Structural and Computational Characterization of Kunitz-Type Domains*

Project Report for Laboratory of Bioinformatics I Course

Marina Mariano, May 2025

Abstract

We developed a statistical model to detect Kunitz-type domains, small cysteine-rich protein regions with protease inhibitory activity, using a profile Hidden Markov Model (HMM). The model encodes position-specific residue conservation and variability, accounting for amino acid substitutions, insertions, and deletions.

To build the model, we curated a non-redundant set of crystal structures from the Protein Data Bank (PDB), selecting high-resolution entries with a single Kunitz domain. Structural alignments were performed using PDBeFold to maximize accuracy across conserved regions and capture shared 3D motifs.

The resulting profile HMM was validated against curated SwissProt datasets representing true positives and true negatives. Performance was assessed through standard classification metrics, including precision, recall, F1-score, and Matthews correlation coefficient (MCC), confirming the effectiveness of our structure-based approach in identifying Kunitz domains.

Contents

*Supplementary materials available at: https://github.com/MarinaMariano/kunitz-hmm-model

1 Introduction

Kunitz domains are small, evolutionarily conserved protein domains that function primarily as protease inhibitors, especially against serine proteases like trypsin, chymotrypsin, and elastase. It is a protein domain of about 50-60 amino acids (see fig 1). First discovered in the soybean trypsin inhibitor (STI). It is characterized by a compact stable fold, stabilized by three disulfide bridges (six cysteines), typically forms a beta sheet and an alfa helix (see fig 3). The serine protease inhibition is performed blocking the active site of target enzyme to prevent unwanted proteolysis. Some Kunitz domains modulate ion channels (e.g., voltage-gated potassium channels). They are Involved in anti-coagulant, anti-inflammatory, and anti-tumor pathways (see fig 4). Examples of proteins with Kunitz domains: BPTI (Bovine Pancreatic Trypsin Inhibitor), Tissue Factor Pathway Inhibitor (TFPI), Amyloid precursor protein (APP), Protease inhibitors in venom. Biological Relevance: it protects tissues from excessive protease activity during inflammation or injury, in neurobiology, Kunitz-containing proteins may regulate synaptic activity, in cancer, their dysregulation can affect invasion and metastasis due to changes in proteolysis. Owing to the structural and functional features outlined above, Kunitz domains represent an ideal target for computational modeling using probabilistic approaches such as Profile Hidden Markov Models (HMMs), enhancing the model's ability to discriminate true domain instances from unrelated sequences. Also, automated identification of Kunitz domains across large proteomic datasets has practical applications in functional annotation, domain prediction in newly sequenced proteins, and discovery of novel protease inhibitors, with potential relevance in biomedical fields such as inflammation, neurobiology, and cancer research.

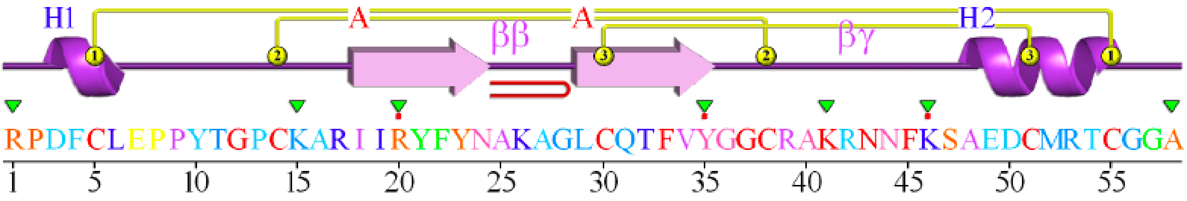


Figure 1: Residue conservation in chain A of the Bovine Pancreatic Trypsin Inhibitor (PDB ID: 1BPI). The protein sequence is colored according to evolutionary conservation levels computed by ConSurf-DB, ranging from 1 (low conservation, blue) to 9 (high conservation, red). Secondary structure elements: helices (H1, H2) shown as purple cylinders and b-strands (A, B) as light-purple arrows. Structural motifs (beta-turnsand gamma-turns), along with disulfide bridges (yellow "S–S" symbols) contribute to the stability of the Kunitz domain. Green arrows labeled "AC1" indicate residues involved in functional sites. The image was obtained from PDBsum (UniProt: P00974, Pfam: PF00014).

