Prediction of T-cell receptor specificity

Uchit Bhadauriya
Faculty of Engineering
University of Bristol
Bristol, United Kingdom
ir23063@bristol.ac.uk

Letian Zhang
Faculty of Engineering
University of Bristol
Bristol, United Kingdom
xq23145@bristol.ac.uk

Yu Gu
Faculty of Engineering
University of Bristol
Bristol, United Kingdom
iq23054@bristol.ac.uk

Lubin Wan
Faculty of Engineering
University of Bristol
Bristol, United Kingdom
dm23227@bristol.ac.uk

***Abstract*—T-cell is one of the most important immune system cells that recognize targets by specifically binding to receptors expressed on the cell membrane. T-cell receptor (TCR) is one of the most promising emerging therapies. The TCR gene library can reflect a person's health status. However, TCR varies greatly and its specificity is not easy to predict using traditional empirical methods. In order to use machine learning to predict TCRs that can bind to specific epitopes, we first analysed and preprocessed the data in the open source database VDJdb [1], then we created a distance/similarity matrix representation of TCR sequences based on a common method for predicting specificity from sequences proposed by Vujovic et al [2]., after that a two-dimensional map of the TCR was drawn and colored according to specificity. Then, the TCR sequences were clustered according to specificity using the DBSCAN algorithm, and the antigen specificity was predicted based on the sequences through the KNN algorithm. In the final prediction model, predictions can be made based on different species, and the prediction accuracy is high.**

I. Introduction

T-cell is one of the most important immune system cells and plays an important role in adaptive immunity. T-cell can specifically recognize TCR expressed on the cell membrane to identify and eliminate cells infected by viruses, bacteria or cancerous cells in the body, thereby preventing infection or further spread.

In this process, TCR plays a vital role. A large number of unique TCRs are generated under the interaction between pathogens and the immune system, and their diversity is greatly enhanced by the complex genetic mechanism of VDJ recombination. TCR is composed of α subunit and β subunit. Each α subunit contains a V and a J segment. Each β subunit contains a V, a D and a J segment, encoding the DNA of the TCR variable region. Formed by joining different gene segments to create numerous unique TCRs. At the same time, the process of VDJ recombination involves a mechanism of inserting and deleting nucleotides in a pseudo-random manner, thereby greatly increasing the diversity of the junction region (CDR3).

Although TCRs play a crucial role in the immune system, TCR diversity also poses significant challenges, which complicates the understanding of TCR specificity from a genomic perspective. Because TCR varies greatly, its specificity is difficult to predict using traditional empirical methods.

With the rise of machine learning methods, applying machine learning methods to TCR-specific prediction has become a feasible method. This project aims to use machine learning technology to analyse TCR sequences in the VDJdb database [1] and predict their specificity for specific epitopes. Based on the method of predicting specificity from sequence described by Vujovic et al. [2], a distance/similarity matrix of TCR sequences was built and used to train a model that can classify TCRs based on specificity. This method of applying machine learning to TCR specificity prediction can effectively improve the accuracy of predicting TCR specificity and may provide reference ideas for other immunotherapies.

II. Literature Review

The study of TCR specificity is crucial in the field of biomedical. Its main applications include predicting the development trend of diseases and monitoring the effects of treatment. By analysing TCR sequences, scientists can predict its response to specific antigens and then develop targeted treatment plans. Therefore, clustering TCRs based on sequence characteristics to reveal their biological similarities and antigen specificities has become an important task [2].

The use of antigen-rich TCR libraries to form meta-clonotypes is a new clustering strategy that groups TCRs based on biochemical similarity, allowing researchers to identify functionally similar TCRs from complex TCR libraries. In practical applications, this strategy has been used to analyse TCR data of COVID-19 patients. After research and analysis, multiple public meta-clonotypes were discovered, which have shown extremely high application value in identifying and quantifying TCRs with similar functions. However, converting TCR sequence data into actual medical information still faces many challenges such as large data volume and high processing complexity [2].

High-throughput sequencing and single-cell sequencing technologies have transformed the understanding of TCR repertoires, allowing scientists to precisely pair the alpha and beta chains of TCRs. By correlating TCR sequences with specific cell phenotypes through single-cell RNA sequencing technology, we can learn more about individual TCR lineage characteristics and obtain key information needed for personalized treatment. This has already been applied in the field of cancer treatment [3].

