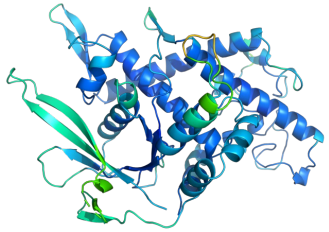


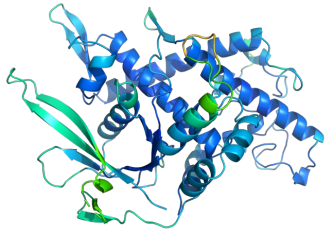
Weekly lab meeting

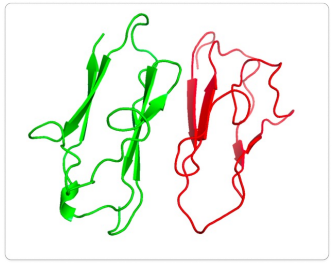
2024. 06. 10.

Edina Balázs

Protein-protein interaction prediction with deep learning language models.

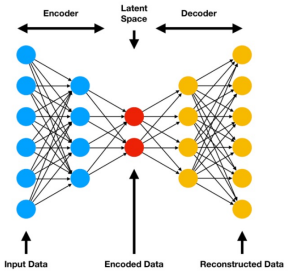
Project summary





python Keras scikit learn

CD-HIT Representative Sequences.



File extraction

PDB

Fetature extractiopn

Feature extraction single chain

- INPUT:
  - PDB file
- extract chains
- create AA matrix
- create submatrixes
- saves submatrixes as pickle
- creates backbone TSV
- OUTPUT:
  - train.tsv

Feature extraction single chain

- INPUT:
  - PDB file
- extract chains
- create AA matrix
- create submatrixes
- saves submatrixes as pickle
- creates backbone TSV
- OUTPUT:
  - train.tsv

sub_matrix.pickle

bert_train.tsv

Redundancy Remove

interactome.fasta

nonredundant sequence file

filtered_train.tsv

PDB2Fasta

- INPUT :
  - train.tsv
- reads the chain sequence from TSV
- saves as a properly formated .FASTA file
- OUTPUT:
  - interactome.fasta

CD-Hit

- INPUT:
  - interactome.fasta file
- removes sequences with 80 % similarity for representetive sequences
- OUTPUT:
  - non-redundant sequence
  - cluster file

Redundancy Remove

- INPUT :
  - the non-redundant (nr) sequence file
  - train.TSV
- removes the sequences nor present in the nr-sequence file
- OUTPUT:
  - filtered_train.tsv

Autoencoder

autoencoder_trainning_summary

autoencoder_testing_summary

bert_vectorized_train.tsv

Autoencoder training

- INPUT :
  - pickle sub matrix files
- trainies encoders and creates tsatistics about the training
- saved the autoencoder models
- OUTPUT:
  - autoencoder_trainning_summary.tsv

Autoencoder testing

- INPUT :
  - pickle sub matrix files
- loades trained encoders one by one and test them for each test file
- creates statistics about the testing
- chooses the best perfoming model with the highest correlation
- OUTPUT:
  - autoencoder_testing_summary.tsv

Adding vector

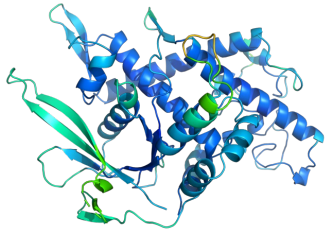
- INPUT :
  - filtered_train.tsv
- loades the best performed trained encoder
- uses to create vectors from each pickle file correspondinding to the Protein ID
- adds the vector to the chains row
- OUTPUT:
  - bert_vectorized_train.tsv

BERT

BERT training

- INPUT :
  - bert_vectorized_train.tsv
- trainies BERT model on the .tsv file
- trainies as many models as many attributes we have
- adds ttributes dinamically
- creates statistics about the training
- saves the BERT models
- OUTPUT:
  - bert_trainning_summary.tsv

Current pipeline

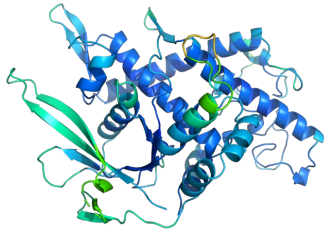


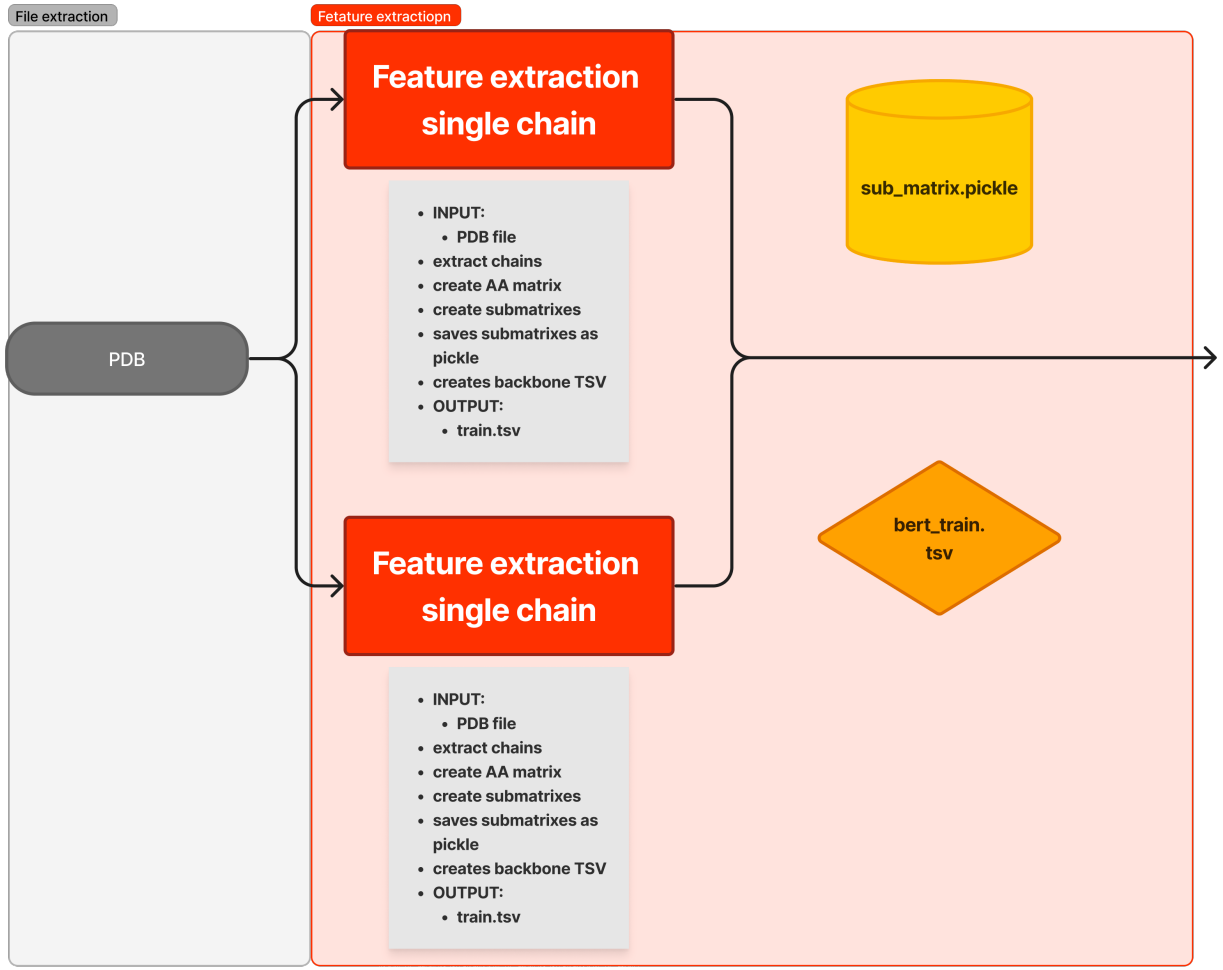
- Created a pipeline for training and validation: **main.py**
- Runs all the moduls and give to other moduls the intermediate variables and files.
- Logs all the terminal and modul outputs, to help detect faulty runs and problems.

```
2024-06-05 17:17:39,886 - INFO - Run directory created at 2024-06-05_17-17-39
2024-06-05 17:17:39,886 - INFO - Trainin directory: TRAIN
2024-06-05 17:17:39,886 - INFO -
2024-06-05 17:17:39,886 - INFO - Run TRAIN with pickle directory : /home/dina/Documents/PPI_WDLLM/Matrices_CA/train
2024-06-05 17:17:39,886 - INFO -
##########################################################################################################################################################
2024-06-05 17:17:39,886 - INFO - Run TEST with pickle directory : /home/dina/Documents/PPI_WDLLM/Matrices_CA/test
2024-06-05 17:17:40,631 - INFO - ----------------------------------------------------------------------------------------------------
2024-06-05 17:17:40,631 - INFO - Starting feature extraction for DOUBLE chain with:
sample_batch=500
sub_size=7
feature_tsv=2024-06-05_17-17-39_500_s_wDataloader.tsv
2024-06-05 17:17:40,631 - INFO - Running Feature Extraction...
2024-06-05 17:17:40,634 - INFO - /home/dina/Documents/PPI_WDLLM/workdir/TRAIN/O95259-P48547-MDD-Ion_trans-Ion_trans-6o77-D-728-981-C-728-981.pdb
2024-06-05 17:17:41,456 - INFO - CA pickle file for O95259 already exists. Skipping.
2024-06-05 17:17:41,456 - INFO - Mean pickle file for O95259 already exists. Skipping.
2024-06-05 17:17:41,456 - INFO - CA pickle file for P48547 already exists. Skipping.
2024-06-05 17:17:41,456 - INFO - Mean pickle file for P48547 already exists. Skipping.
2024-06-05 17:17:41,456 - INFO - /home/dina/Documents/PPI_WDLLM/workdir/TRAIN/P57086-Q3MJ62-MDD-SCAN-SCAN-3lhr-A-5-92-B-5-93.pdb
2024-06-05 17:17:41,511 - INFO - CA pickle file for P57086 already exists. Skipping.
2024-06-05 17:17:41,511 - INFO - Mean pickle file for P57086 already exists. Skipping.
2024-06-05 17:17:41,511 - INFO - CA pickle file for Q3MJ62 already exists. Skipping.
2024-06-05 17:17:41,511 - INFO - Mean pickle file for Q3MJ62 already exists. Skipping.
2024-06-05 17:17:41,511 - INFO - /home/dina/Documents/PPI_WDLLM/workdir/TRAIN/P25098-P29317-MDD-Pkinase-fn3-6cpg-A-133-383-B-5-85.pdb