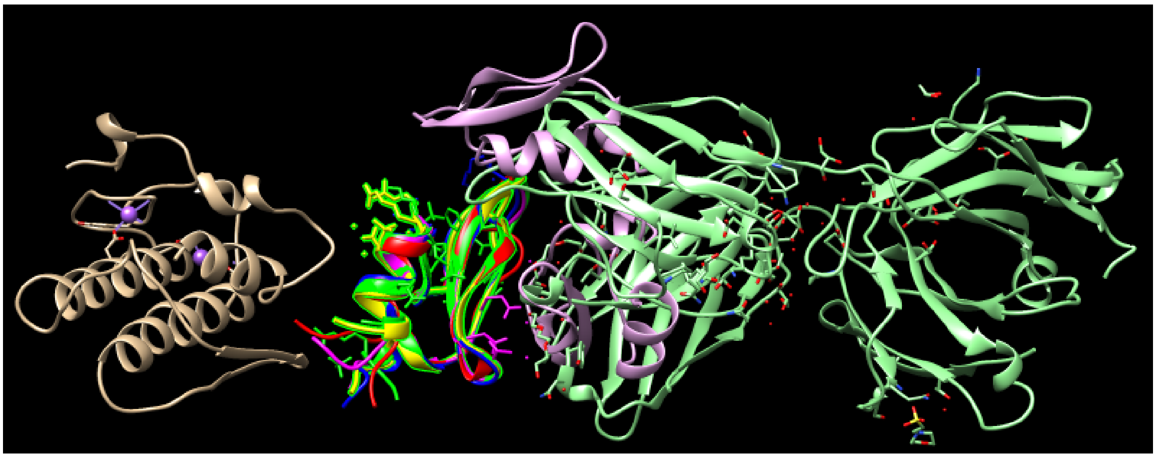


Figure 2: Structural superimposition of selected Kunitz domain-containing proteins used for HMM model generation: the five protein structures 1BUN:B (bovine pancreatic trypsin inhibitor), 1DTX:A, 3BYB:A, 4DTG:K, and 6Q61:A were aligned using Chimera (Tools → Structure Comparison → MatchMaker), with 1BUN:B set as the reference structure. Superimposing these structures highlights the high degree of conservation in the core domain architecture, despite minor sequence and conformational differences.

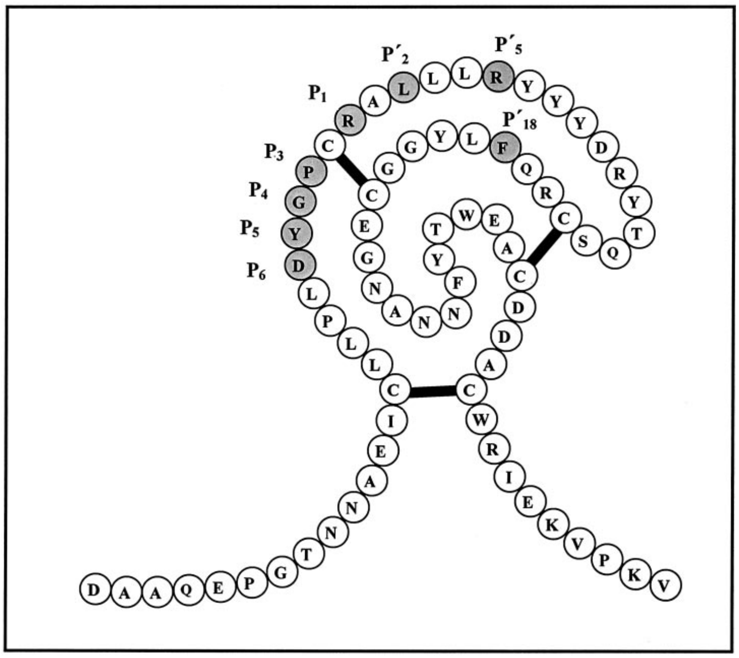


Figure 3: Schematic representation of the conserved fold of the Kunitz-type protease inhibitor, characterized by the three disulfide bonds formed by six cysteine residues, crucial for stabilizing the domain's conformation and facilitating its interaction with proteases.

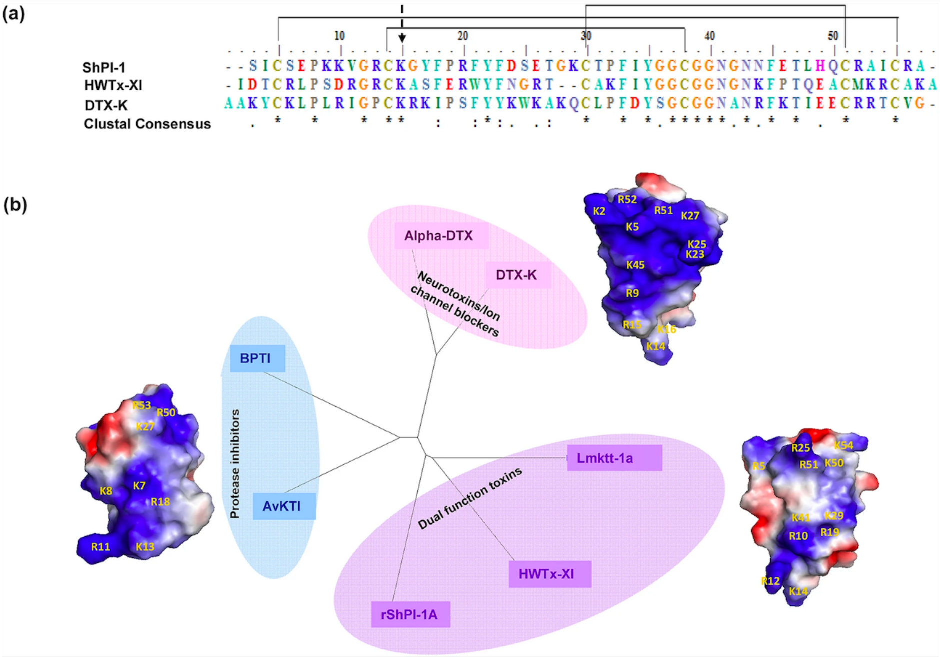

(a)
10
20
30
40
50
ShPI-1 --SICSEPKKVGRCKGYFPRFYFDSETGKCTPFIYGGCGGNGNNFETLHQCRAICRA-
HWTx-XI -IDTCRLPSDRGRCKASFERWYFNGRT--CAKFIYGGCGGNGNKFPTQEACMKRCAKA
DTX-K AAKYCKLPLRIGPCKRKIPSFYYKWKAKQCLPFDYSGCGGNANRFKTIEECRRTCVG-
Clustal Consensus
(b)
Alpha-DTX
DTX-K
Neurotoxins/Ion channel blockers
BPTI
AvKTI
Protease inhibitors
Lmktt-1a
HWTx-XI
rShPI-1A
Dual function toxins
K2
R52
R51
K27
K5
K25
K23
K45
R9
R15
K16
K14
R33
R50
K27
K7
K8
R18
R11
K13
R25
K54
R5
R51
K50
K41
R39
R10
R19
R12
K14

Figure 4: Structural and sequence comparison of Kunitz-type peptides. (a) Multiple sequence alignment of representative Kunitz peptides showing conserved residues. (b) Electrostatic surface representations and functional clustering of protease inhibitors, ion channel blockers, and dual-function toxins.

