Classification of contacts in protein structures

Gai Annafabia, Tiberio Filippo, Zanatta Riccardo

June 24, 2025

Introduction

In this project, the objective was to develop software capable of predicting the RING [MVB+11] classification of a contact based on a supervised deep learning model.

In this report, we present and describe several experiments on different models that after training were able to predict the contact type between residues. Moreover we verify the reliability and truthfulness of the contact types used for the training set obtained from RING by comparing them with PyMol.

Pre-processing of the data

We were given a data set of contacts obtained by RING on approximately 4000 PDB proteins. Before developing a model that can classify the type of contact between different residues, we need to understand and correctly manipulate the given input data for the training of the various models.

After retrieving the samples we cleaned the dataset from missing values and assessed the problem of multiple contacts for the same source and target residues. Initially, as it is described later, we naively removed the duplicates by following the alphabetical order of the contact types, but later in the experiments we then used a more sophisticated approach which involved studying the specification of each contact type. The specific idea was to select only one contact between the same pair of residues, by following a specific hierarchy that leverages the concept of a different level of energy strength among different kind of interactions [Bal07].

As shown in Table 1, disulfide bonds *(SSBOND)* were assigned the highest priority, as they represent covalent linkages with typical bond energies around $60\,kcal/mol$ and play a critical role in stabilizing the tertiary and quaternary structures of proteins. Ionic interactions, were prioritized next due to their electrostatic nature and relatively high energetic contribution (1–$5\,kcal/mol$), followed by hydrogen bonds, which are directional and widespread in secondary structures, with bond energies around 1–$3\,kcal/mol$. Interactions involving π-systems were ranked lower since they are generally weaker and less specific, though still relevant in stabilizing local conformations. Also to *PIHBOND* was given even lower priority according to the fact that is a more subtle and less frequently observed interaction. Finally, van der Waals *(VDW)* contacts were assigned the lowest priority due to their weak energetic contribution. These interactions arise from nonspecific atomic proximity and are often present alongside stronger, more meaningful contacts such as hydrogen bonds or ionic interactions. In addition, the interactions labeled as Missing were assigned the lowest overall rank.

This hierarchy ensures that, for any given residue pair, only the most chemically meaningful interaction is retained for downstream analysis, thereby reducing redundancy and emphasizing biologically relevant contacts.

Then we analyzed the specific features that appear in the data set, in order to understand how and which one to use in to achieve the best results. First of all, we had to transform categorical columns such as s_{ss8} and t_{ss8} into numerical values, to allow our models to understand and process them effectively. This was achieved by performing one hot encoding on all categorical features.

More over, we dropped some columns from the original feature space that were not sufficiently informative during training; consequently reducing the dimensionality on which the following models would operate. We removed the following: s_{ch}, s_{resi}, s_{ins}, s_{resn}, $s_{\text{3di\_letter}}$, t_{ch}, t_{ins}, t_{resn}, $t_{\text{3di\_letter}}$.

Table 1: Strength of different types of residue-residue interactions.

Interaction Type	**Typical Strength (kcal/mol)**
SSBOND (Disulfide bond)	∼60
IONIC (Salt bridge)	1–5
HBOND (Hydrogen bond)	1–3
PIPISTACK (π–π stacking)	0.5–2
PICATION (Cation–π)	1–3
PIHBOND (π–H bond)	∼1
VDW (van der Waals)	¡1
Missing	–

This is why we decided to remove these nine features:

- s_{ch}, t_{ch}: These features indicate the protein chain in which the amino acids are located. In the context of inferring the type of bond between the two amino acids, these two raw columns are not directly informative, but are instead combined into an engineered feature, *c_is_same_chain*, and subsequently discarded.

- s_{resi}, t_{resi}: These features represent the indices of the residues in the primary structure of the protein. On their own, these two columns are not very useful for our purpose, this is why they were dropped. However, by combining them, we obtained a new feature representing the residue distance inside the protein. This feature proved to be very useful during training. Thus, these two columns were not entirely removed, they were merged into a new feature that we called *c_dist*.