```

Feature extraction

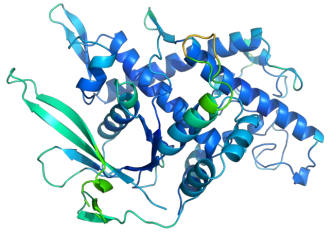




Extracting features from the file:

- Chain , sequences, C Alfa atom or Mean distance of C atoms
- Interaction site based on sequence
- DSSP (secondary-structure) , RSA (surface solvent accesability) for each **Amino Acid**
- Self matrix
- Submatrixes
- Create the backbone training .tsv

Feature extraction: positive interaction

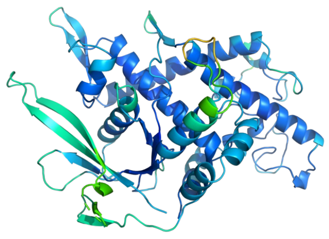


Extracting features from the PDB ad create postivive data set in the training .tsv

- Chain , sequences, C Alfa atom or Mean distance of C atoms
- DSSP (secondary-structure) , RSA (surface solvent accesability) for each **Amino Acid**
- Interaction **= 1**

Protein ID	Interact	Residues	DSSP Index	DSSP Structure
[Chain A]_[Chain B]	1 or 0	[Sequence A]_[Sequence B]	[Index A]_[Index B]	[Struct A]_[Struct B]
P27986_P62993	1	['WNVGSSNRNKAENLLRGKRDGTFLVRESSKQGCYACSVVVDGEVKHCVINKTATGYGFAEPYNLYSSLKELVLHY']_['QALFDFDPQEDGELGFRRGDFIHVMDNSDPNWWKGACHGQTGMFP']	['-' '-' '-' '-' '-' '-' 'E' '-' '-' '-' '-' '-' 'E' '-' 'E' '-' '-' 'E' '-' '-' '-' '-' '-' '-' '-' '-' '-' 'E' 'E' '-' 'E' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-']_['-' '-' '-' '-' '-' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' '-' 'E' '-' '-' 'E' '-' '-' '-' '-' '-' '-' '-' '-' 'E' 'E' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-']	tensor([0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 0])_tensor([0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 0])

Feature extraction: negative interaction

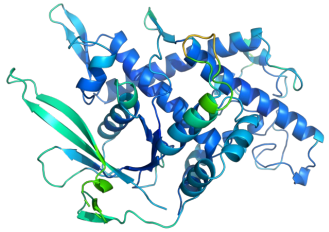


Create negative data set in the training .tsv, based in the earlier processed interacting proteins list, choosing random pairs.

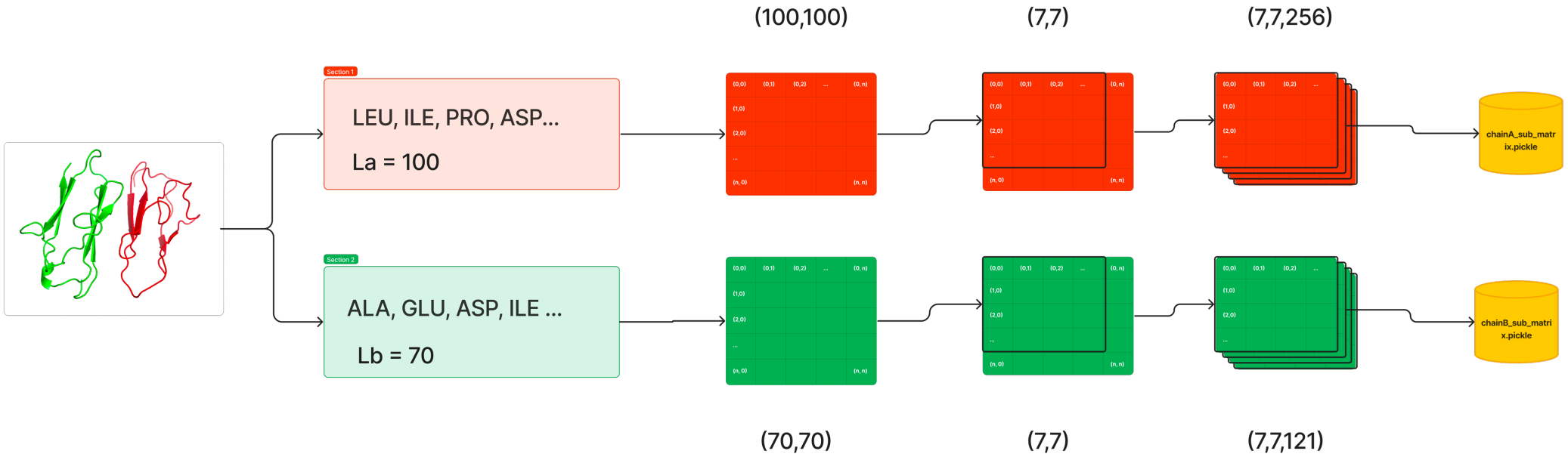
If they not interacting in the data set **interact = 0**, append the two proteins features.

Protein ID	Interact	Residues	DSSP Index	DSSP Structure
[Chain A]_[Chain B]	1 or 0	[Sequence A]_[Sequence B]	[Index A]_[Index B]	[Struct A]_[Struct B]
P27986_P62993	1	['WNVGSSNRNKAENLLRGKRDGTFLVRESSKQ GCYACSVVVDGEVKHCVINKTATGYGFAEPYN LYSSLKELVLHY']_['QALFDFDPQEDGELGFRR GDFIHVMDNSDPNWWKGACHGQTGMFP']	['-' '-' '-' '-' '-' 'E' '-' '-' '-' '-' 'E' '-' 'E' '-' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' 'E' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-']_['-' '-' '-' '-' '-' 'E' '-' '-' '-' '-' 'E' '-' 'E' '-' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' 'E' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-']	tensor([0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 0])_tensor([0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 0])
Q9P202_O75400	0	['AKAHEGLGFSIRGGSEHGVGIYVSLVEPGSLA EKEGLRVGDQILRVNDKSLARVTHAEAVKALK GSKKLVL']_['MWTEHKSPDGRTYYYNTETKQS TW']	['-' '-' '-' '-' '-' '-' '-' '-' '-' '-' '-' '-' '-' '-' 'E' '-' '-' 'E' '-' '-' '-' '-' '-' '-' '-' '-' '-' '-' '-' '-' 'H' 'H' 'H' 'H' 'H' '-' '-' '-' '-' '-' '-' '-' '-' '-' 'E' 'E' '-' '-' 'E' 'E' '-' '-' '-' '-' '-' 'H' 'H' 'H' 'H' 'H' 'H' 'H' 'H' 'H' 'H' '-' '-' '-' '-' '-' '-']_['-' '-']	tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0])_tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

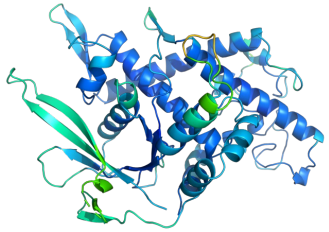
Feature extraction

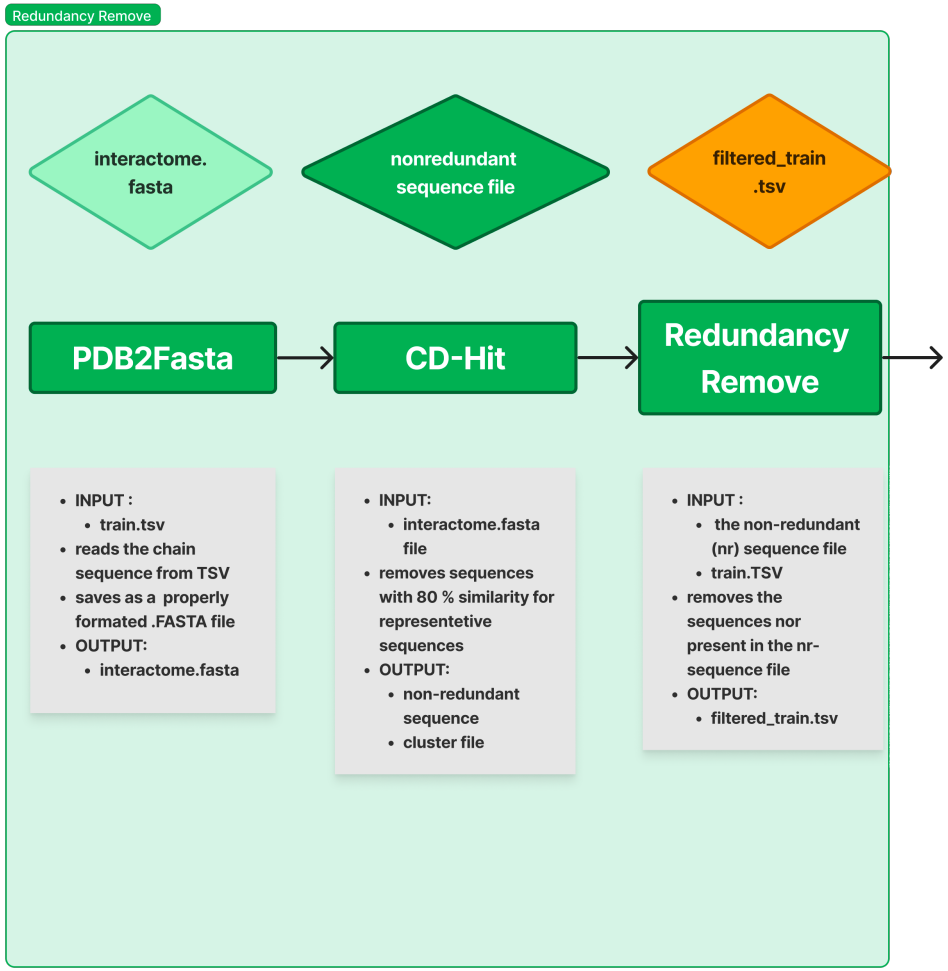


- Creating and saving sub-matrixes as .pickle files for each protein ID for the training of the Autoencoder