2 Materials and Methods

The problem of identifying proteins containing the Kunitz domain can be addressed using either a profile Hidden Markov Model (HMM) approach with HMMER or a sequence similarity search with BLAST. While both methods aim to retrieve homologous sequences, they differ significantly in their underlying principles and performances. HMMER uses a statistical model built from a multiple sequence alignment, making it more sensitive and capable of detecting remote homologs with conserved patterns that may not be evident in pairwise alignments. In contrast, BLAST relies on direct sequence similarity and is generally faster and simpler to implement, but it may fail to detect more divergent domain instances. For these reasons, the profile HMM-based approach was selected as the most suitable method for identifying Kunitz domain-containing proteins.

2.1 Data Retrieval and Filtering

To build a structurally informed profile Hidden Markov Model (HMM) for the Kunitz domain, we started by retrieving structural data from the RCSB Protein Data Bank (PDB). This can be done using the advanced search function, where it is possible to specify relevant filters such as the Pfam ID `PF00014` (which identifies the Kunitz-type protease inhibitor domain), a maximum resolution threshold (e.g., ≤ 3.5 Å) to ensure structural quality, and a minimum sequence length (e.g., >45 and <80 amino acids) to exclude incomplete or truncated entries.

After running the search, we generated a GraphQL query directly from the "Custom Report" interface. We then used this query in a Python script to submit the request to the RCSB GraphQL API endpoint. The script successfully retrieved a JSON file containing detailed structural and sequence information for each matching entity (query_kunitz_pdb.py, kunitz_structures_graphql.json). By leveraging structured JSON data and automating the parsing process with Python, this approach ensures reproducibility and precision in domain filtering.

Particularly, we manually compiled a list of 159 PDB IDs obtained through the advanced search filters and integrated them into a Python script (query_kunitz_pdb.py). This script was designed to divide the list into batches of 100 IDs (due to API constraints) and perform consecutive queries to the RCSB GraphQL API, retrieving detailed information for each structure. The output was saved in a consolidated JSON file named all_kunitz_structures.json, which contained structural metadata, Pfam domain annotations, chain identifiers, sequence lengths, and canonical amino acid sequences for each polymer entity.

To prepare this dataset for downstream use, we developed a second script filter_kunitz_to_fasta .py to parse the JSON file and extract only the sequences that met the following conditions: (i) contained the Pfam domain PF00014, (ii) had an experimental resolution of ≤ 3.5 Å, (iii) exhibited a sequence length between 45 and 80 amino acids, and (iv) were not redundant. The script identified and removed duplicate sequences, retaining only unique entries. It then generated a FASTA-formatted file named filtered_kunitz_sequences.fasta, which can finally be clustered using CD-HIT.

After running the filtering script on the full dataset of 159 PDB entries, a total of 85 unique sequences were retained in the resulting FASTA file (filtered_kunitz_sequences.fasta). This reduction is expected and biologically justified. Many of the PDB structures share identical Kunitz domain sequences, either because they represent the same protein solved under different experimental conditions or due to the presence of multiple identical chains within the same structure. The script was designed to include each sequence only once, regardless of how many PDB entries or chains it appears in. The resulting FASTA therefore contains a non-redundant, high-confidence set of Kunitz domain sequences suitable for clustering and model building.

2.2 Sequence Clustering and Structural Alignment

After generating a filtered FASTA file (filtered_kunitz_sequences.fasta) containing 85 non-identical sequences, CD-HIT can be used to group highly similar sequences and retain one representative per cluster.

```
cd-hit -i filtered_kunitz_sequences.fasta -o pdb_kunitz_nr.fasta -c 0.7 -n 5
```

Here, -c sets the sequence identity threshold (70% in this case) and -n specifies the word size required by CD-HIT for that threshold. The output includes a representative FASTA file (pdb_kunitz_nr.fasta) and a .clstr file describing cluster membership. To interpret the clustering results, the .clstr file can be converted to a tabular format using the script clstr2txt.pl, which makes it easier to identify representative sequences and assess the size and composition of each cluster.

CD-HIT successfully grouped the 85 sequences into 19 clusters, each representing a group of highly similar sequences. As output, it generated two files: pdb_kunitz_nr.fasta, which contains the 19 representative sequences (one per cluster), and pdb_kunitz_nr.fasta.clstr, which encodes the cluster membership for each input sequence.

Since these representative sequences all correspond to proteins with solved 3D structures in the PDB, they can be directly used in PDBeFold for structural alignment. The transition from CD-HIT to PDBeFold involves extracting the PDB ID and chain from the

FASTA headers to generate an input file (pdbefold_input.txt) in the required format (e.g., 1AAL:A).

Thus, CD-HIT serves as a critical filtering step that condenses the dataset into a non-redundant set of structurally annotated entries, laying the foundation for an accurate and interpretable multiple structure alignment with PDBeFold.