- s_{ins}, t_{ins}: These features represent charachters used to denote inserted residues. These features, like the previous ones, are not informative and, in the majority of the cases, empty. Also, since they are character-based, one-hot encoding would generate many unnecessary columns.

- s_{resn}, t_{resn}: These features specify the types of amino acids for s and t. This information is already captured by the Atchley features (numerical representations that capture physicochemical properties of amino acids), so storing this information would be redundant. Then again, being charachter-coded, they would generate many columns during one-hot encoding.

- $s_{\text{3di\_letter}}$, $t_{\text{3di\_letter}}$: These features represent the alphabet letters associated with the 3di state of the residues. This information is already stored in the features $s_{\text{3di\_state}}$ and $t_{\text{3di\_state}}$, so they can be easily dropped. We chose to remove these columns and not the other two because they are charachter-based.

Eventually the training was performed by considering the following features: s_{ss8}, s_{rsa}, s_{phi}, s_{psi}, s_{a1}, s_{a2}, s_{a3}, s_{a4}, s_{a5}, $s_{\text{3di\_state}}$, the respective features of t and the features *c_distance* and *c_is_same_chain*, calculated earlier. This resulted in reshaping the dataset size by working from 223 columns to 23 columns, leading to better performances in terms of velocity of the models.

Figure 1 represents the importance during training of each specific feature in the prediction of contact classification. It can be seen that the feature we introduced: *c_distance*, retrieves the highest score.

Finally, before starting to train the models, we split the dataset into training, validation and test set, respectively 60%, 20%, 20% of the whole data set.

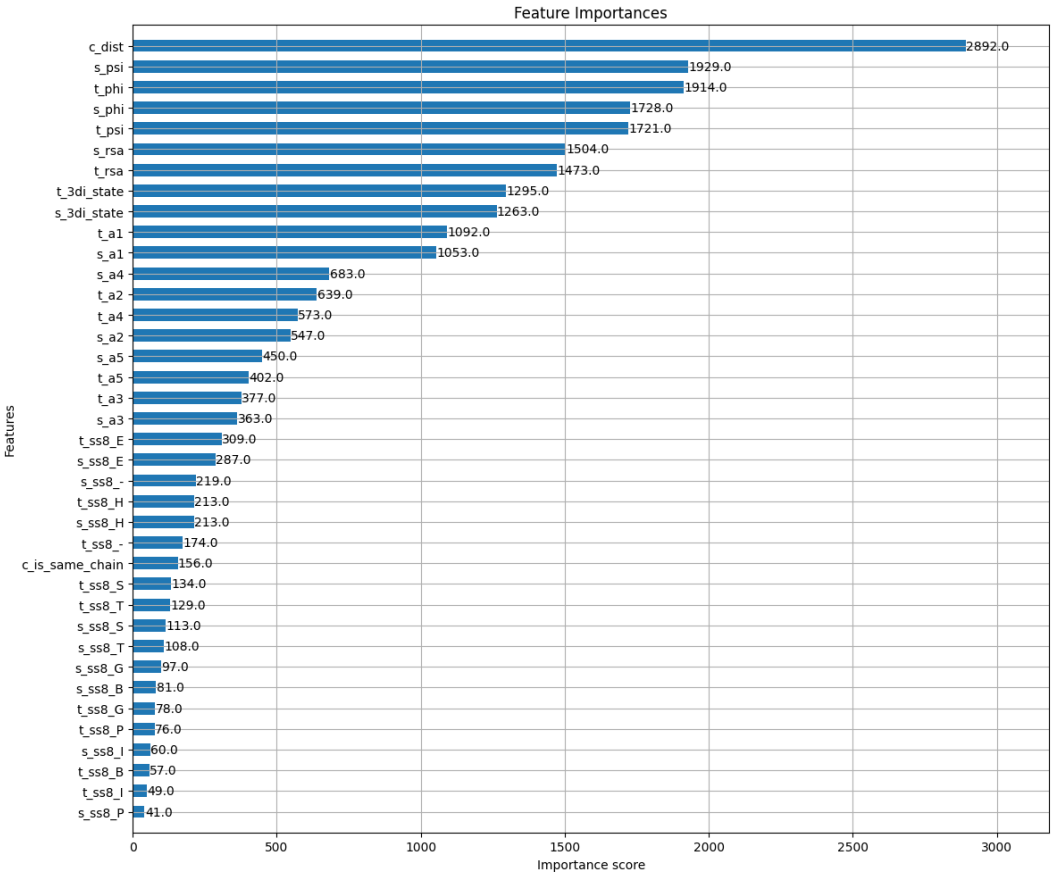

Figure 1: Representation of features importance during the training of the models