Redundancy remove

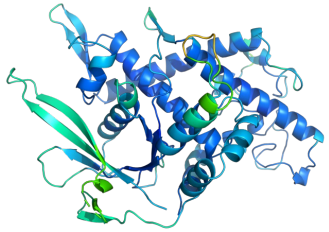


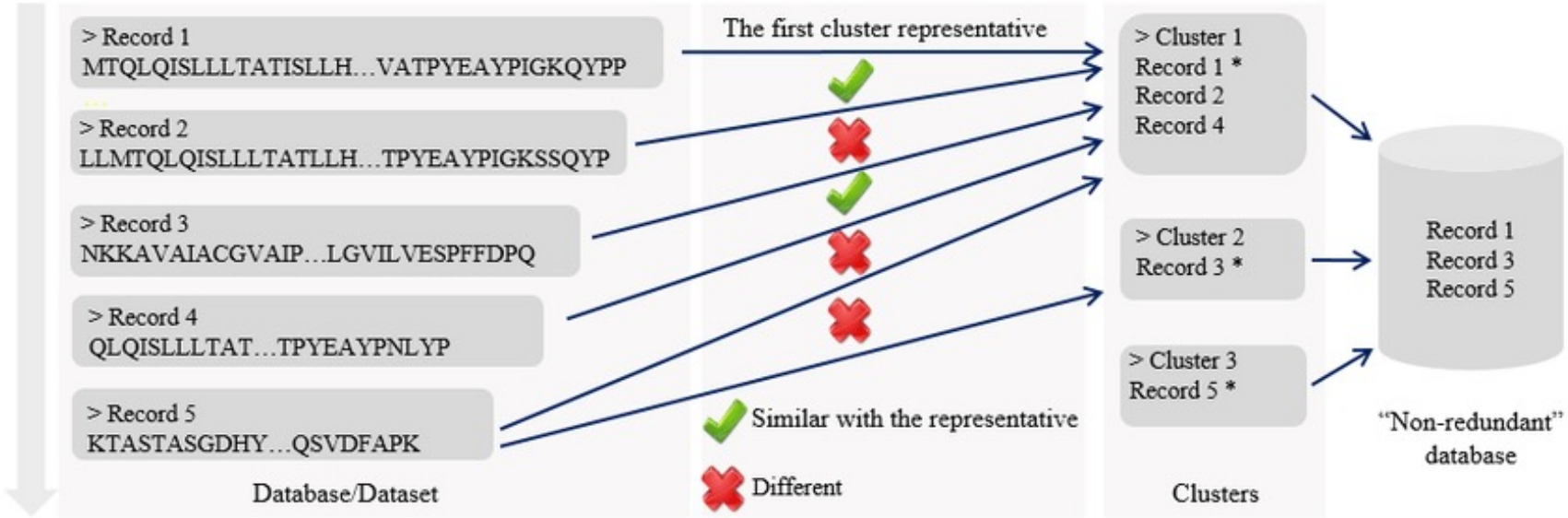


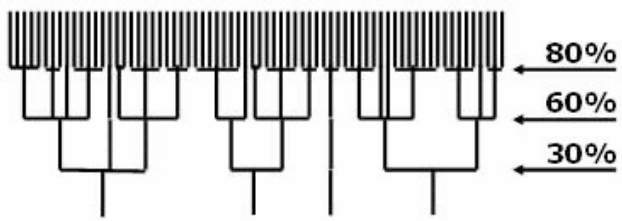
Representative sequence for training BERT:

- Read the sequences from the training .tsv
- Create an interactome.fasta file based on the sequences for CD-hit input
- CD-hit reads the interactome and creates a non-redundant list of sequence file
- Removing the redundant sequences from the training .tsv
- Outputs filtered_trains.tsv

Redundancy remove





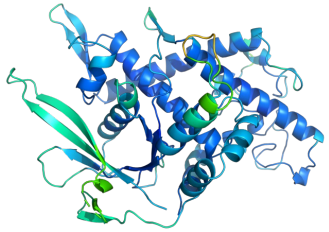


High similarity chains, with over 80% sequence match, were removed using CD-Hit to produce a set of 'non-redundant' representative sequences.

How the CD-HIT main paradigm works:

- Record 1 is the first cluster representative by default.
- Record 2 satisfies the similarity threshold in relation to Record 1 so it joins Cluster 1; same for Record 4.
- As long as a record is similar with the representative, it will join that cluster without comparing with other representatives.
- If a record is not similar to any existing representative, it will become a new cluster representative.

Redundancy remove



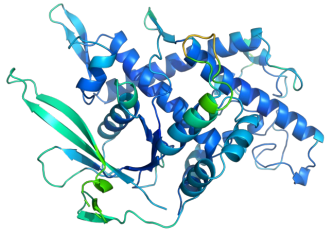
Input fatsa formation for CD-HIT :

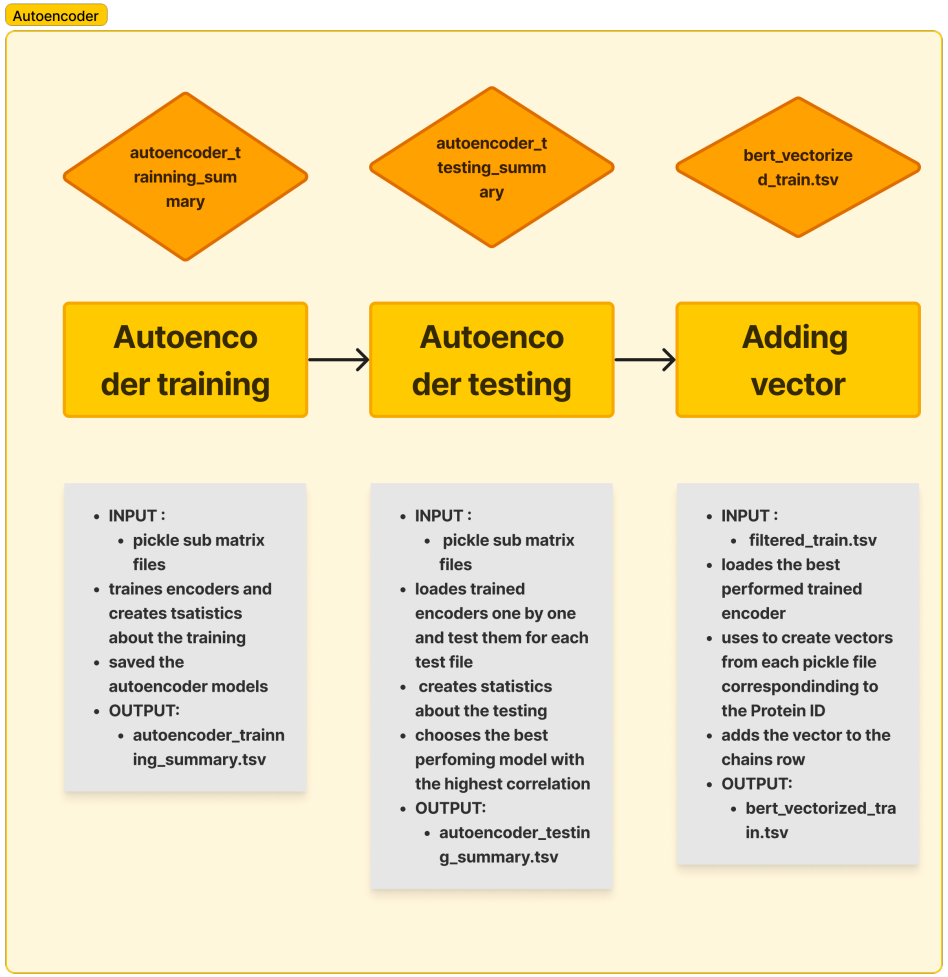
```
pdb:A
P53350-Q16659-MDD-Pkinase-Pkinase-2wnt-B-418-675-A-418-675.pdb_Chain_A
YVRGRFLGKGGFAKCFEISDADTKEVFAGKIVPKSLLLKPHQREKMSMEISIHRSLAHQHVVGFHGFFEDNDFVFVVLELCRRRSLLELHKRRKALTEPEA
RYYLRQIVLGCQYLHRNRVIHRDLKLGNLFLNEDLEVKIGDFGLATKVEYDGERKKTLCGTPNYIAPEVLSKKGHSFEVDVWSIGCIMYTLLVGKPPFETS
CLKETYLRIKKNEYSIPKHINPVAASLIQKMLQTDPTARPTINELLNDEFF
pdb:B
P53350-Q16659-MDD-Pkinase-Pkinase-2wnt-B-418-675-A-418-675.pdb_Chain_B
YMDLKPLGCGGNGLVFSAVDNDCDKRVAIKKIVLTDPQSVKHALREIKIIRRLDHDNIVKVFEILGPSGSQLTDDVGSLTELNSVYIVQEYMETDLANVLE
QGPLLEEHARLFMYQLLRGLKYIHSANVLHRDLKPANLFINTEDLVLKIGDFGLARIMDPHYSHKGHLSEGLVTKWYRSPRLLLSPNNYTKAIDMWAAGCI
FAEMLTGKTLFAGAHELEQMQLILESIPVVHEEDRQELLSVIPVYIRNDMTEPHKPLTQLLPGISREALDFLEQILTFSPMDRLTAEEALSHPYM
```

