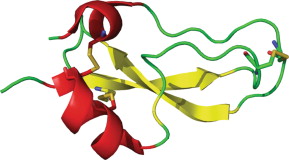
**1.Introduction**

Kunitz-type domain protease inhibitors (KDPIs) are an important type of protease inhibitor and belong to the I2 family of protease inhibitors. Kunitz-type domains are ubiquitously found in natural systems as serine protease.The pancreatic Kunitz inhibitor, also known as bovine basic pancreatic trypsin inhibitor (BPTI), and trypsin-kallikrein inhibitor, is one of the most extensively studied globular proteins. It has proved to be a particularly attractive and powerful tool for studying protein conformation as well as molecular bases of protein/protein interaction(s) and (macro)molecular recognition. BPTI has a relatively broad specificity, inhibiting trypsin- as well as chymotrypsin- and elastase-like serine (pro)enzymes endowed with very different primary specificity. They are about 50–60 amino acids long with a molecular weight of about 6 kDa and fold into a disulfide-rich alpha/beta structure; in particular there are six conserved cysteine residues forming three disulphide bonds that keep the structure compact, 2 alpha helices and 2 beta sheets (Fig. 1)

****

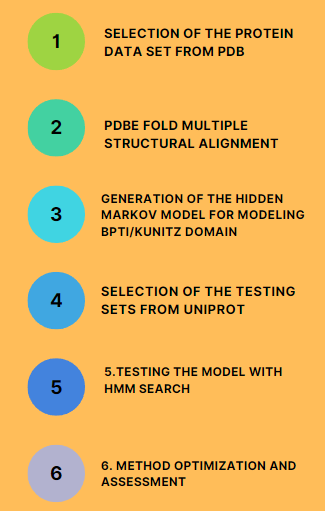
**Figure 1** Structure of BPTI: In this structure (PDB ID: 1BPI) (Parkin et al., 1996)[3], a-helices are shown in red, b-sheets in yellow, and loop regions in green. Three conserved disulfide bonds are shown as sticks.

Kunitz domains have been used as a framework for the development of new pharmaceutical drugs, like aprotinin, that was used as a medication to reduce bleeding during complex surgery. Aprotinin is a monomeric (single-chain) globular polypeptide derived from bovine lung tissue. It has a molecular weight of 6512 and consists of 16 different amino acid types arranged in a chain 58 residues long that folds into a stable, compact tertiary structure of the 'small SS-rich" type, containing 3 disulfides, a twisted β-hairpin and a C-terminal α-helix.

The specific aims of this project are:

1. Build your own model for the Kunitz domain, starting from available structural information.

2. Use the model for annotating Kunitz domains in SwissProt.



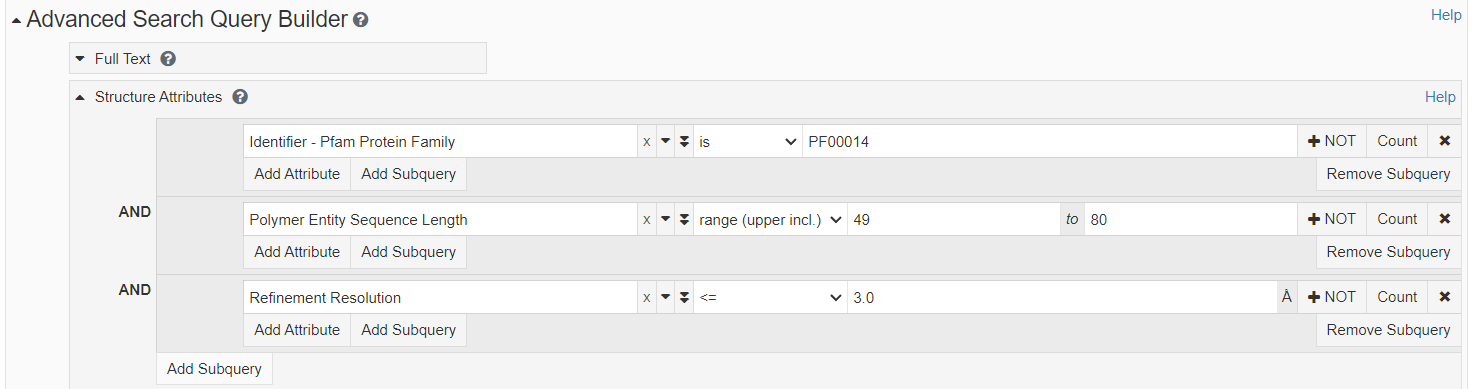
**2.Materials and Methods**

1.Selection of the protein data set from PDB

Before generating an HMM of the BPTI Kunitz domain the dataset of proteins containing the Kunitz domain was retrieved with an advanced search among the structures on PDB (<https://www.rcsb.org/>) applying the following filters:

