying object detection methods to the problem of protein domain detection is an interesting prospect, and could possibly yield gains over more traditional methods in accuracy or speed, especially with the large amounts of training data that are becoming available.

# References


"Align." PyMOLWiki, 6 Dec. 2017, pymolwiki.org/index.php/Align.

Ding, Feng, and Nikolay V Dokholyan. "Emergence of Protein Fold Families through Rational Design." PLOS Computational Biology, Public Library of Science, 7 July 2006, journals.plos.org/ploscompbiol/article?id=10.1371

Finn, Robert D., et al. "HMMER Web Server: Interactive Sequence Similarity Searching." Nucleic Acids Research, vol. 39, no. Web Server issue, July 2011, pp. W29–37. PubMed Central, https://doi.org/10.1093/nar/gkr367.

Hunter, Sarah, et al. "InterPro: The Integrative Protein Signature Database." Nucleic Acids Research, vol. 37, no. Database issue, Jan. 2009, pp. D211-215. PubMed, https://doi.org/10.1093/nar/gkn785.

Illergård, Kristoffer, et al. "Structure Is Three to Ten Times More Conserved than Sequence—A Study of Structural Response in Protein Cores." WILEY Online Library, proteins, 15 Nov. 2009, onlinelibrary.wiley.com/.

Larralde, Martin, and Georg Zeller. "PyHMMER: A Python Library Binding to HMMER for Efficient Sequence Analysis." Bioinformatics, edited by Can Alkan, vol. 39, no. 5, May 2023, p. btad214. DOI.org (Crossref), https://doi.org/10.1093/bioinformatics/btad214.

Mi, Huaiyu, et al. "The PANTHER Database of Protein Families, Subfamilies, Functions and Pathways." Nucleic Acids Research, vol. 33, no. Database Issue, Jan. 2005, pp. D284–88. PubMed Central, https://doi.org/10.1093/nar/gki078.

Paszke, Adam, et al. Automatic Differentiation in PyTorch. Oct. 2017. openreview.net, https://openreview.net/forum?id=BJJsrmfCZ.

Punta, Marco, et al. "The Pfam Protein Families Database." Nucleic Acids Research, vol. 40, no. Database issue, Jan. 2012, pp. D290-301. PubMed, https://doi.org/10.1093/nar/gkr1065.

Rives, Alexander, et al. "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences." Proceedings of the National Academy of Sciences, vol. 118, no. 15, Apr. 2021, p. e2016239118. DOI.org (Crossref), https://doi.org/10.1073/pnas.2016239118.

Singh, Rohit, and Mitul Saha. "Identifying Structural Motifs in Proteins." Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, U.S. National Library of Medicine, 2003, pubmed.ncbi.nlm.nih.gov/12603031/.

UniProt. https://www.uniprot.org/help/uniprotkb. Accessed 10 Apr. 2024. Wang, Yan, et al. "Protein Domain Identification Methods and Online Resources." Computational and Structural Biotechnology Journal, vol. 19, Feb. 2021, pp. 1145–53. PubMed Central, https://doi.org/10.1016/j.csbj.2021.01.041.

Zhou, Xingyi, et al. Objects as Points. arXiv:1904.07850, arXiv, 25 Apr. 2019. arXiv.org, https://doi.org/10.48550/arXiv.1904.07850.