Models

Metrics

Before analyzing each of the models we implemented, we briefly describe some of the main metrics used to assess the quality of each experiment.

- *Accuracy*: proportion of correct predictions
- *Precision* with three different variants:
  - *micro*: TP / (TP + FP) computed globally across all classes
  - *macro*: average of per-class precision (treats all classes equally)
  - *weighted*: macro average weighted by the number of samples per class
- *Recall*: how many true positive detected with the same variants as for precision
- *F1_score*: mean of precision and recall with the same variants as for precision
- *ROC curve*: measure of ability to distinguish between classes
- *Matthews correlation coefficient*: correlation coefficient between true and predicted labels

Gaussian Naive Bayes

As a baseline model, from which future experiments will be compared to, we used the Gaussian Naive Bayes classifier, provided by the professor,which is a simple probabilistic model based on Bayes' theorem that assumes features follow a Gaussian (normal) distribution and are conditionally independent

given the class. For this experiment we removed the duplicates for equal contacts by alphabetical order and not by hierarchy ranking. After training, predictions were made on the test set and performance was evaluated using a confusion matrix and classification report to assess accuracy, precision, recall and F1-score across all classes.

Class	**Precision**	**Recall**	**F1-score**	**Support**
HBOND	0.40	0.42	0.41	102,762
IONIC	0.12	0.82	0.20	3,393
Missing	0.43	0.57	0.49	106,096
PICATION	0.07	0.20	0.10	791
PIHBOND	0.00	0.00	0.00	178
PIPISTACK	0.30	0.95	0.45	3,705
SSBOND	0.39	1.00	0.56	230
VDW	0.31	0.01	0.02	71,888
Accuracy			0.39	289,043
Macro avg	0.25	0.50	0.28	289,043
Weighted avg	0.39	0.39	0.34	289,043

Table 2: Gaussian Naive Bayes classification results

Metric	**Value**
Matthew's Correlation Coefficient	0.1148
Balanced Accuracy	0.4809
Average Precision Score	0.2394
ROC AUC (One-vs-One)	0.8366
ROC AUC (One-vs-Rest)	0.7939

Table 3: Specific metrics for the Gaussian Naive Bayes model

XGBOOST without interaction hierarchy

As second experiment we employed the XGBoost classifier (Extreme Gradient Boosting), which builds an ensemble of decision trees in a sequential manner, where each new tree corrects the errors made by the previous ones, optimizing a differentiable loss function. It is particularly well-suited for tabular data that might contained noisy values. In our setup, the model was trained using the logloss evaluation metric and we removed the duplicate contacts by alphabetical order. After training, the model was evaluated on the test set using both class predictions and predicted class probabilities, achieving an accuracy of 0.58, representing a first improvement over the Naive Bayes model. Improvements can be also seen in terms of the other metrics as shown in Table 4 and Table 5.

Class	**Precision**	**Recall**	**F1-score**	**Support**
HBOND	0.59	0.75	0.66	205,416
IONIC	0.34	0.11	0.17	6,805
Missing	0.58	0.81	0.68	212,444
PICATION	0.29	0.06	0.10	1,748
PIHBOND	0.00	0.00	0.00	340
PIPISTACK	0.45	0.80	0.57	7,439
SSBOND	0.44	0.62	0.52	434
VDW	0.31	0.02	0.04	143,460
Accuracy			0.58	578,086
Macro avg	0.38	0.40	0.34	578,086
Weighted avg	0.51	0.58	0.50	578,086