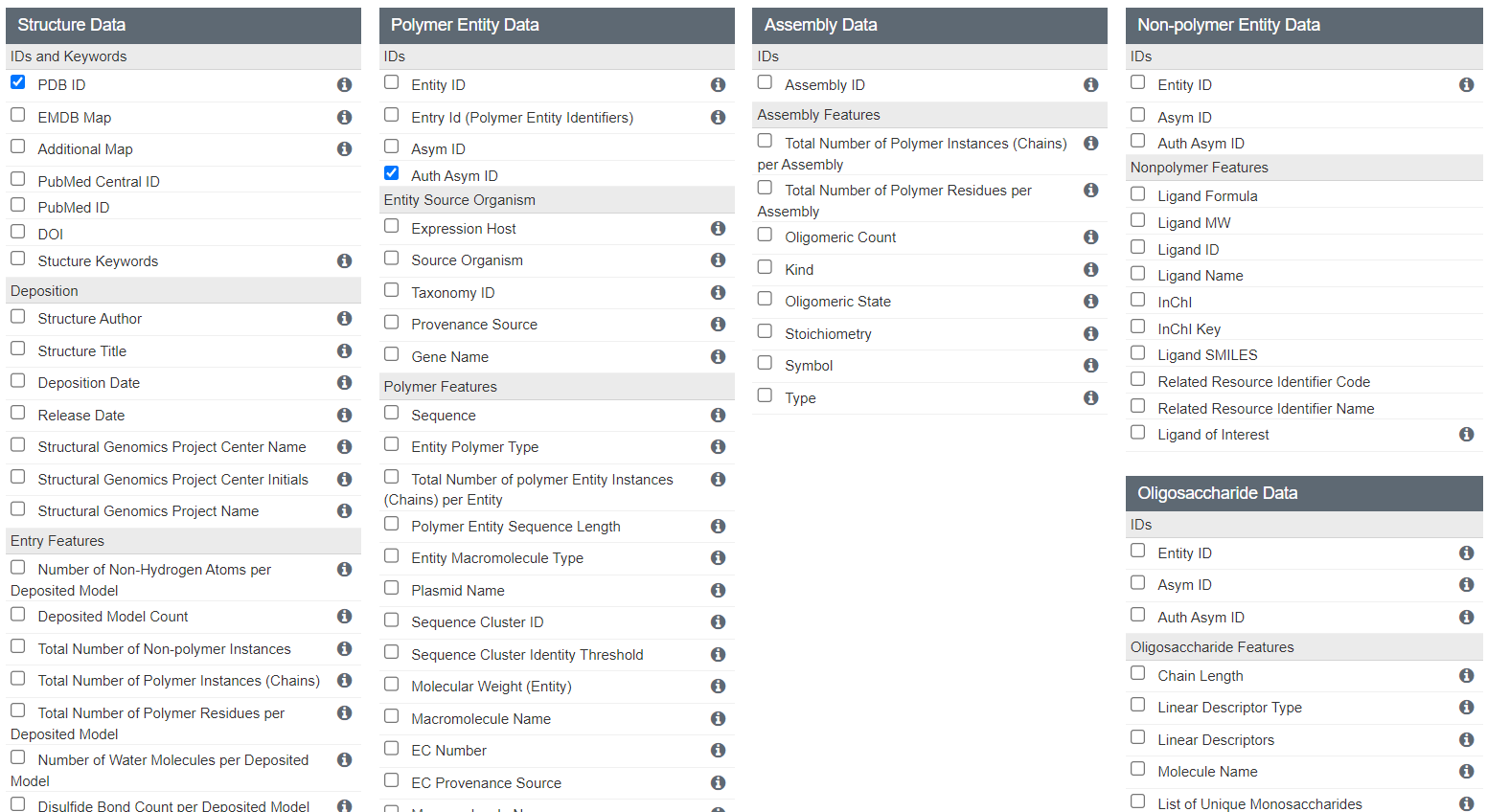
1. **PFAM domain**: we include only proteins that match with the PFAM ID PF00014, associated with the Kunitz/Bovine pancreatic trypsin inhibitor (BPTI) domain.
2. **Length of polymer chain**: we include only proteins with a length in the range 50-80 aa. to include **merely the domains themselves**.
3. **Resolution**: we chose structures with a resolution max **3.0 Å**. Structures with higher resolution would **compromise the reliability** of our model. This threshold of **3 Å** is chosen since it can model the interaction bonds and the **distance between two carbons** in the backbone. Hydrogen bonds are crucial for the formation of secondary structures and protein-protein interactions.