After performing the multiple structural alignment, we observed that three structures—1bun:B, 4ntw:B, and 4u30:X—had an RMSD greater than 1.0, and one structure, 4bqd:A, had a Q-score lower than 0.7. The RMSD (Root Mean Square Deviation) measures the average distance between aligned atoms after superposition: values above 1.0 generally indicate poor structural overlap. The Q-score reflects the quality of structural alignment, considering both residue equivalence and proximity, with higher values indicating better alignment consistency. A low Q-score suggests less reliable structural similarity.

Based on these criteria, we decided to exclude these four sequences from downstream analysis. We then manually selected the remaining 15 structures and used the "Download FASTA alignment" option, which exports the amino acid sequence alignment derived from the structural alignment—not 3D coordinates. This is advantageous because it provides a more accurate alignment than one based solely on sequence.

$$\text{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\vec{x}_i - \vec{y}_i\|^2} \tag{1}$$

Where N is the number of aligned atoms, and $\vec{x}_i$ and $\vec{y}_i$ represent the 3D coordinates of atom i in the two structures.

$$Q = \frac{1}{L}\sum_{i=1}^{L}\exp\left(-\frac{d_i^2}{\sigma^2}\right) \tag{2}$$

Where L is the number of aligned residues, d_i is the spatial distance between aligned residues i, and σ is a scaling factor typically set to 3 Å.

Table 1: Structural alignment summary of representative Kunitz sequences used in model construction (not filtered yet).

Structure	N_{res}	N_{SSE}	RMSD	Q-score
1bun:B	61	3	1.0897	0.7386
1dtx:A	59	4	0.4730	0.8434
1fsr:I	57	4	0.7152	0.8466
1knt:A	55	4	0.6894	0.8808
1yco:I	66	4	0.4770	0.7537
1zr0:B	63	4	0.4394	0.7925
3byb:A	58	4	0.4699	0.8583
3m7q:B	61	4	0.6567	0.7978
3wny:A	56	4	0.7027	0.8634
4bqd:A	78	6	1.1643	0.6415
4dtg:K	60	4	0.4319	0.8327
4ntw:B	60	4	1.2502	0.7242
4u30:X	54	3	1.2376	0.8071
4u32:X	54	3	0.7432	0.8247
5mwz:A	55	4	0.5192	0.8687
5vy7:A	60	4	0.4852	0.8283
6bx8:B	55	4	0.4233	0.9092
6har:E	54	4	0.4867	0.9222
6q61:A	59	4	0.5506	0.8362

Note. N_{res} = number of residues in the aligned domain; N_{SSE} = number of aligned secondary structure elements. The average number of aligned residues was 51, the overall RMSD was 0.9912, and the overall Q-score was 0.5567.

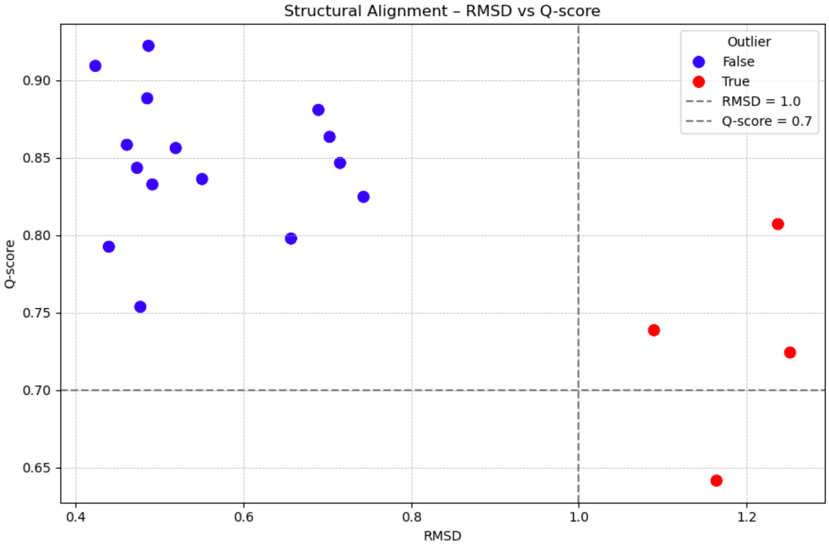


Figure 5: Scatter plot of RMSD versus Q-score for structures used in the multiple structural alignment. Outlier structures (in red) exceed the threshold of RMSD $>$ 1.0 or Q-score $<$ 0.7 and were excluded from the final alignment used for HMM construction.

2.3 HMM Model Construction

Although removing sequences might seem risky in terms of quantity, the final dataset of 15 is more than sufficient for constructing a robust HMM. General guidelines for reliable HMM construction, such as those followed by Pfam, recommend having at least 5–10 non-redundant sequences, a high-quality alignment (with minimal gaps and strong conservation), and representation of natural domain variability. The sequences retained in this dataset are well-aligned, structurally consistent, represent diverse organisms, and provide excellent coverage of the Kunitz domain, making them a solid foundation for building a structurally informed HMM with strong predictive power.

Since not all the sequences in `kunitz_15_seq_structural_MSA.fasta` are of the same length (including gap counts), one outlier needed to be removed. In particular, `1dtx:A C` is 132 residues long instead of the expected 83, and would otherwise prevent `hmmbuild` from generating a valid model.