Table 4: XGBOOST results without interaction hierarchy

Metric	Value
Matthew's Correlation Coefficient	0.3780
Balanced Accuracy	0.3961
Average Precision Score	0.3763
ROC AUC (One-vs-One)	0.8818
ROC AUC (One-vs-Rest)	0.8926

Table 5: Specific metric for XGBOOST model without interaction hierarchy

XGBOOST with interaction hierarchy

In this experiment, we finally incorporated the hierarchical ranking of interactions when removing duplicate contacts to evaluate how it would affect classification performance. The results demonstrate the significance of this ranking, as the accuracy improved to 0.67 and a general improvement of the other metrics. These results come from the fact that when adopting this hierarchical ranking we avoid to construct a dataset with multiple overlapping or weak interactions that may obscure key structural features. And second, it helps disambiguate cases where several types of interactions co-occur, allowing the model to focus on the most functionally relevant one. However, the main challenge remains, as evidenced by the confusion matrix: accurately classifying Van der Waals (VDW). This difficulty arises not only because VDW features often overlap with other interaction types, but also due to their inherent chemical and structural nature. VDW interactions occur at very close spatial proximity between residues and frequently coexist with other types of interactions, making them intrinsically harder to distinguish.

Class	Precision	Recall	F1-score	Support
HBOND	0.75	0.75	0.75	199,220
IONIC	0.55	0.45	0.49	6,872
Missing	0.62	0.81	0.70	212,337
PICATION	0.46	0.10	0.17	1,326
PIHBOND	1.00	0.05	0.09	149
PIPISTACK	0.57	0.85	0.68	5,383
SSBOND	0.76	0.96	0.85	409
VDW	0.37	0.00	0.00	69,347
Accuracy			0.67	495,043
Macro avg	0.63	0.50	0.47	495,043
Weighted avg	0.63	0.67	0.62	495,043

Table 6: XGBOOST results with interaction hierarchy

Metric	Value
Matthew's Correlation Coefficient	0.4573
Balanced Accuracy	0.4951
Average Precision Score	0.6337
ROC AUC (One-vs-One)	0.9297
ROC AUC (One-vs-Rest)	0.9193

Table 7: Specific metric for XGBOOST model with interaction hierarchy

XGBOOST without VDW interaction

Here we tried to handle the problem of miss classification of VDW contacts by removing them from the previous trained model. This choice was made because they also represent the least important type of interactions between residues, as shown in 1. They might be inferred from the classification of other kind of interactions, for instance when predicting PIPISTACK interaction we might as well predict a VDW contact since they are a direct consequence of Pipistack contacts.

By removing this class we achieved the results at Table 8, the value of accuracy was further improved.

Class	Precision	Recall	F1-score	Support
HBOND	0.79	0.75	0.77	199,095
IONIC	0.58	0.44	0.50	6,904
Missing	0.78	0.82	0.80	212,256
PICATION	0.52	0.12	0.19	1,309
PIHBOND	0.89	0.05	0.10	158
PIPISTACK	0.61	0.82	0.70	5,550
SSBOND	0.82	0.94	0.87	426
Accuracy			0.78	425,698
Macro avg	0.71	0.56	0.56	425,698
Weighted avg	0.78	0.78	0.78	425,698

Table 8: XGBOOST results without VDW contacts

Metric	Value
Matthew's Correlation Coefficient	0.5833
Balanced Accuracy	0.5603
Average Precision Score	0.7200
ROC AUC (One-vs-One)	0.9580
ROC AUC (One-vs-Rest)	0.9595

Table 9: Specific metric for XGBOOST model without VDW contacts

Advanced XGBOOST

In this model we used the hierarchy of interactions type, we excluded VDW interactions and to increase the performance results we performed hyperparameter tuning via Randomized Cross-Validation to find the optimal hyperparameter for the model. The model was then trained and tested using these settings, resulting in the final and best-performing model with accuracy level of 0.82.

Improvements are particularly notable in precision, recall, and F1-score metrics across most classes. For example, the IONIC class' recall increased from 0.44 to 0.59, and the PICATION class' F1-score improved from 0.19 to 0.48, reflecting better sensitivity to minority classes. The macro-average metrics also show progress, with F1-score rising from 0.56 to 0.68, highlighting a more balanced performance across all classes. We selected this as our final model.