Non-redundant output:

```
P53350-Q16659-MDD-Pkinase-Pkinase-2wnt-B-418-675-A-418-675.pdb_Chain_A
YVRGRFLGKGGFAKCFEISDADTKEVFAGKIVPKSLLLKPHQREKMSMEISIHRSLAHQHVVGFHGFFEDNDFVFVVLELCRRRSLLELHKRRKALTEPEA
RYYLRQIVLGCQYLHRNRVIHRDLKLGNLFLNEDLEVKIGDFGLATKVEYDGERKKTLCGTPNYIAPEVLSKKGHSFEVDVWSIGCIMYTLLVGKPPFETS
CLKETYLRIKKNEYSIPKHINPVAASLIQKMLQTDPTARPTINELLNDEFF
P53350-Q16659-MDD-Pkinase-Pkinase-2wnt-B-418-675-A-418-675.pdb_Chain_B
YMDLKPLGCGGNGLVFSAVDNDCDKRVAIKKIVLTDPQSVKHALREIKIIRRLDHDNIVKVFEILGPSGSQLTDDVGSLTELNSVYIVQEYMETDLANVLE
QGPLLEEHARLFMYQLLRGLKYIHSANVLHRDLKPANLFINTEDLVLKIGDFGLARIMDPHYSHKGHLSEGLVTKWYRSPRLLLSPNNYTKAIDMWAAGCI
FAEMLTGKTLFAGAHELEQMQLILESIPVVHEEDRQELLSVIPVYIRNDMTEPHKPLTQLLPGISREALDFLEQILTFSPMDRLTAEEALSHPYM
```

Autoencoder for adding vector

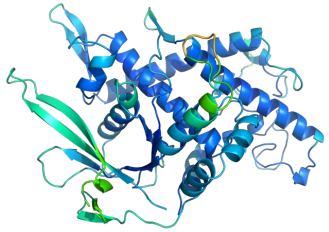


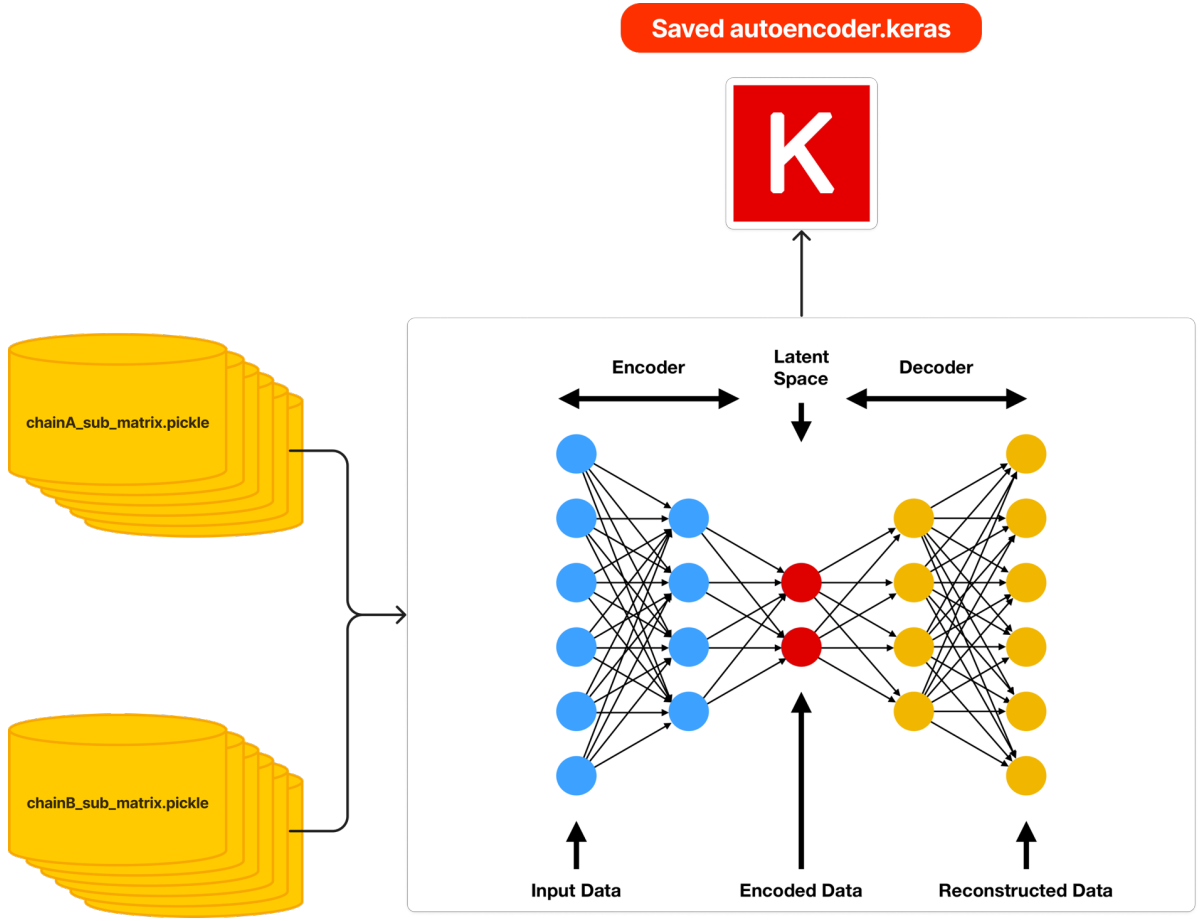


Training, testing, choosing autoencoder model and adding vectors :

- Best model is needed to produce the vector embedding from the concatenated submatrixes of each protein for the training data set of BERT

Autoencoder for adding vector

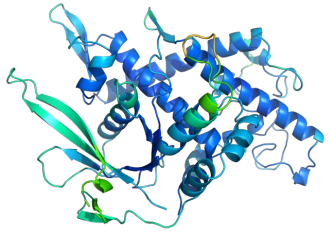


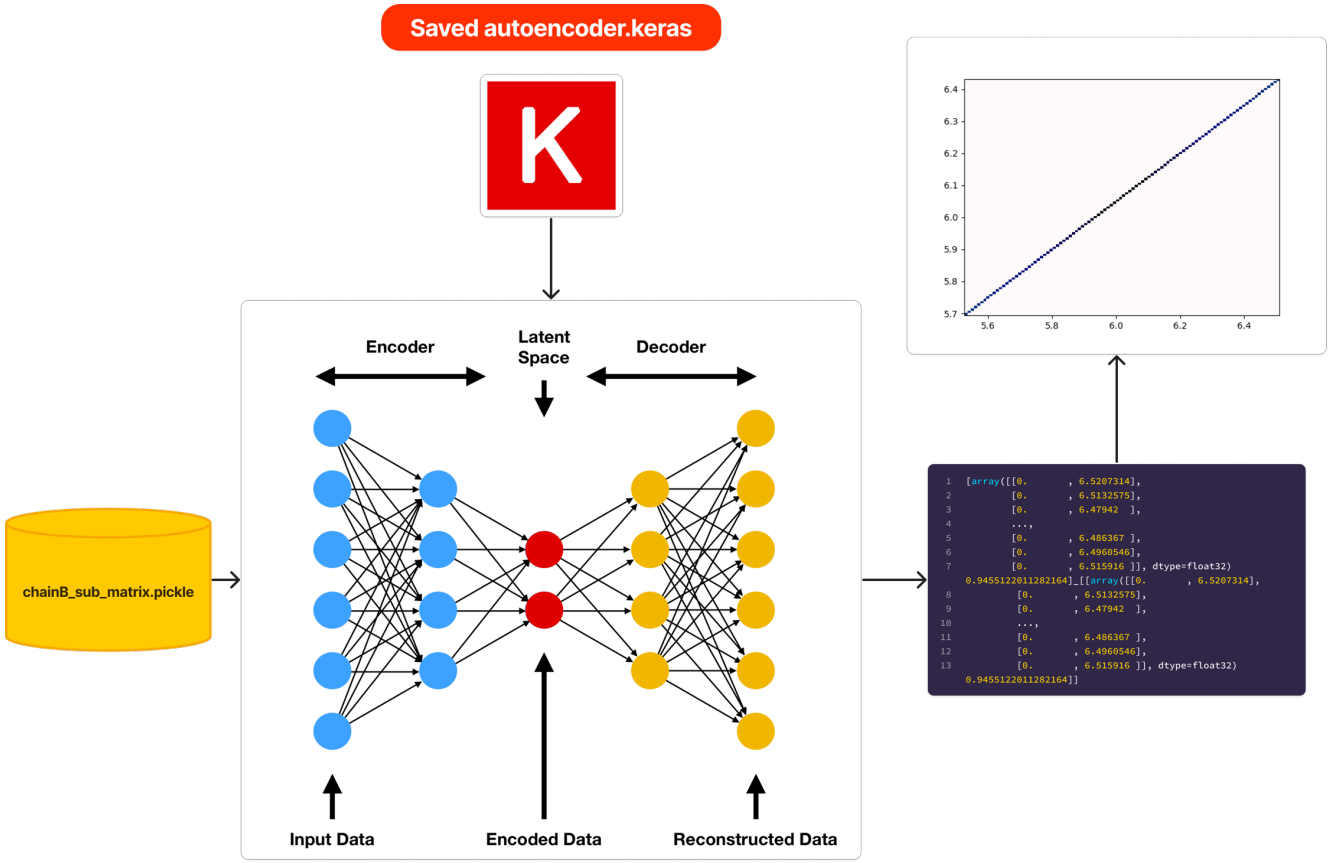


Training, testing, choosing autoencoder model, adding vectors :