To remove it, the following command was used:

```
awk 'BEGIN{RS=">"; ORS=""} NR==1{next} $0 !~ /^PDB:1dtx:A/ {
   print ">" $0}' kunitz_15_seq_structural_MSA.fasta >
   kunitz_clean_14_MSA.fasta
```

To verify the result:

```
grep -c "^>" kunitz_clean_14_MSA.fasta
```

This command returned 14, confirming the removal.

The model was then constructed with:

```
hmmbuild kunitz_structural.hmm kunitz_clean_14_MSA.fasta
```

The resulting model spans 58 match states, which is consistent with the typical size of the Kunitz domain (∼60 amino acids). It was built from 14 structurally aligned sequences, emphasizing quality over quantity. The resulting profile HMM is well-conserved and reflects key structural and functional constraints. The average relative entropy per position (information content) is approximately 0.96 bits, indicating a strong and informative alignment.

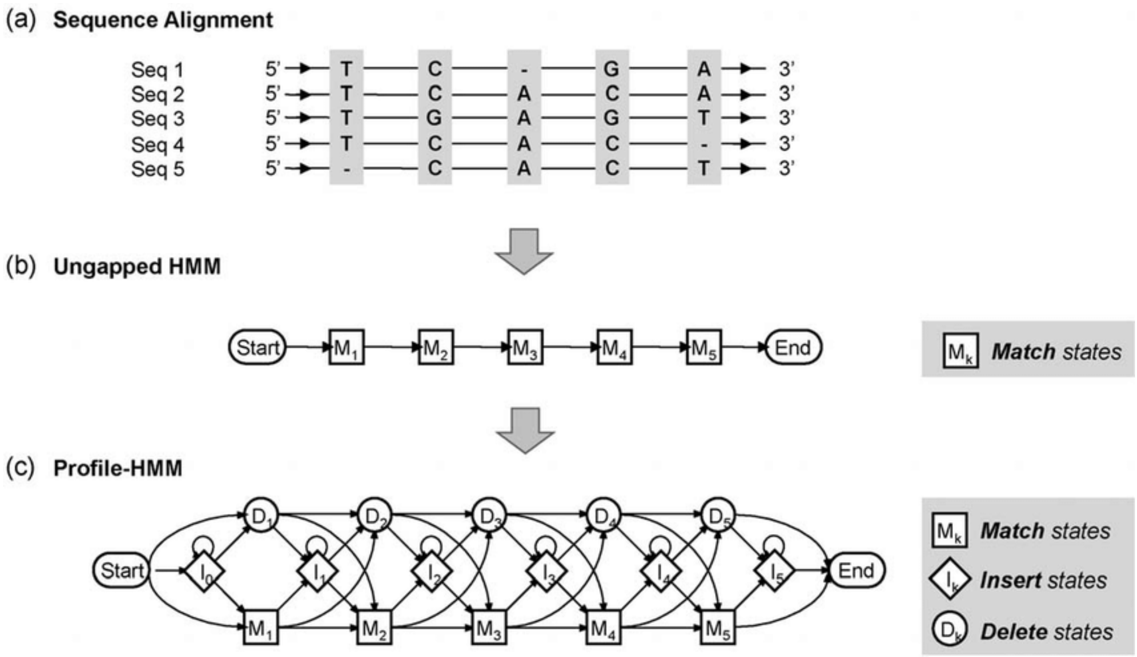


Figure 6: Illustration of the construction of a Profile Hidden Markov Model (HMM). (a) Sequence alignment showing aligned sequences with gaps. (b) Ungapped HMM representing match states. (c) Profile-HMM incorporating match states (M), insertion states (I), and deletion states (D), capturing evolutionary changes and variations in sequence alignment.

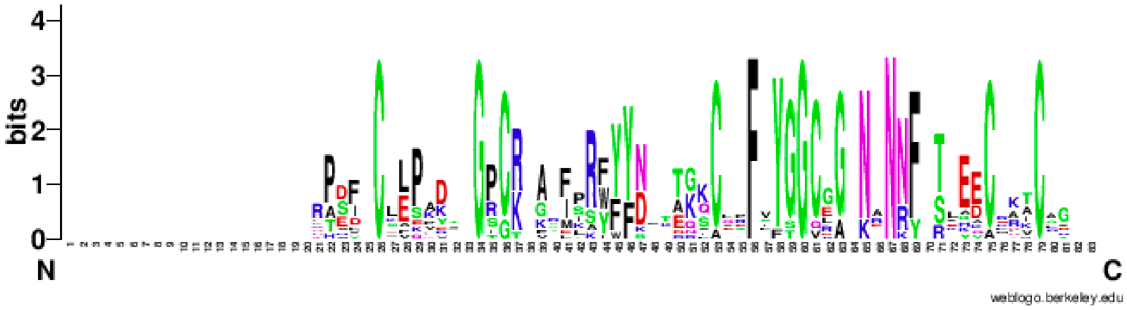


Figure 7: Sequence logo generated from the multiple sequence alignment, showing residue conservation based on observed frequencies at each position. It was produced using WebLogo.

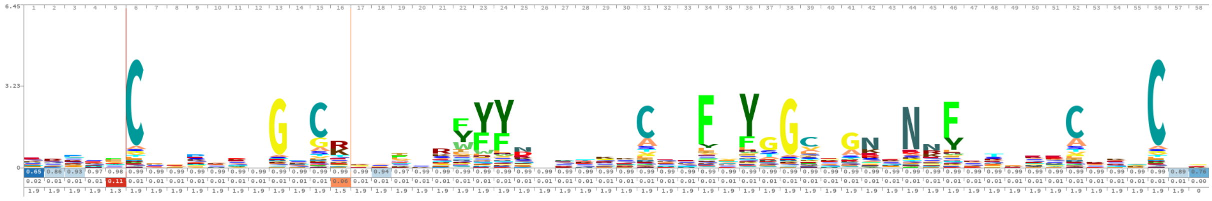


Figure 8: HMM-based logo derived from the profile model, integrating emission probabilities to represent conserved and variable positions more accurately. It was produced using Skylign.