**QUERY:** ( Identifier = "PF00014" **AND** Annotation Type = "Pfam" ) **AND** Polymer Entity Sequence Length = [ 49 - 80 ] **AND** Refinement Resolution **<=** 3





After this filtering process there were 33 structures. The Custom Tabular Report was created with the following options selected: PDB ID, AUTH ASYM ID.

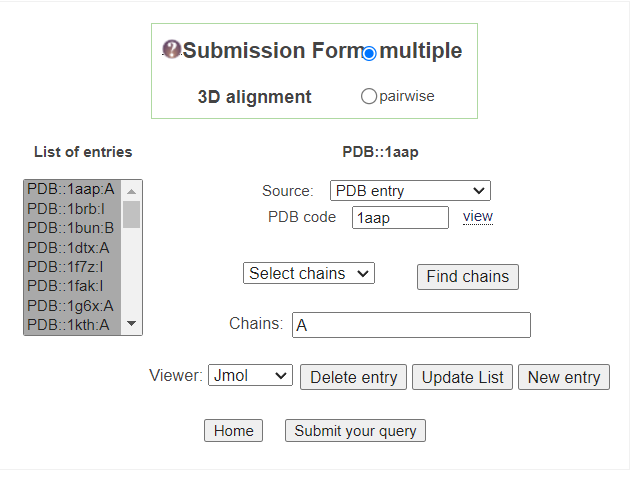


After downloading this report as csv file rcsb\_pdb\_custom\_report.csv, it was cleaned (step 1.1 and 1.2 in supplementary material, materials\_methods\_steps file) and saved as pdbe\_list.txt ( see supplementary material, pdbe\_list.txt)

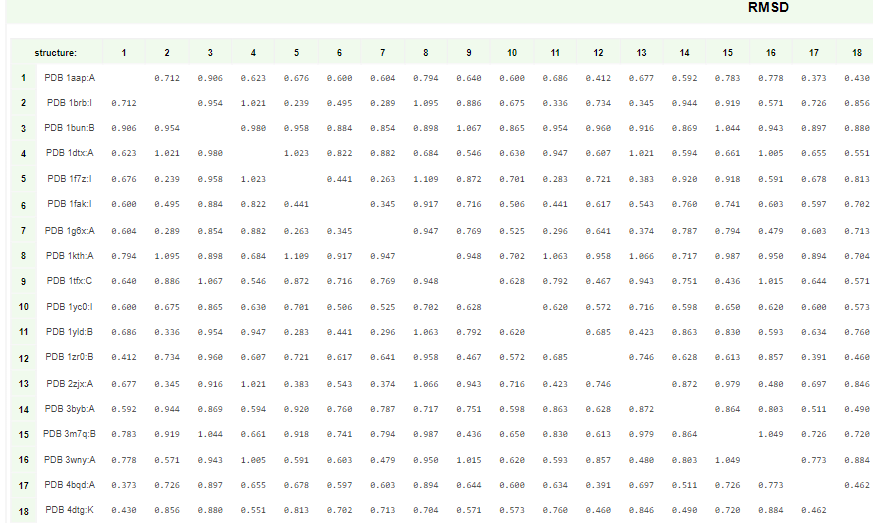
2. PDBe Fold Multiple structural alignment

PDBe fold is an algorithm that uses elements of secondary structure (SSE) to do multiple structural alignment. An alternative to PDBe Fold is **mTM-align (**<https://yanglab.nankai.edu.cn/mTM-align/>**).**

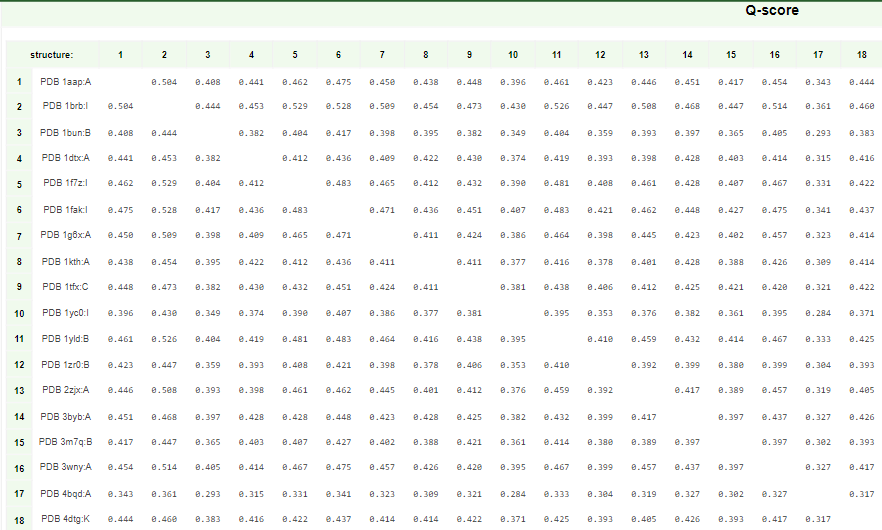
For using **PDBe fold** it’s necessary to start a session on the website of PDBe Fold (<https://www.ebi.ac.uk/msd-srv/ssm/>) and submit the cleaned file called pdbe\_list.txt.the 'multiple' box was checked and then 'List of PDB codes' was chosen. After uploading the file, the query was submitted.