1. Training:
   - Reads in all the pickles files that was created during the feature extraction
   - Several autoencoder trained based in the pickle files
   - Saves the autoencoder model
   - Creates an autoencoder_training_summary.tsv

Autoencoder for adding vector

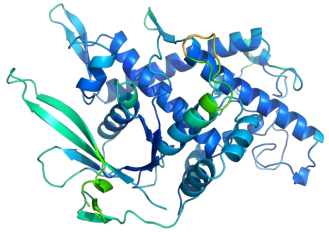


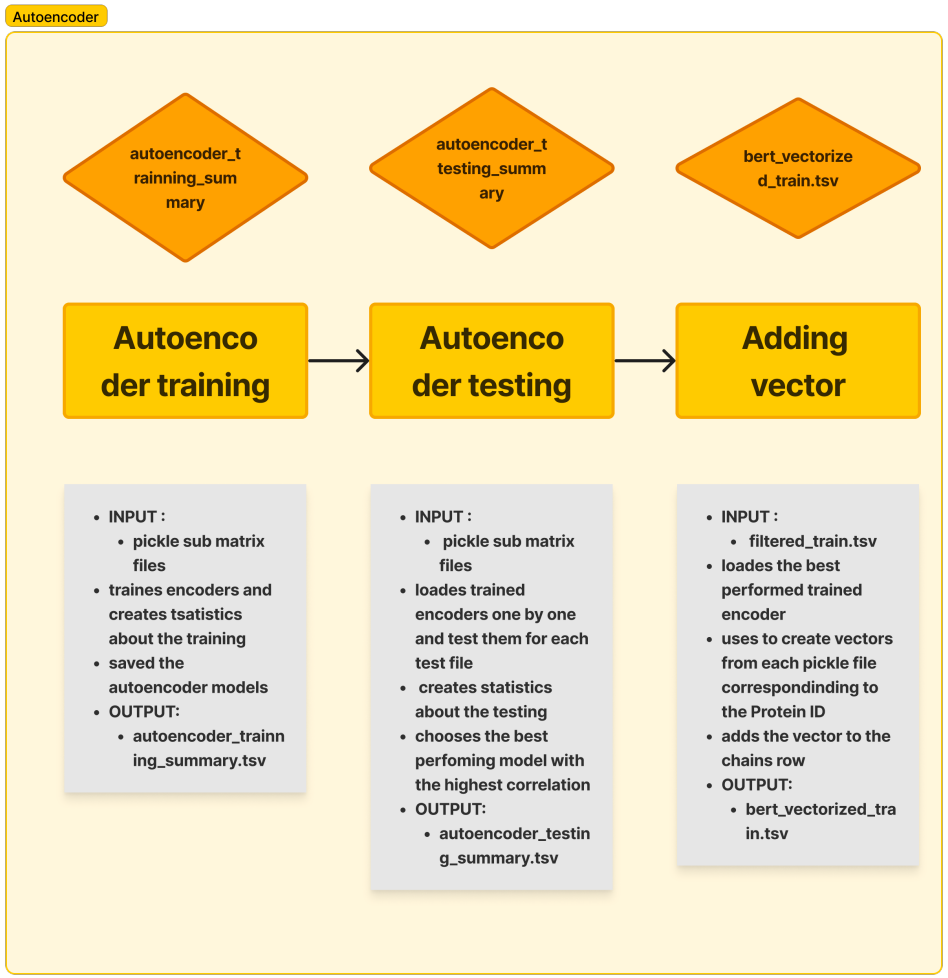


Training, testing, choosing autoencoder model, adding vectors:

1. Training
2. Testing:
   - Test all the trained and saved autoencoder on all the test files
   - Creates a autoencoder_testing_summary.tsv
   - Calculates the mean of the correlations of each autoencoder model
   - Chooses the one with the highest correlation

Autoencoder for adding vector

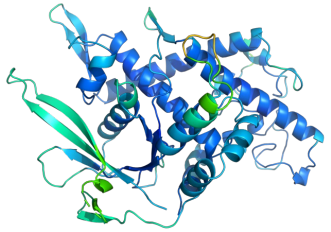




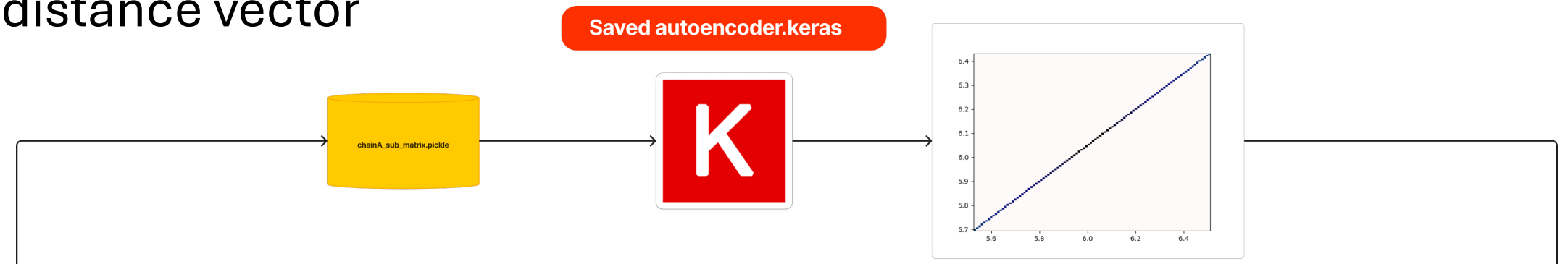
Training, testing, choosing autoencoder model, adding vectors:

1. Training
2. Testing
3. Adding Vector:
   - The choosen autoencoder model with the highest correlation
   - Best model is used to create the encoded vectors for each protein chain in the filtered_train.tsv
   - Adding the vector to the corresponding chains row in the .tsv
   - Create a bert_vectorized_train_tsv

Autoencoder for adding vector

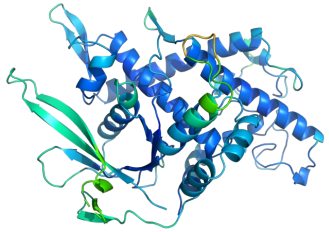


Pre-trained model to create for each chain a distance vector



Protein ID	Interact	Residues	DSSP Index	DSSP Structure	Vector
[Chain A]_[Chain B]	1 or 0	[Sequence A]_[Sequence B]	[Index A]_[Index B]	[Struct A]_[Struct B]	[Vector A]_[Vector B]
P27986_P62993	1	['WNVGSSNRNKAENLLRGKRDGTFLVRESSKQGCYACSVVVDGEVKHCVINKTATGYGFAEPYNLYSSLKELVLHY']_['QALFDFDPQEDGELGFRRGDFIHVMDNSDPNWWKGACHGQTGMFP']	['-' '-' '-' '-' '-' 'E' '-' '-' '-' '-' '-' 'E' '-' 'E' '-' '-' 'E' '-' '-' '-' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-']_['-' '-' '-' '-' '-' 'E' '-' '-' '-' 'E' '-' '-' '-' '-' '-' 'E' '-' 'E' '-' '-' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' 'E' '-' 'E' '-' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-' '-' '-' '-' '-' 'E' 'E' 'E' 'E' '-']	tensor([0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 0])_tensor([0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 0])	[array([[0. , 6.5151434], [0. , 6.5051436], [0. , 6.45627], ..., [0. , 6.490299], [0. , 6.5006433], [0. , 6.519981]], dtype=float32) 0.8566325732948376]_[[array([[0. , 6.5151434], [0. , 6.5051436], [0. , 6.45627], ..., [0. , 6.490299], [0. , 6.5006433], [0. , 6.519981]], dtype=float32) 0.8566325732948376]]

Autoencoder data loader



Problem : Training the autoencoder on large amount of data overloads the memory

Solution : Data loader and generator

- **Data Loader (create_tf_dataset function):**

Loads and preprocesses data from pickle files into a TensorFlow dataset for training and testing an autoencoder.

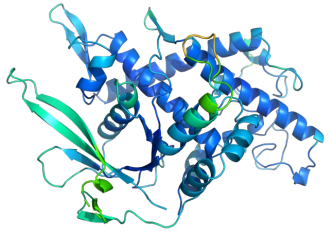
  - Loads pickle files containing preprocessed data matrices.
  - Normalizes the data by reshaping and scaling the matrices.
  - Divides the data into batches using a generator.
  - Creates a TensorFlow dataset using the generator, specifying the output shape and type.