Class	Precision	Recall	F1-score	Support
HBOND	0.82	0.81	0.82	199,095
IONIC	0.66	0.59	0.62	6,904
Missing	0.82	0.84	0.83	212,256
PICATION	0.71	0.37	0.48	1,309
PIHBOND	0.85	0.25	0.39	158
PIPISTACK	0.73	0.83	0.78	5,550
SSBOND	0.84	0.87	0.86	426
Accuracy			0.82	425,698
Macro avg	0.78	0.65	0.68	425,698
Weighted avg	0.82	0.82	0.82	425,698

Table 10: XGBOOST results after parameter tuning

Metric	Value
Matthew's Correlation Coefficient	0.6603
Balanced Accuracy	0.6518
Average Precision Score (Macro)	0.7387
ROC AUC (One-vs-One)	0.9653
ROC AUC (One-vs-Rest)	0.9670

Table 11: Specific metrics for Advanced XGBOOST

Multi-label Model: Optimal LGBMClassifier Chain

Additionally, we propose another model that is able to classify the multiple interaction types among residues per a single protein. We implemented this Multi-label model in order to present a possible future idea to leverage, extend and improve to classify the contact types. Specifically this model is built using a Classifier Chain with a LightGBM (LGBMClassifier) as the base estimator. The Classifier Chain trains a sequence of binary classifiers, where each classifier's prediction is used as input for the next. Using LightGBM as the base allows for fast and efficient gradient boosting at each step. This model was chosen since the labels on which we are working one are not independent and leveraging their correlations can improve predictive performance.

Table 12: Multi-label results

Class	Precision	Recall	F1-score	Support
HBOND	0.82	0.71	0.76	206,155
IONIC	0.66	0.33	0.44	7,014
Missing	0.60	0.88	0.71	211,924
PICATION	0.35	0.08	0.13	1,707
PIHBOND	0.03	0.03	0.03	358
PIPISTACK	0.73	0.79	0.76	7,524
SSBOND	0.59	0.51	0.55	431
VDW	0.53	0.08	0.14	143,378
Micro Avg	0.67	0.61	0.64	578,491
Macro Avg	0.54	0.43	0.44	578,491
Weighted Avg	0.66	0.61	0.58	578,491
Samples Avg	0.67	0.64	0.65	578,491
Weighted F1-score: 0.5816				
Hamming Loss: 0.1002				
Subset Accuracy: 0.5764				
Micro F1-score: 0.6392				

Experimental Evironment

The training of the models was performed using the entire provided dataset on a cloud computing machine equipped with:

- L40S GPU
- 16 vCPUs
- 94 GB of RAM

PyMol tests

In this section, we evaluate the quality and reliability of the interaction types identified by RING by comparing them with those detected using PyMOL [Sch15]. Specifically, we checked if some of the misclassified predictions made by our model were simply due to evaluation errors made by our model or error noise input in the training set computed by RING.

Looking at the results 13, it appears that sometimes the feature '*c_dist*' has too much weight on contact prediction. Often, when the model predicts a missing contact, the value of *c_dist* is quite high. This may mean that in certain occasions the model is not able to fully infer relative positions and spatial distance information, and relies too much on the information given by *c_dist*. In fact, looking at residues that may have this issue using PyMol (row 1), they appear spatially not too far apart, and RING predictions appear to be correct.

The issue of our model not being able to correctly evaluate spatial and geometrical data is reinforced by the following observations: in some cases the sole presence of aromatic rings inside residues that may be able to generate π-bonds, makes the model predict that for sure there will be a π-bond of some kind (row 3,8). Again, using PyMOL, the residues appear to be too far apart or in relative positions that cannot facilitate a π-contact of any kind.

The same problem occurs for SS-bonds. It seems like whenever the two residues considered are CYS, the model predicts an SS-bond, even if the residues are very far apart. The distance at which an SS-bond can happen is ≤ 2.5 Å. On the tested couple (row 9), the distance between the two residues was above 4 Å, so the contact for sure is not possible.