2.4 Benchmarking the HMM Model

To evaluate the performance of our HMM, we applied it to a curated subset of the SwissProt database. To avoid artificially inflated metrics, sequences identical or highly similar to those used for training were excluded. This filtering was accomplished by aligning the 14 training sequences (from the PDB) against a set of 397 known Kunitz-domain-containing proteins from SwissProt (18 human, 379 non-human) using `blastp`. Sequences showing $\geq$95% identity and alignment lengths $\geq$50 residues were excluded, resulting in a final set of 1856 non-redundant positive examples. These sequences were extracted using a custom script and stored in a FASTA file (`filtered_kunitz_sprot.fasta`).

Negative examples were derived from the full SwissProt proteome (`uniprot_sprot-2.fasta`). All UniProt IDs were extracted and compared against those in the filtered positive set, and all non-overlapping IDs were considered negatives. After random shuffling, both the positive and negative sets were split into two halves: `pos_1.ids`/`pos_2.ids` and `neg_1.ids`/`neg_2.ids`.

FASTA files were generated using the script `extract_fasta_by_ids.py`, yielding balanced subsets (`pos_1.fasta`, `neg_1.fasta`, etc.). These were used to evaluate the HMM with `hmmsearch`, setting the virtual database size to 1000 via the `-Z` option for comparability. The `--max` flag was also used to disable heuristics and ensure sensitive detection. Four search results were produced: `pos_1.out`, `pos_2.out`, `neg_1.out`, and `neg_2.out`.

To construct .class files, each output was parsed to extract UniProt IDs, classification labels (1 for positives, 0 for negatives), global E-values (column 5), and best domain E-values (column 8). Missing sequences in the negative .class files were reintroduced with default values (`E=10.0`), ensuring all IDs were represented. Final balanced sets were named `set_1.class` and `set_2.class`.

2.5 Cross-Validation and Threshold Optimization

To assess generalization, we performed a 2-fold cross-validation: each set (`set_1.class`, `set_2.class`) was used once as a validation set while the other was used to identify the optimal threshold. The script `score.py` computes performance metrics at various E-value cutoffs, including accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC).

A sweep from $1e^{-1}$ to $1e^{-30}$ was performed to find the threshold maximizing MCC. For `set_1`, an optimal threshold of 10^{-6} yielded perfect classification (TPR = 0.994536, FPR = 0.000007, MCC = 0.991824). When applied to `set_2`, the same threshold yielded nearly perfect performance (TPR = 0.983607, FPR = 0.000007, MCC = 0.986296), confirming its robustness. Thus, 10^{-6} was chosen as the optimal threshold balancing sensitivity and specificity.

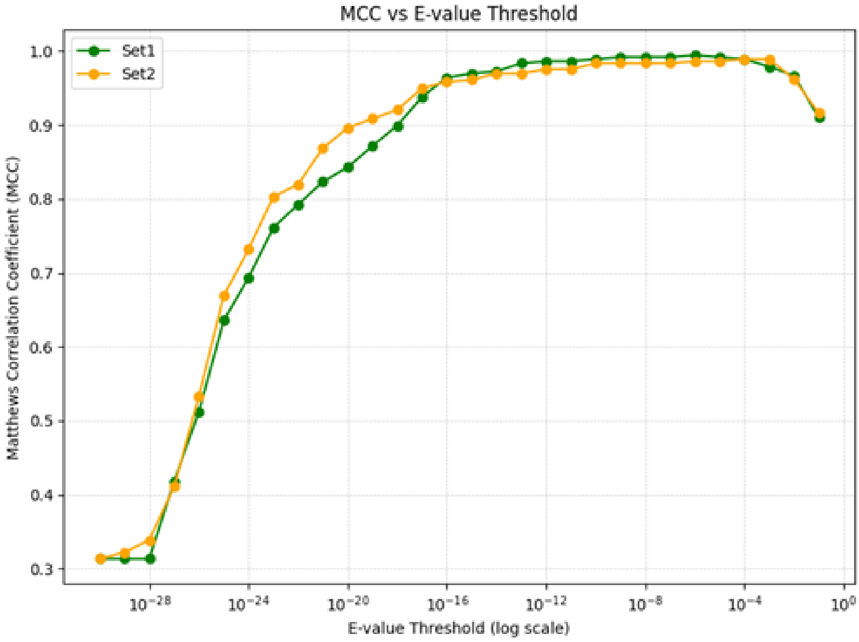


Figure 9: Comparison of Matthews Correlation Coefficient (MCC) values across multiple classification thresholds for two independent validation sets. The MCC was calculated from the confusion matrix at each threshold using the results provided by HMMER. A vertical red dashed line marks the threshold of 10^{-6}, which was selected as the optimal cutoff based on its consistently high MCC performance in both sets. The curve shows how MCC evolves across a wide range of thresholds, highlighting a plateau region of near-optimal performance between 10^{-14} and 10^{-6}. The performance peaks around 10^{-6} for both validation sets, supporting the selection of this value as the optimal threshold.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