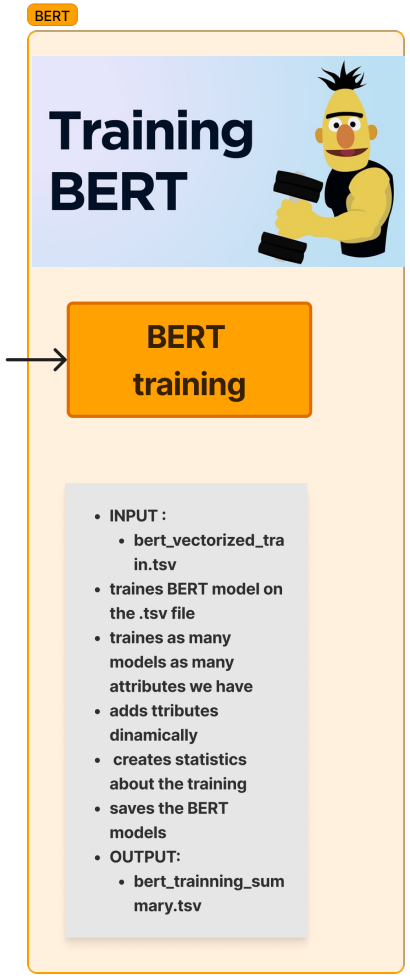
- **Data Generator (inside create_tf_dataset function):**

Generates batches of data from the loaded pickle files.

  - Iterates through each pickle file.
  - Loads and reshapes the data matrices from the file.
  - Normalizes the data by dividing by 255.
  - Splits the data into batches.
  - Yields each batch for use in the TensorFlow dataset.

BERT training





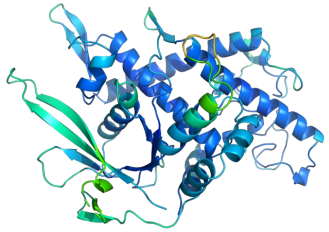
BERT is a Language model and accepts strings as an input, it was needed to augment it so it can receive numerical and categorical data at the same time for the training.

Employ **unsupervised learning** , the model is trained for **binary classification** to discern interacting (1) from non-interacting (0) protein chains. Prediction is based on the ”Interact” label .

Augmented BERT solution:

- Adding each information from the train.tsv column into a sequence of string with {:}[SEP]{:}[SEP]{:}".format .
- Adding each attribute dinamically. No need to adjust int the BERT code just the call of the BERT file.
- Saving each BERT model
- Creating a summary tsv from the run to determine the BEST perfoming BERT.

BERT training



It was needed to evaluate how each attribute added for the training data contributes to the prediction. I conducted the F1 score to validate it during training.

Training different BERT models based on how many attributes are found in the training tsv. The ‘attrnum’ in the BERT models name correspond to how many were used for the training.

Used attributes :

Sequence, DSSP index, DSSP struct, Encoded Vector

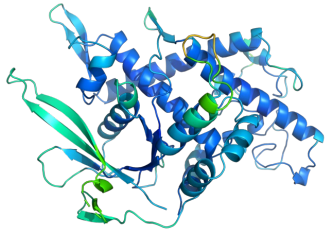
```
2024-06-05 19:03:41,961 - INFO - Running BERT model bert_attrnum_1_no_500_s_wDataloader with combined fields: ['Residues']
2024-06-05 19:04:48,642 - INFO - F1 Score: 0.676

2024-06-05 19:04:48,646 - INFO - Running BERT model bert_attrnum_2_no_500_s_wDataloader with combined fields: ['Residues', 'DSSP Structure']
2024-06-05 19:09:37,348 - INFO - F1 Score: 0.805

2024-06-05 19:09:37,352 - INFO - Running BERT model bert_attrnum_3_no_500_s_wDataloader with combined fields: ['Residues', 'DSSP Structure', 'DSSP Index']
2024-06-05 19:14:28,685 - INFO - F1 Score: 0.785

2024-06-05 19:14:28,689 - INFO - Running BERT model bert_attrnum_4_no_500_s_wDataloader with combined fields: ['Residues', 'DSSP Structure', 'DSSP Index', 'Vector']
2024-06-05 19:19:22,398 - INFO - F1 Score: 0.763
```

BERT testing



The saved BERT models were tested on unseen data.

Performance Metrics:

- **Accuracy**: Overall accuracy of the model, representing the proportion of correct predictions.

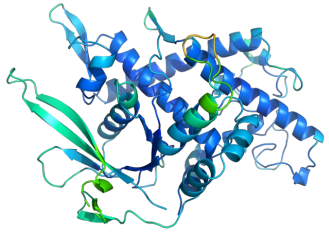
- **False Labels**:
  - Percentage of positive labels (1s) incorrectly labeled as negative (0s).
  - Percentage of negative labels (0s) incorrectly labeled as positive (1s).

- **Confusion Matrix Components**:
  - True Positives (TP): Number of correctly predicted positive cases.
  - True Negatives (TN): Number of correctly predicted negative cases.
  - False Positives (FP): Number of negative cases incorrectly predicted as positive.
  - False Negatives (FN): Number of positive cases incorrectly predicted as negative.

- **Sensitivity** : Proportion of actual positives correctly identified by the model.
- **Specificity**: Proportion of actual negatives correctly identified by the model.
- **AUC (Area Under the Curve)**: Measure of the model's ability to distinguish between positive and negative classes.

```
2024-06-05 16:27:39,864 - INFO - ##################################################
2024-06-05 16:27:39,864 - INFO - # BERT TESTING
bert_attrnum_1_no_500_s_wDataloader_negativex2
2024-06-05 16:27:39,864 - INFO -  With combined fields : ['Residues']
2024-06-05 16:27:40,146 - INFO - bert_attrnum_1_no_500_s_wDataloader_negativex2
2024-06-05 16:28:01,847 - INFO - Accuracy: 0.6150
2024-06-05 16:28:01,847 - INFO - Percentage of 1s labeled as 0: 31.10%
2024-06-05 16:28:01,847 - INFO - Percentage of 0s labeled as 1: 39.31%
2024-06-05 16:28:01,847 - INFO - True Positives (TP): 113
2024-06-05 16:28:01,848 - INFO - True Negatives (TN): 903
2024-06-05 16:28:01,848 - INFO - False Positives (FP): 585
2024-06-05 16:28:01,848 - INFO - False Negatives (FN): 51
2024-06-05 16:28:01,848 - INFO - Sensitivity: 0.6890
2024-06-05 16:28:01,848 - INFO - Specificity: 0.6069
2024-06-05 16:28:01,849 - INFO - AUC: 0.6487
2024-06-05 16:28:01,937 - INFO - ##################################################
2024-06-05 16:28:01,937 - INFO - # BERT TESTING
bert_attrnum_2_no_500_s_wDataloader_negativex2
2024-06-05 16:28:01,937 - INFO -  With combined fields : ['Residues', 'DSSP Structure']
2024-06-05 16:28:02,156 - INFO - bert_attrnum_2_no_500_s_wDataloader_negativex2
2024-06-05 16:28:25,860 - INFO - Accuracy: 0.6429
2024-06-05 16:28:25,860 - INFO - Percentage of 1s labeled as 0: 18.90%
2024-06-05 16:28:25,860 - INFO - Percentage of 0s labeled as 1: 37.57%
2024-06-05 16:28:25,860 - INFO - True Positives (TP): 133
2024-06-05 16:28:25,860 - INFO - True Negatives (TN): 929
2024-06-05 16:28:25,860 - INFO - False Positives (FP): 559
2024-06-05 16:28:25,860 - INFO - False Negatives (FN): 31
2024-06-05 16:28:25,860 - INFO - Sensitivity: 0.8110
2024-06-05 16:28:25,860 - INFO - Specificity: 0.6243
2024-06-05 16:28:25,862 - INFO - AUC: 0.7831
2024-06-05 16:28:25,902 - INFO - ##################################################
2024-06-05 16:28:25,903 - INFO - # BERT TESTING
bert_attrnum_3_no_500_s_wDataloader_negativex2
2024-06-05 16:28:25,903 - INFO -  With combined fields : ['Residues', 'DSSP Structure', 'DSSP Index']
2024-06-05 16:28:26,121 - INFO - bert_attrnum_3_no_500_s_wDataloader_negativex2
2024-06-05 16:28:51,913 - INFO - Accuracy: 0.6265
2024-06-05 16:28:51,914 - INFO - Percentage of 1s labeled as 0: 18.29%
2024-06-05 16:28:51,914 - INFO - Percentage of 0s labeled as 1: 39.45%
2024-06-05 16:28:51,914 - INFO - True Positives (TP): 134
2024-06-05 16:28:51,914 - INFO - True Negatives (TN): 901
2024-06-05 16:28:51,914 - INFO - False Positives (FP): 587
2024-06-05 16:28:51,914 - INFO - False Negatives (FN): 30
2024-06-05 16:28:51,914 - INFO - Sensitivity: 0.8171
2024-06-05 16:28:51,914 - INFO - Specificity: 0.6055
2024-06-05 16:28:51,916 - INFO - AUC: 0.7730
2024-06-05 16:28:51,956 - INFO - ##################################################
2024-06-05 16:28:51,956 - INFO - # BERT TESTING
bert_attrnum_4_no_500_s_wDataloader_negativex2
2024-06-05 16:28:51,956 - INFO -  With combined fields : ['Residues', 'DSSP Structure', 'DSSP Index', 'Vector']
2024-06-05 16:28:52,189 - INFO - bert_attrnum_4_no_500_s_wDataloader_negativex2
2024-06-05 16:29:20,680 - INFO - Accuracy: 0.6435
2024-06-05 16:29:20,680 - INFO - Percentage of 1s labeled as 0: 21.34%
2024-06-05 16:29:20,680 - INFO - Percentage of 0s labeled as 1: 37.23%
2024-06-05 16:29:20,681 - INFO - True Positives (TP): 129
2024-06-05 16:29:20,681 - INFO - True Negatives (TN): 934
2024-06-05 16:29:20,681 - INFO - False Positives (FP): 554
2024-06-05 16:29:20,681 - INFO - False Negatives (FN): 35
2024-06-05 16:29:20,681 - INFO - Sensitivity: 0.7866
2024-06-05 16:29:20,681 - INFO - Specificity: 0.6277
2024-06-05 16:29:20,682 - INFO - AUC: 0.7759
```