Through our tests, we found a couple of residues (row 5), for which RING probably missed a π-H bond. This may mean that RING's predictions are not perfect, and possibly that some margin of our model's error is due to this.

Also, on a different couple of residues (row 7), our model was able to predict an ionic bond, while RING calculated a missing contact. In PyMOL, the two atoms that could be ionically bonded were 4.1 Å apart and in a geometrical position that favors a contact. Their distance is a little bit over RING's threshold for ionic bonds, that is 4 Å. So, between the two residues there may be a weak ionic contact, but it cannot be said for sure. It is interesting to note that our model was able to capture it, while RING missed it.

These tests were done on a very small number of contacts, so all these claims need to be further reinforced with deeper analysis and more data.

#	prediction	RING	PYMOL	pdb_id	s_resi	s_resn	t_resi	t_resn
1	Missing	HBOND	HBOND	6sun	57	THR	104	ALA
2	HBOND	HBOND, PICATION	HBOND, PICATION	6sun	64	TYR	154	ARG
3	PIPISTACK	HBOND	HBOND	6sun	206	HIS	243	TYR
4	IONIC	IONIC, HBOND	IONIC, HBOND	6sun	90	ARG	95	GLU
5	HBOND	PICATION, HBOND, VDW	HBOND, PICATION, PIHBOND	2r31	7	TRP	10	ARG
6	HBOND	Missing	Missing	2r31	105	ARG	116	ALA
7	IONIC	Missing	Missing/IONIC (weak)	1iq6	15_A	LYS	21_A	GLU
8	PIPISTACK	VDW	Missing/VDW	1iq6	47_A	PHE	88_A	PHE
9	SSBOND	VDW	Missing/VDW	1jm1	145	CYS	170	CYS
10	Missing	Missing	Missing	1jm2	52	GLY	211	VAL

Table 13: Interactions from: predictions, RING, and PyMOL across different PDB structures.

Conclusion

In this experiment, the progression from a basic Gaussian Naive Bayes model to an optimized XGBoost model demonstrates substantial improvements in predicting residue interaction types. Starting with the naive Bayes approach, the results were modest, with a Matthew's Correlation Coefficient (MCC) of 0.1148 and a balanced accuracy below 0.5, reflecting limited predictive power. Transitioning to XGBoost with duplicates removed by alphabetical ordering, boosted the MCC to 0.3780 but incorporating an interaction hierarchy based on interaction strength further enhanced performance, increasing MCC to 0.4573 and improving average precision substantially, indicating the model better captured meaningful interaction patterns. Moreover removing VDW interactions, which may introduce noise due to their weak and numerous nature, led to a marked jump in MCC (0.5833) and ROC AUC (above 0.95), suggesting that filtering less informative interaction types can refine model focus and accuracy. Finally, parameter tuning of the hierarchical XGBoost model without VDW interactions achieved the best results, with MCC rising to 0.6603 and balanced accuracy to 0.6518, and with a test accuracy of 0.82.

As a final and extra idea we proposed the Multi-label model which wasn't chosen as the final predictor due to its poor performance compared to Advanced XGBoost. But still with this model we tried to face the problem of classification of interactions in a different way proposing a possible future idea to expand and improve.

As an additional test we perform a reliability assessment on the type of contacts between residues identified by our model, RING and PyMol.

To conclude, we highlighted the fact that careful feature selection combined with hyperparameter optimization can unlock significant predictive gains, making the final model more accurate and robust for this classification task.

References

[Bal07] Robert L. Baldwin. Energetics of protein folding. *Journal of Molecular Biology*, 371(2):283–301, 2007.

[MVB+11] A. J. M. Martin, M. Vidotto, F. Boscariol, T. Di Domenico, I. Walsh, and S. C. E. Tosatto. Ring: networking interacting residues, evolutionary information and energetics in protein structures. *Bioinformatics*, 27(14):2003–2005, 2011.

[Sch15] Schrödinger, LLC. The pymol molecular graphics system, version 1.8, 2015. Available at https://pymol.org/.