In the matrix containing all the **RMSD** (between all against all), the majority of the RMSD are below 1 Å , which means they are quite good alignments.



The **Q-score** considers both **alignment length** and **RMSD value**, the higher the Q-score the better the similarity. The Q-score depends also on the size of the structure that we align and in this case it is a short domain.



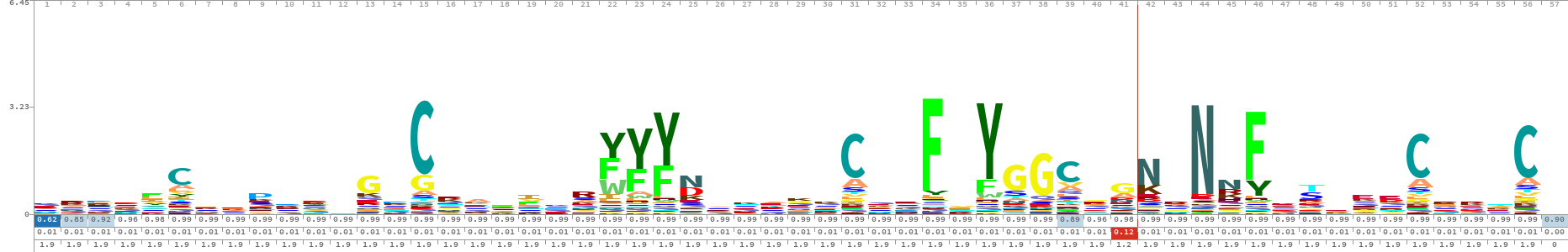
The file of the resulting MSA was downloaded in FASTA format (step 2.1 in supplementary material, materials\_methods\_steps file)

There are some parts of this alignment that are not important for this project, like the initial gap.The cleaning of this file was done transforming the fasta file into a sort of clustal format ( step 2.2 in supplementary material, materials\_methods\_steps file).

3.Generation of the Hidden Markov Model for modeling BPTI/Kunitz domain

The HMM model from the cleaned version of the alignment file was obtained (3.1 in supplementary material, materials\_methods\_steps file).

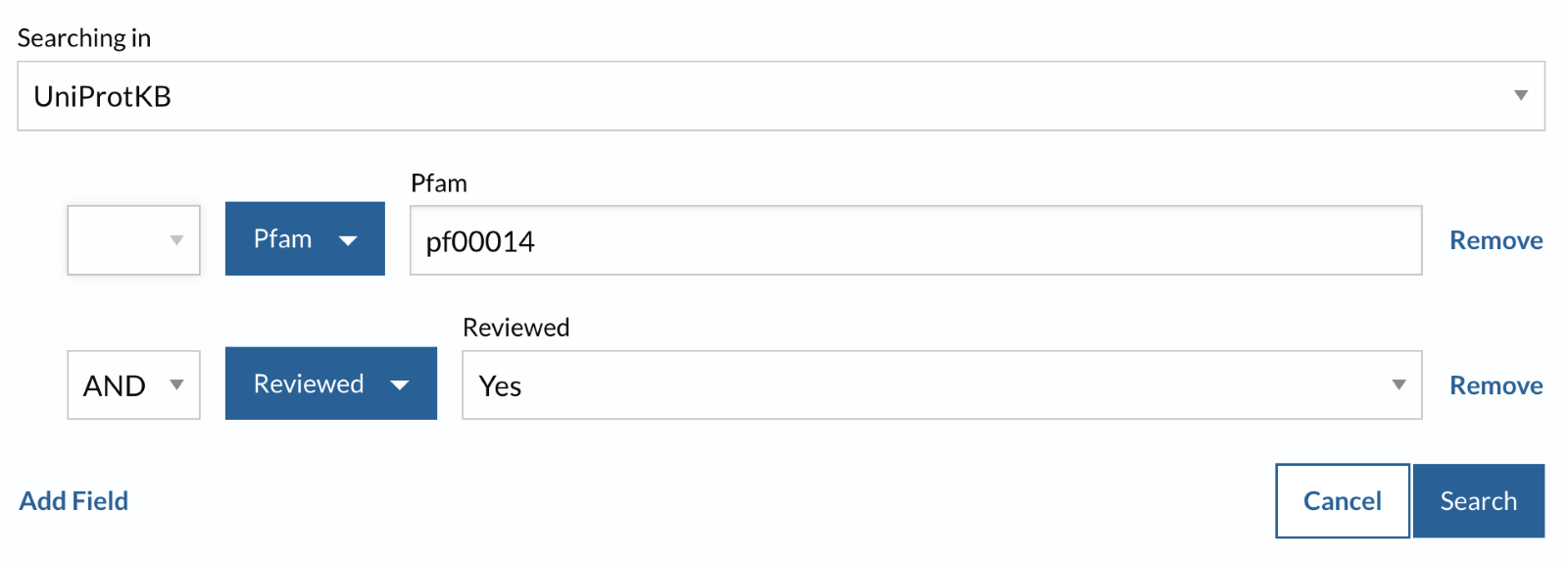
Then a sequence logo in skyalign.org with the hmm file (see supplementary material, ali\_3d\_clean.hmm) was generated.