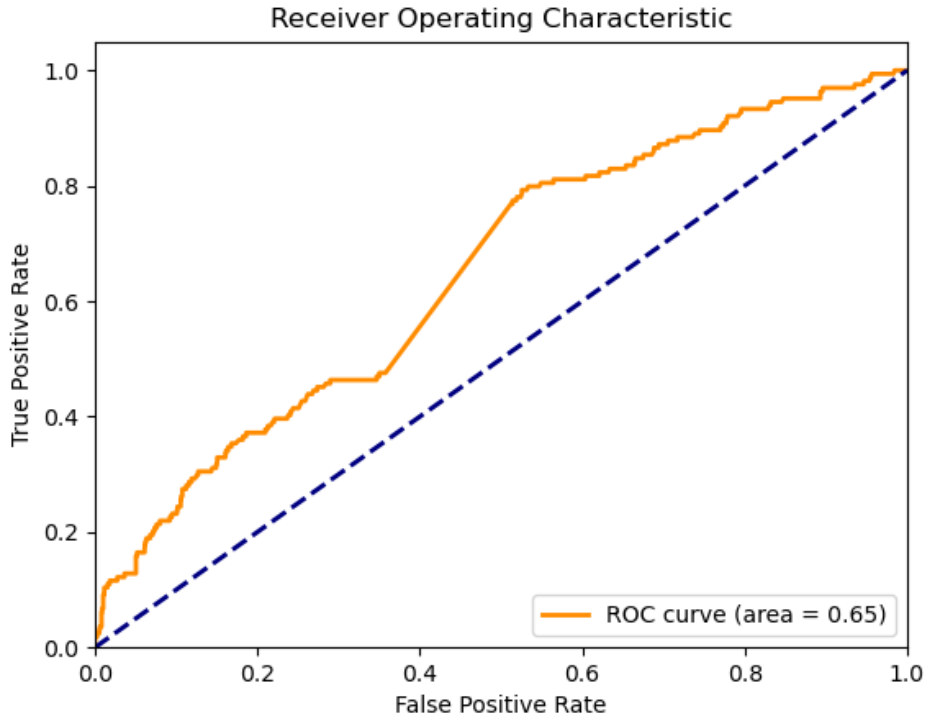
BERT testing: ['Residues']



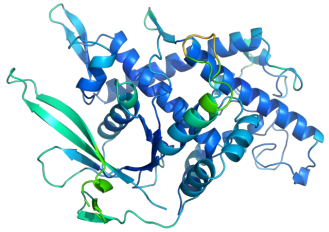
Model name : bert_attrnum_1_no_500_s_wDataloader_negativex2

With combined fields : ['Residues']

- Accuracy: 0.6150
- Percentage of 1s labeled as 0: 31.10%
- Percentage of 0s labeled as 1: 39.31%
- True Positives (TP): 113
- True Negatives (TN): 903
- False Positives (FP): 585
- False Negatives (FN): 51
- Sensitivity: 0.6890
- Specificity: 0.6069
- AUC: 0.6487



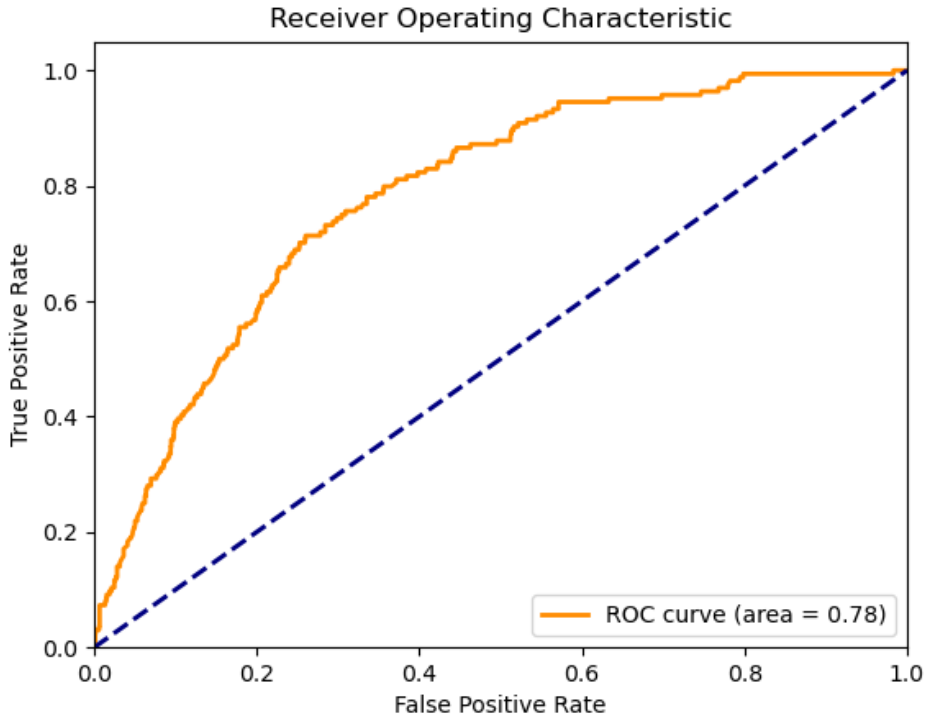
BERT testing : ['Residues', 'DSSP Structure']



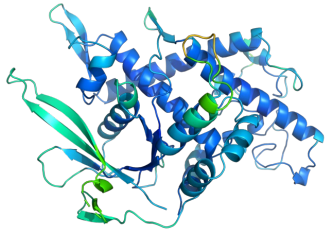
Model name : bert_attrnum_2_no_500_s_wDataloader_negativex2

With combined fields : ['Residues', 'DSSP Structure']

- Accuracy: 0.6429
- Percentage of 1s labeled as 0: 18.90 %
- Percentage of 0s labeled as 1: 37.57 %
- True Positives (TP): 133
- True Negatives (TN): 929
- False Positives (FP): 559
- False Negatives (FN): 31
- Sensitivity: 0.8110
- Specificity: 0.6243
- AUC: 0.7831



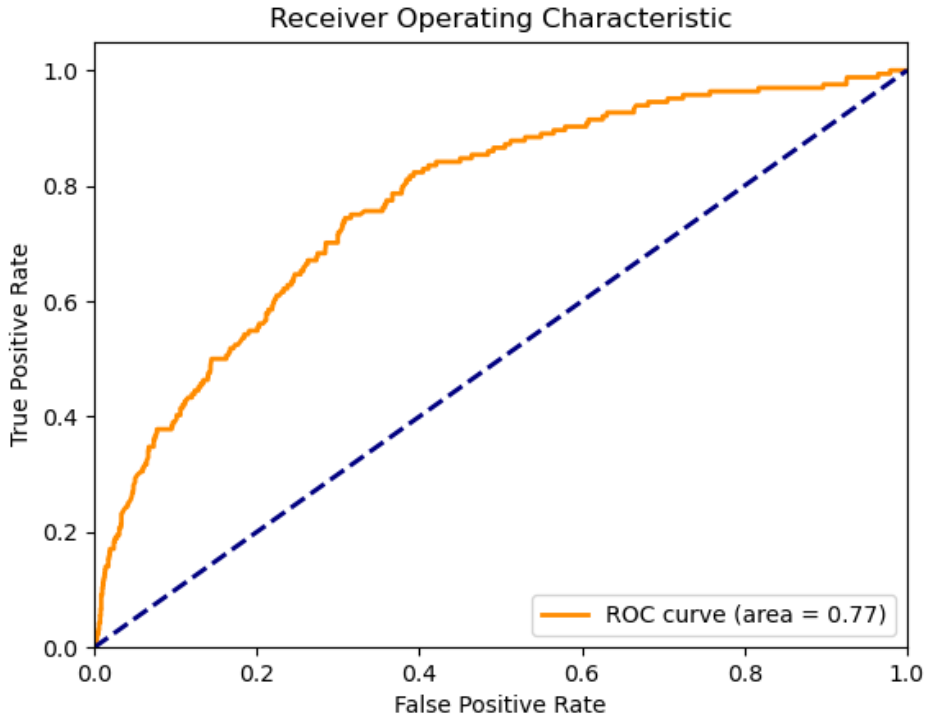
BERT testing: ['Residues', 'DSSP Structure', 'DSSP Index']



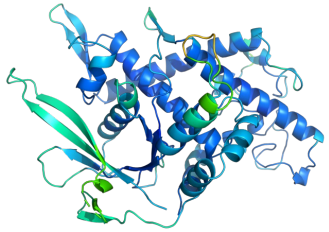
Model name : bert_attrnum_3_no_500_s_wDataloader_negativex2

With combined fields ['Residues', 'DSSP Structure', 'DSSP Index']

- Accuracy: 0.6265
- Percentage of 1s labeled as 0: 18.29 %
- Percentage of 0s labeled as 1: 39.45 %
- True Positives (TP): 134
- True Negatives (TN): 901
- False Positives (FP): 587
- False Negatives (FN): 30
- Sensitivity: 0.8171
- Specificity: 0.6055
- AUC: 0.7730