2.5.1 ROC Analysis

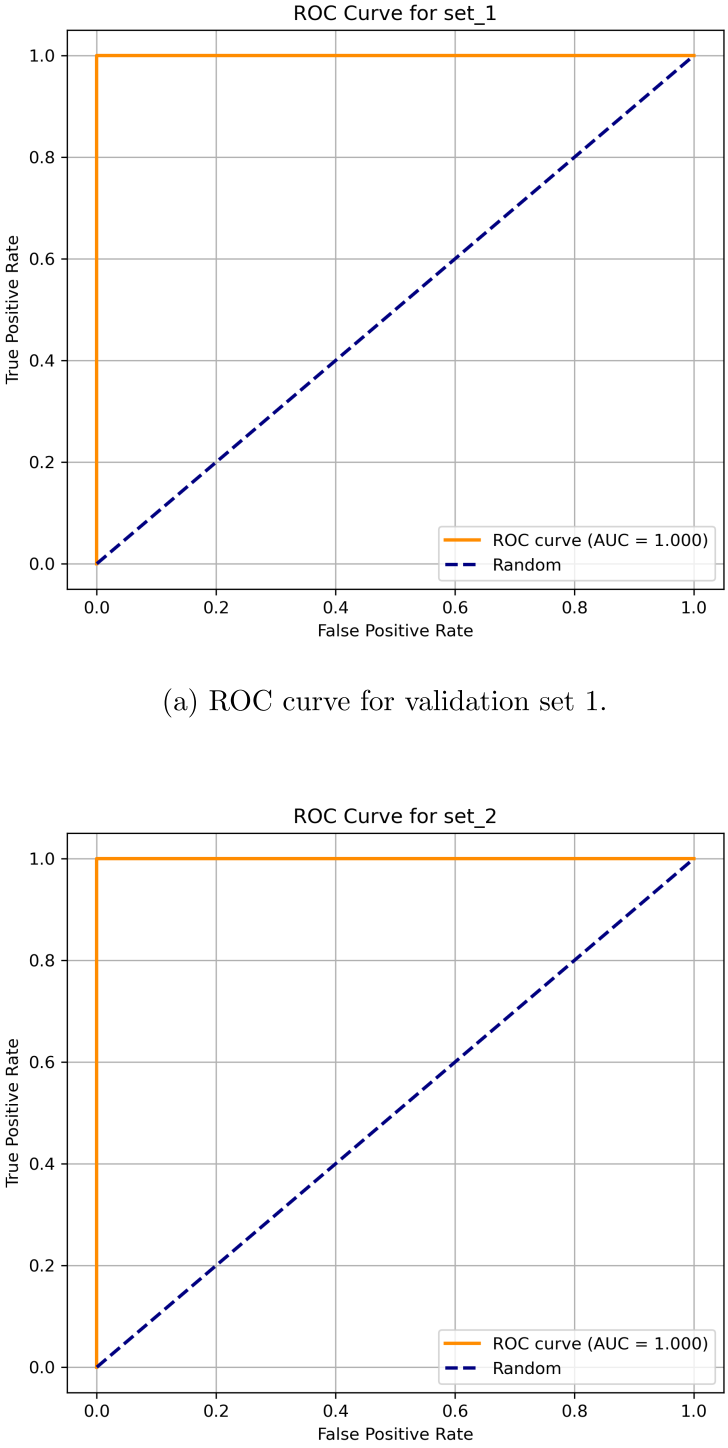
To complement threshold-based evaluation, we employed a Receiver Operating Characteristic (ROC) analysis, a standard approach for visualizing and assessing the performance of binary classifiers across varying decision thresholds. ROC curves plot the true positive rate (TPR) against the false positive rate (FPR) at different E-value cutoffs, providing an aggregate view of a model's ability to discriminate between positive and negative cases independently of any fixed threshold.

A custom Python script was developed to parse the `.class` files and convert the E-values returned by `hmmsearch` into confidence scores using negative logarithms (i.e.,

$-\log_{10}$(E-value)). This transformation ensures that lower E-values (indicating stronger matches) correspond to higher confidence scores. At each threshold, the script computes TPR and FPR, then plots the corresponding ROC curve.

The resulting curves demonstrated near-perfect separation. The area under the curve (AUC) was calculated for both validation sets: set 1 and set 2. In both cases, the AUC reached a value of 1.000, indicating flawless discrimination between sequences containing and not containing the Kunitz domain. Such a result suggests that the model assigns consistently higher scores to true positives than to false positives across the entire range of thresholds.

These findings are further supported by the confusion matrices and performance scores computed at the optimal E-value threshold of 10^{-6}. For Set 1, the model correctly classified 181 out of 182 positive sequences (TP) and 286,415 out of 286,416 negative sequences (TN), resulting in only one false positive (FP) and one false negative (FN). For Set 2, the performance remains comparably high, with 182 true positives and 286,415 true negatives. However, this time the model produced three false positives and two false negatives.



(a) ROC curve for validation set 1.

(b) ROC curve for validation set 2.

Figure 10: Comparison of ROC curves for the two validation sets.