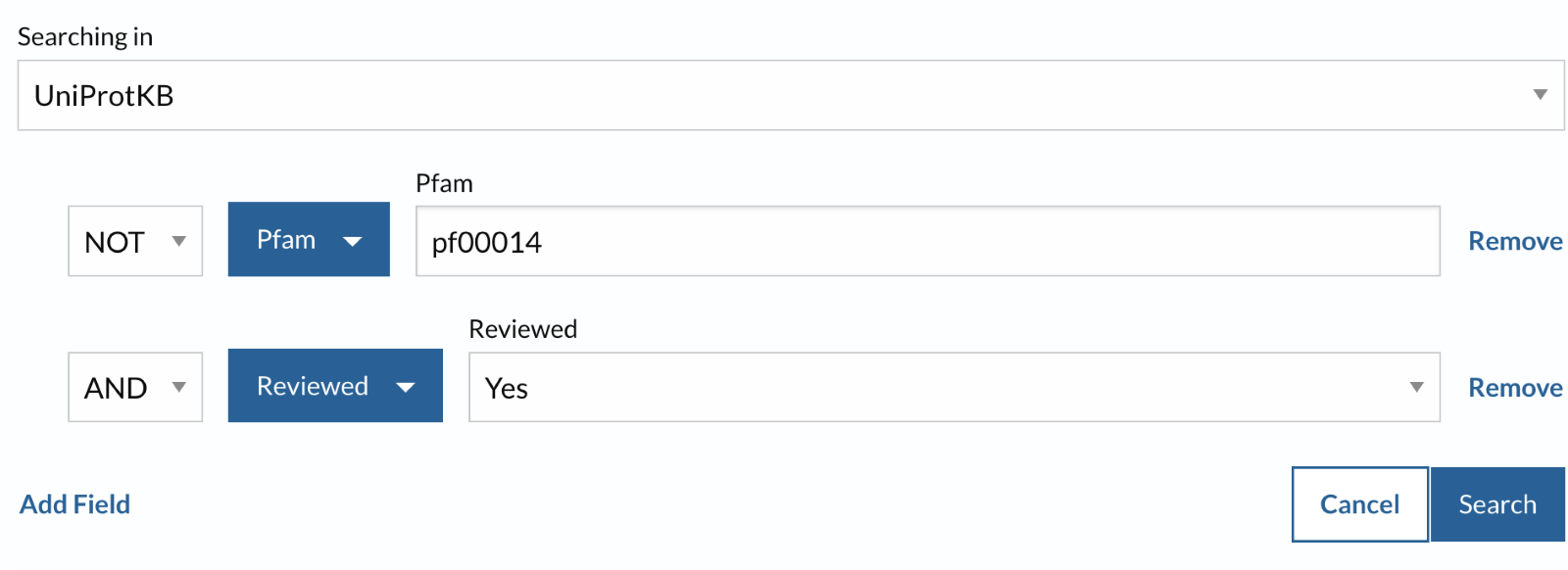
By studying the seq logo it’s clear that there are 6 cysteins(C) conserved. They are important for keeping the structure compact with three disulfide bonds. Then there are other sites that are potentially important for the function of the protein such as the tyrosines (Y).

4.Selection of the testing sets from UniProt

**positive dataset**: proteins containing Kunitz domain in SwissProt (using the Advanced search in Uniprot <https://www.uniprot.org/>)

query: (xref:pfam-pf00014) AND (reviewed:true)

**negative dataset**: proteins not containing the Kunitz domain in SwissProt (using the Advanced search in Uniprot <https://www.uniprot.org/>



query: NOT (xref:pfam-pf00014) AND (reviewed:true)

It was necessary to exclude the situation of having in the testing set the same sequences present in the training.

The main idea is that the set of ids used for building the model should be removed from the positive dataset from uniprot.

First of all the file of these initial pdb ids (see supplementary material, pdbe\_list.txt) was cleaned to obtain only the pdb ids without the chains.