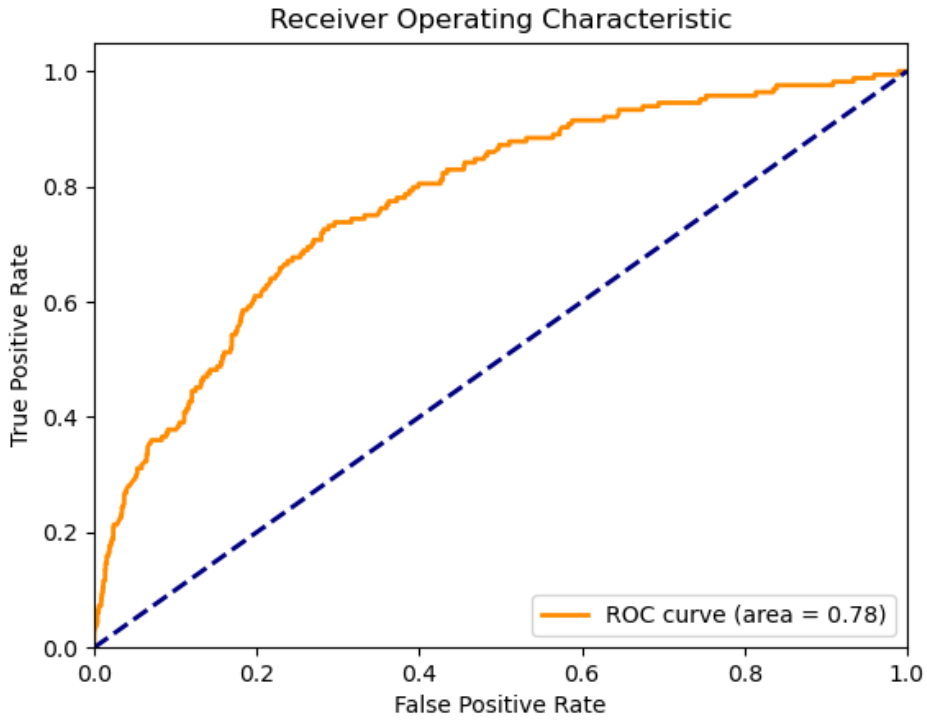
BERT testing: ['Residues', 'DSSP Structure', 'DSSP Index', 'Vector']



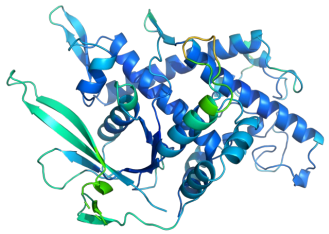
Model name : bert_attrnum_4_no_500_s_wDataloader_negativex2

With combined fields ['Residues', 'DSSP Structure', 'DSSP Index', 'Vector']

- Accuracy: 0.6435
- Percentage of 1s labeled as 0: 21.34 %
- Percentage of 0s labeled as 1: 37.23 %
- True Positives (TP): 129
- True Negatives (TN): 934
- False Positives (FP): 554
- False Negatives (FN): 35
- Sensitivity: 0.7866
- Specificity: 0.6277
- AUC: 0.7759



BERT testing: Summary



- Overall Accuracy:

  The accuracy of the models ranges between 61.50% and 64.35%, which indicates moderate performance.

- Sensitivity and Specificity:

  Sensitivity is relatively high, especially when combining more fields, indicating the model is good at identifying positive cases.

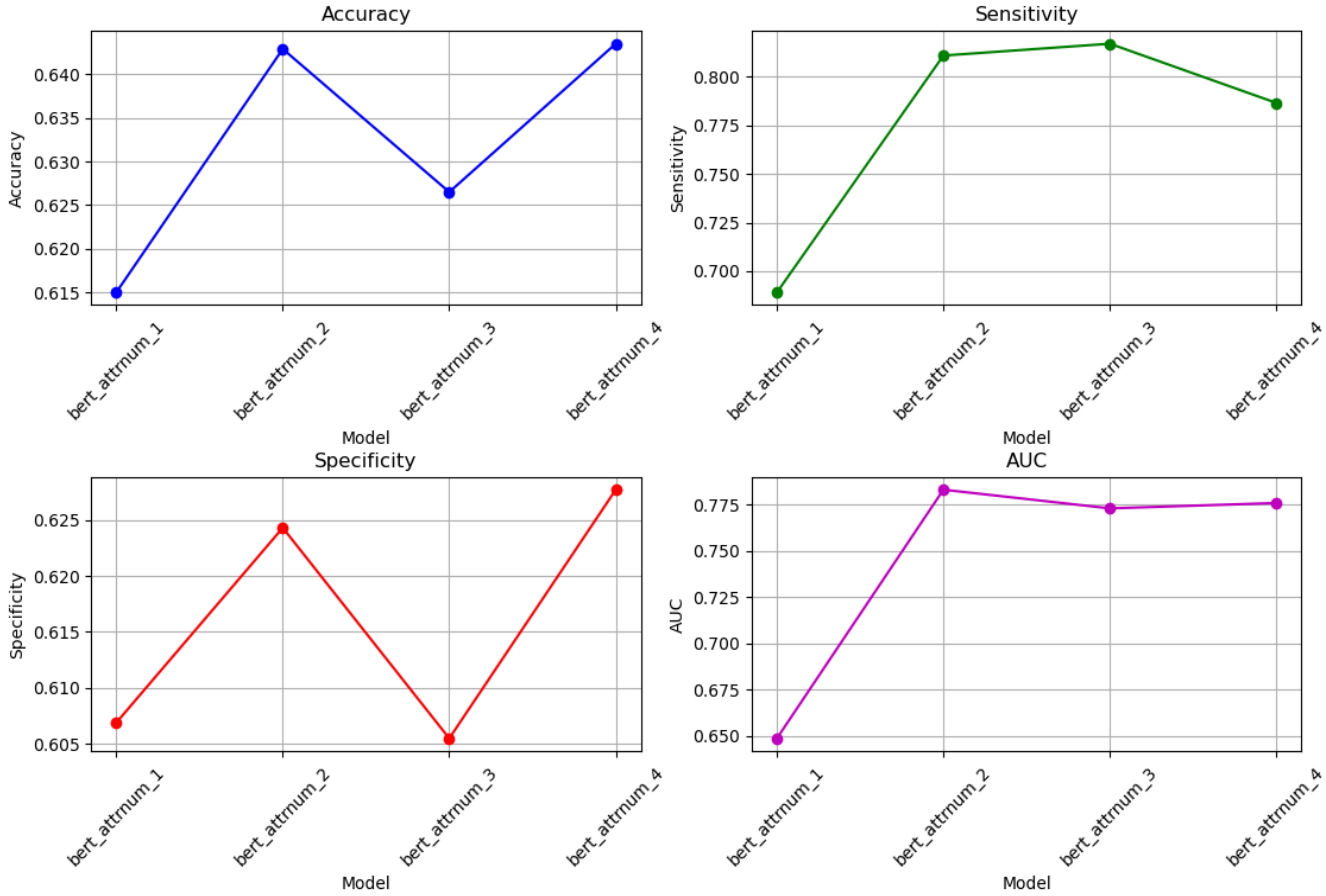
  Specificity is lower, suggesting the model struggles more with correctly identifying negative cases.

- False Labels:

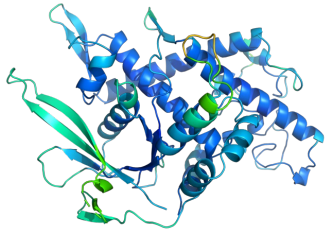
  There is a significant percentage of mislabeling, with a higher percentage of negatives being labeled as positives.

- AUC:

  The AUC values are moderate, with the highest being 0.7831, showing a decent but not excellent ability to distinguish between positive and negative cases.



HuRI, Alphafold data set for single cha



Creating new unseen data set for the testing of the trained BERT models, false positive data predictions were common, important to see if a different negative dataset would influence the training.

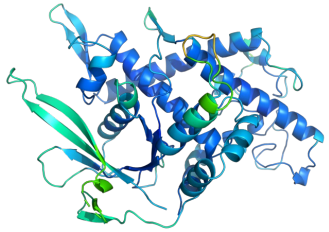
From HURI : http://www.interactome-atlas.org/download HuRI.tsv

Mapped in : https://www.uniprot.org/id-mapping from Ensembl To Uni/Swiss

Dowloaded as FASTA and interactions as TSV:

1. Parse the FASTA file and extract protein identifiers
2. Extract the UniProt identifier (e.g., 'J3KPH0', 'Q6NUM6', etc.)
3. Load the HuRI dataset
4. Load the mapping file from UniProt
5. Adjust this based on the actual column name for UniProt IDs in your mapping DataFrame : uniport_column = 'Entry'
6. Create a dictionary for quick lookup from identifiers in the 'From' column to UniProt ID
7. Map Ensembl Gene IDs in HuRI to UniProt IDs
8. Drop rows with missing mapping
9. Convert the list of UniProt IDs from the FASTA file to a set for faster lookups
10. Filter interactions where both proteins are in the AlphaFold dataset
11. Save filtered interactions

Feature extraction for single chains



Issue : Currently extraction is setted for interactom files, set up single chained .pdbs from the AlphaFold data set, based on HuRI interactions list.

New version for single chain read in and creating interaction pattern based on the positive_paris_txt attribute, which is created based on the mapping of the proteins and containes the interacting protein pairs.

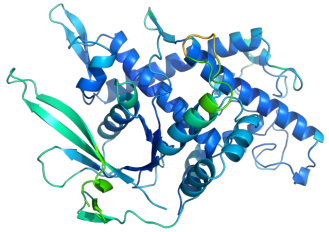
It links the chains as we did earlier as object during the interactom feature extraction.

INPUT : sample number, subsize, pdb files directory path, pickle directory path

- run feature extraction on single chain .pdb files
- need a paired.txt for the interacting protein pairs determination
- creates the train tsv and the pickle files for the vectorization

OUPUT: train.tsv, pickle files

This Week



- Training on larger data set
- Solving memory issues
- Create a true negative data set
- Fragmented surface input, based on interaction site and surface detection with other models

Important notice:

Leaving 🛫 Barcelona 🏝️ on 06.16.