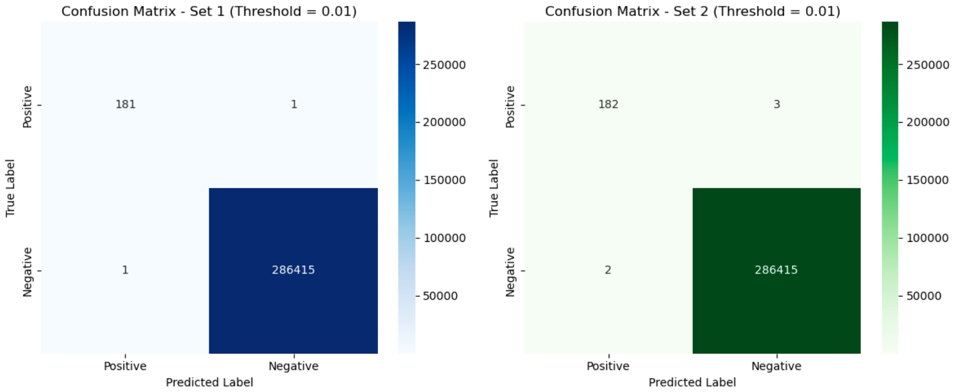


Figure 11: Confusion matrices for validation sets at threshold 10^{-6}. Color intensities reflect absolute sequence counts.

3 Results and Discussion

The performance of the structure-informed HMM was evaluated on two independent validation sets using standard classification metrics: accuracy, sensitivity, specificity, and Matthews Correlation Coefficient (MCC). To further refine classification performance, a more stringent E-value threshold of 1×10^{-6} was applied to both validation sets. Under this setting, the model achieved exceptional results. For **Set 1**, the accuracy reached **99.9990%**, with a **Matthews Correlation Coefficient (MCC)** of **0.9918**, indicating an excellent balance between sensitivity and specificity. The model correctly predicted 182 true positives and 286,416 true negatives, with only **1 false positive** and **1 false negative**, yielding a sensitivity of **0.9945** and an extremely low false positive rate of **0.000007**.

In **Set 2**, performance remained similarly high, with an accuracy of **99.9983%** and an MCC of **0.9863**. Although the number of false negatives increased slightly to 3, and false positives to 2, the sensitivity remained strong at **0.9836**, confirming the model's ability to generalize across independent datasets. These results demonstrate that a threshold of 1×10^{-6} provides an optimal trade-off between precision and recall, ensuring robust and reliable domain detection across both test sets. These results highlight the model's ability to generalize across diverse sequence inputs, including variants not present in the training data. Notably, its performance remained stable across a range of classification thresholds. The structural alignment allows the model to capture conserved three-dimensional motifs that may not be evident from sequence data alone, particularly in highly divergent homologs. Incorporating these structurally conserved residues into the multiple sequence alignment ensures that the resulting HMM encodes biologically meaningful positions. Probabilistic modeling with HMMER then leverages this information to detect sequences with subtle similarities that standard sequence-based methods might miss, thus improving sensitivity without compromising specificity.

4 Conclusion

In summary, we developed and validated a profile Hidden Markov Model informed by structural alignments to detect Kunitz-type protease inhibitor domains with near-perfect classification accuracy. The integration of structural features into the alignment process enhanced the model's sensitivity to conserved motifs, while maintaining low rates of false classification. The high performance across independent datasets underscores the model's potential as a reliable tool for large-scale domain annotation tasks. Future applications may include its integration into annotation pipelines for poorly characterized proteomes or in comparative genomic studies aimed at identifying novel Kunitz-like inhibitors in non-model organisms.

References

1. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research, 25*(17), 3389–3402. https://doi.org/10.1093/nar/25.17.3389

2. Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics, 14*(9), 755–763. https://doi.org/10.1093/bioinformatics/14.9.755

3. Eddy, S. R. (2011). Accelerated profile HMM searches. *PLoS Computational Biology, 7*(10), e1002195. https://doi.org/10.1371/journal.pcbi.1002195

4. Laht, S., Koua, D., Kaplinski, L., Lisacek, F., Stöcklin, R., Remm, M. (2012). Identification and classification of conopeptides using profile Hidden Markov Models. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics, 1824*(3), 488–492. https://doi.org/10.1016/j.bbapap.2011.12.004

5. Mishra, M. (2020). Evolutionary aspects of the structural convergence and functional diversification of Kunitz-domain inhibitors. *Journal of Molecular Evolution, 88*(7), 537–548. https://doi.org/10.1007/s00239-020-09959-9

6. de Magalhães, M. T. Q., Mambelli, F. S., Santos, B. P. O., Morais, S. B., Oliveira, S. C. (2018). Serine protease inhibitors containing a Kunitz domain: Their role in modulation of host inflammatory responses and parasite survival. *Microbes and Infection, 20*(9–10), 606–609. https://doi.org/10.1016/j.micinf.2018.01.003

7. Pfam database. (n.d.). Retrieved May 12, 2025, from http://pfam.xfam.org

8. Ranasinghe, S., McManus, D. P. (2013). Structure and function of invertebrate Kunitz serine protease inhibitors. *Developmental Comparative Immunology, 39*(3), 219–227. https://doi.org/10.1016/j.dci.2012.10.005

9. RCSB Protein Data Bank. (n.d.). Retrieved May 12, 2025, from https://www.rcsb.org

10. Roumeliotis, S., Schurgers, J., Tsalikakis, D. G., D'Arrigo, G., Gori, M., Pitino, A., Leonardis, D., Tripepi, G., Liakopoulos, V. (2024). ROC curve analysis: A useful statistic multi-tool in the research of nephrology. *International Urology and Nephrology, 56*(8), 2651–2658. https://doi.org/10.1007/s11255-024-04022-8

11. Skylign. (n.d.). Retrieved May 12, 2025, from https://skylign.org