Then there were few possibilities:

1) One possibility is to extract from the pdb list of ids (see supplementary material, ids\_pdb.txt) the corresponding uniprot ids with ids mapping on uniprot to obtain a list file called to\_remove.list (see in the supplementary material, to\_remove.list). Once found the correspondence between them, these uniprot ids were removed from the uniprot list of positives with few possible procedures:

1. Create a python code (see supplementary material, uniprot\_cleaning.py) to remove these ids (to\_remove.list) from the positive data set (see supplementary material, uniprot\_positive.list) and to create a new cleaned file with the final result (see supplementary material, file\_result.txt)
2. with ”comm” command

The next step consists in doing the mapping ids with the resulting file of step b (see supplementary material, ids\_uniprot\_not\_redundant.txt) that contains ids in uniprot and getting the fasta file (see supplementary material, kunitz\_clean.fasta) that is the file containing the sequences that will be used as positive test set.

2) Another possibility is to use **blast** and see when the sequences are exact matches (100%)between the pdb list of codes for the model and the uniprot positive dataset and remove them from the uniprot positive dataset. Blast is a local alignment so it will work only on a fraction of the sequence so two cutoffs should be set: one on the fraction (length of the alignment) and one cut off on sequence identity( 100%)

(step 4.3 in supplementary material, materials\_methods\_steps file)

5. Testing the model with hmm search

**hmm search** takes as input an hmm file and a fasta sequence input.

To facilitate file parsing and help the program to work in a more robust manner the following options are suggested:

--noali : don't output alignments, so output is smaller, because we only want the E-value for classification.

--max : Turn all heuristic filters off (less speed, more power)

With hmm search a set of sequences was matched with the HMM model and



the resulting file was cleaned (step 5.1 and 5.2 in supplementary material, materials\_methods\_steps file)

The cleaned file (kunitz\_clean.out) has 368 sequences like the file of the positive test dataset (kunitz\_clean.fasta) before running hmm search. So it means every sequence was found as a match, all these proteins of the positive test dataset contain the kunitz domain as expected.

The same procedure was done with negative test dataset (without kunitz domain): first hmm search and then the resulting file was cleaned (step 5.3 and 5.4 in supplementary material, materials\_methods\_steps file).The resulting file (nonkunitz\_new.out) has 33 sequences. So out of 569126 sequences that were provided in input the result is 33 matches. It was necessary to extract the identifier and the e-value and finally to add “0” (step 5.5 in supplementary material, materials\_methods\_steps file). The same step was done with the positive file with kunitz matches but “1” was added instead of “0” (step 5.6 in supplementary material, materials\_methods\_steps file).

non\_kunitz\_new.class has 33 proteins that is much less than 500k in the list of negatives at the beginning, so it was necessary to recover the ones that were missing; to have the same number of sequences as in the negative dataset at the beginning (569126), the number of missing lines can be added with an e-value that is higher than the highest e-value in the non\_kunitz\_new.class. So after checking in the non\_kunitz\_new.class that the highest e-value is 26 so to the ones that have not been identified with the hmmsearch an e-value of 100 can be assigned. The missing lines can be added checking what the missing identifiers are and assigning them higher e-value and the class 0. But it’s not strictly necessary to assign the exact missing identifiers, they can be any type of identifier. (step 5.7 in supplementary material, materials\_methods\_steps file).

In the end the non\_kunitz\_new.class file has the same size of the initial dataset ( see google drive link for negative1.fasta)

6. Method optimization and assessment

**2-fold-cross-validation:** shuffle of these 2 files (see supplementary material, non\_kunitz\_new.class and kunitz.class) and selection of half of the set in a file and half of the other set in another file to make them comparable and reversible (step 6.1 in supplementary material, materials\_methods\_steps file). Creation of a python code( see supplementary material, program.py) that takes in input a file containing protein ID, e-value and protein class (0/1) to calculate the performance of the classification method at different e-value thresholds. After running program.py on the file set 1 and file set 2 and the output was the threshold, the accuracy(acc), Matthew correlation coefficient (mcc) and the confusion matrix(cm). (step 6.2 in supplementary material, materials\_methods\_steps file).

**Optimization:** prediction could be in some cases optimized by changing the E-value threshold or by refining the training alignment. So once it was found the good threshold of e-value to be 0.01 it can be used as cutoff for running hmm search again with the negative test set to see if the number of matches changes. After cleaning the resulting file, the same procedure was done with the positive dataset (step 6.3 in supplementary material, materials\_methods\_steps file).

**ROC**: The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. In step 6.4 (see supplementary material, matherials\_methods\_steps file) a python code called ROC.py (see supplementary material, ROC.py) was created and applied to set1 (see supplementary material, tset1.txt) and set2 (see supplementary material, tset2.txt) and after extracting from the output file (see supplementary material, lists\_roc.txt) the values of FPR and TPR with 20 different threshold, importing matplotlib it was created a plot of the ROC.

**3.Results:**

**Positive dataset:** 384 proteins containing Kuntiz domain were found (step 4.1 in supplementary material, materials\_methods\_steps file and uniprot\_positive.list) and then after the removal of the 33 ids used for building the model (step 4.3 in supplementary material, materials\_methods\_steps file) a file that represents the positive dataset was created (see supplementary material, kunitz\_clean.fasta) and it contains **368** sequences.

**Negative dataset**: **569126** proteins not containing Kunitz domain were found and that will represent the negative test and it was downloaded as FASTA file (step 4.2 in supplementary material, materials\_methods\_steps file and negative1.fasta file)

**hmm search results**

the output of step 5.2 (see supplementary material, materials\_methods\_steps) is a cleaned file of kunitz domain proteins deriving from hmm search (see supplementary material, kunitz\_clean.out) that has 368 sequences like the file of the positive test dataset (see supplementary material, kunitz\_clean.fasta) before running hmm search. So it means every sequence was found as a match, all these proteins of the positive test dataset contain the kunitz domain as expected.

The output of step 5.3 (see supplementary material, materials\_methods\_steps) is a cleaned file of non kunitz domain proteins deriving from hmm search (see supplementary material, nonkunitz\_new.out) that has 33 sequences. So out of 569126 sequences that were provided in input the result is 33 matches.

Analyzing the Results:

**2-fold-cross-validation:** from step 6.2 (see supplementary material, materials\_methods\_steps file). the outputs of the performance of the model are the following:

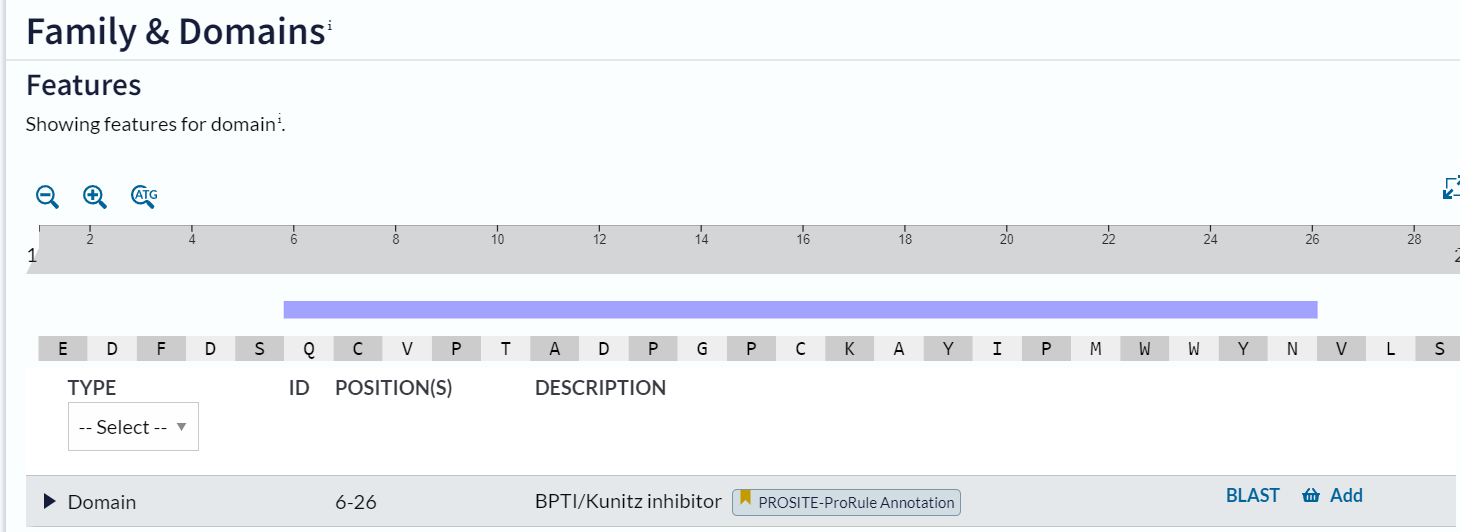
* For set 1 the optimal values of thresholds seem to be more than one because they have same highest ACC and MCC and these thresholds are 0.01, 0.001, 0.0001, 1e-05, 1e-06 (thresholds corresponding to highest ACC and MCC )
* for set 2 the best threshold is 0.01 (threshold corresponding to highest ACC and MCC)
* for set 1 and set 2 together the optimal value of threshold seems to be 0.01

**Optimization results**: in step 6.3 (see supplementary material, materials\_methods\_steps) one of the outputs is a file (see supplementary material, non\_kunitz\_new2.search) deriving from the optimization with the best threshold for running hmm search. After the cleaning, the result is a file (see supplementary material, non\_kunitz2.out) that has less sequences (1) than the ones (33) found before in the cleaned file called nonkunitz\_new.out (see supplementary material, nonkunitz\_new.out) step 5.4 (see supplementary material, materials\_methods\_steps) so it means with a better threshold than the default one hmm search found less matches. all the sequences in the negative dataset should not be recognised by hmm search as matches but actually 1 protein passed the e-value threshold. There are some possible reasons why this happened:

This protein could be false positive because the model might have detected some regions in the negative sequences that resemble the kunitz domain, but are not actually kunitz domains. This can happen if the model is not specific enough, or if the sequence contains regions with similar sequence patterns to the kunitz domain.

Incomplete filtering: It is possible that one of the negative sequences used for testing still contains residues that are part of the kunitz domain, but were not identified by the filtering method. These residues could be responsible for the detection of a false positive hit.

P84555 · TIQ7\_RHISA protein on uniprot seems to contain kunitz domain in position 6-26 so the hmm model built was correct on identifying it as kunitz domain protein, it’s a true positive. The problem could have been on the negative testing set phase where the filtering from uniprot went wrong on this protein.



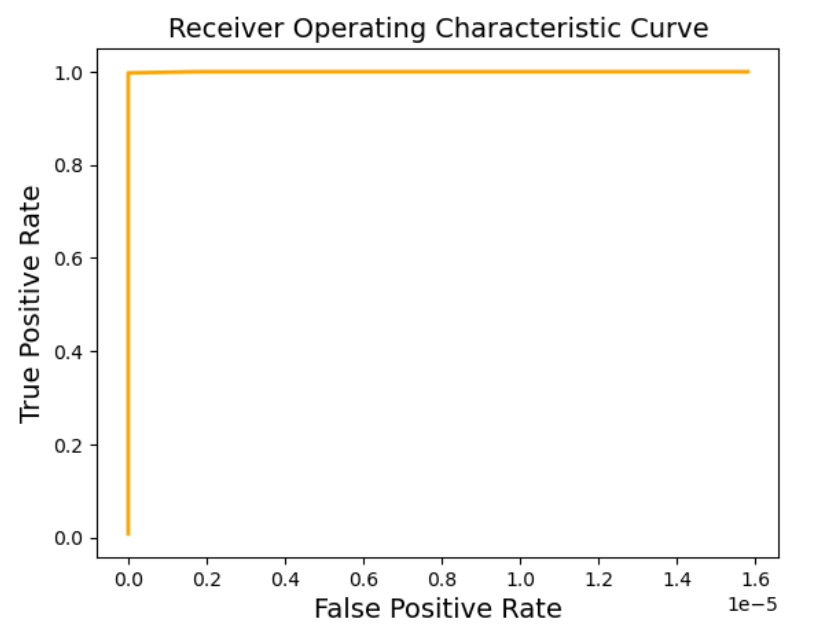
In the step 6.3 (see supplementary material, materials\_methods\_steps) hmm search with the optimal threshold of 0.01 was also runned for with the positive dataset and the resulting file shows that 368 matches were found, the same number was found in the file kunitz\_clean.out (see supplementary material, kunitz\_clean.out) with the e-value cutoff set as default(10)

This means all the positive dataset sequences were found as matches as expected, no false negatives apparently were detected in both cases of thresholds (10 and 0.01) .

**ROC results:** by calculating the area under the ROC curve (AUC) you obtain a measure of the performance of the method that is threshold-independent (because all the possible thresholds were already calculated) while other performance methods that were done previously are threshold dependent. ROC is dependent only on the predictor/model.

The closer to 1, the better the performance (graphically means that minimum level of false positive and maximum level of true positive); closer to diagonal (area of a triangle) the area will be 0,5 so a random predictor.

This is the ROC curve obtained for the hidden markov model built in this project:



This plot means the model built is a perfect classifier.(see step 6.4 in the supplementary material, materials\_methods\_steps file)

**4.Conclusion**

In conclusion, the two specific aims of this study have been successfully achieved. Firstly, a model for the Kunitz domain has been built using available structural information. This model provides valuable insights into the structural and functional characteristics of this important domain, and has the potential to advance our understanding of Kunitz domain-containing proteins. hmm search with the optimal e-value found only 1 match from the negative dataset. After analyzing the sequence on SwissProt it seems that this protein really contains the BPTI/kunitz domain (in position 6-26). This could mean that it is a true positive and the negative dataset was collected with a bad filtering on this protein.Overall, the model built is a good model for BPTI/kunitz domain as proved by the ROC curve. Finally this study has demonstrated the value of developing accurate models for protein domains, and the potential of these models for advancing our understanding of protein structure and function.