GIANA is an efficient TCR analysis tool that can significantly improve the speed and accuracy of TCR sequence clustering, allowing researchers to obtain research results in a shorter time. By introducing efficient data structures, GIANA is able to quickly manage and index a large number of TCR sequences and is particularly suitable for processing huge TCR data sets, thereby effectively supporting in-depth research on complex immune responses. However, the performance of GIANA is highly related to parameters. Inappropriate parameters may lead to inaccurate results, which will increase the complexity of the model [4].

GLIPH2 can efficiently process a large number of TCR sequences. In the study of Mycobacterium tuberculosis (Mtb)-specific T cell targeting epitopes, the researchers used artificial antigen presenting cell (aAPC) technology to conduct 3724 different genes. Studies have found that at least five Pro-Pro-Glu (PPE) proteins are recognized by T cells as targets of Mtb. This experimental result provides new insights into how T cells recognize and respond to Mtb [5].

III. Methodology

This project aims to use machine learning technology to analyse TCR sequences in the VDJdb database and predict their specificity for specific epitopes. This method of applying machine learning to TCR specificity prediction can effectively improve the accuracy of predicting TCR specificity and may provide reference ideas for other immunotherapies.

A. Data processing approach

First, we converted the vdjdb.txt file into a vdjdb.csv file to facilitate subsequent data processing. Then we process the data with a confidence score of 0 to reduce the impact of fuzzy data on the results. After that, we renamed the data selected after analysis to simplify subsequent data processing. Finally, the α and β chains of mouse and human species were processed according to categories, and data lines with missing information were deleted to ensure the accuracy of the analysis process.

B. Tool selection

To achieve the research objectives, this study evaluated multiple analysis methods and selected the tcrdist algorithm. tcrdist is a tool specially designed for TCR sequence analysis and can effectively measure the distance between different TCR sequences. Compared with the other two tools, tcrdist has higher interpretability, and although the calculation amount is large, the model structure is simple, making it easy to maintain and debug. In comparison, although the GIANA algorithm can also quickly process large-scale data sets after training, its training process will take up more time and resources.

$$\text{TCRDist}(s,t) = \sqrt{\sum_{i=1}^{n}(f_i(s) - f_i(t))^2} \tag{1}$$

By using a similarity matrix to define and quantify these distances, subtle differences between TCR sequences can be effectively captured. These distance matrices provide a solid foundation for subsequent cluster analysis and machine learning modeling.

TABLE I: TCR sequence analysis comparison

Method/Tool	TCRDist	GLIPH	GIANA
Interpretability	High, easily interpretable through pairwise distances.	Medium, based on motif identification and complex statistical analysis.	Low, as it uses deep learning models which are typically harder to interpret.
Model Complexity	Medium, based on the calculation of pairwise distances which is straightforward but can be computationally intensive.	High, involves complex pattern recognition and statistical analysis.	High, deep learning models are inherently complex.
Run Time	Medium, the calculation of pairwise distances can be time-consuming.	Slow, due to extensive computation required for motif identification and pattern analysis.	Fast, as deep learning models can quickly process large datasets after training.
Typical Use	Broadly used across different types of TCR sequence analysis.	Specifically used for identifying and analysing motifs within TCR sequences.	Used for high-throughput TCR sequence analysis with deep learning for enhanced prediction accuracy.

C. Algorithm application

After establishing the distance matrix, apply a variety of machine learning techniques to classify and predict TCR sequences:

- K-Nearest Neighbors (KNN) and Random Forest: These algorithms are used to predict the specificity of TCRs and can help to better understand the association of TCRs with specific antigens.
- Unsupervised clustering (DBSCAN): Using the DBSCAN algorithm, TCR sequences are clustered to identify functionally similar TCR clusters. The clustering results

are evaluated through Silhouette Score to ensure the effectiveness of the clustering.

- Dimensionality reduction technology (UMAP and MDS): To better visualize and analyse data, UMAP and MDS technology are used to reduce the dimensionality of the distance matrix and graphically display the clustering results of different TCRs.

TABLE II: Specificity prediction algorithm considerations

Method/Feature	**K-Nearest Neighbors (KNN)**	**Random Forest**
Interpretability	High, the classification of new samples is easily understandable.	Low, as the classification process and results are generally more complex to understand.
Prediction Accuracy	High, but the choice of k can greatly influence the result.	High, and generally provides a good performance on various datasets.
Model Complexity	Low, model complexity is relatively low.	High, as random forests can become quite complex depending on the number and depth of the trees.
Data Dependencies	Sensitive to outliers and the scale of the data.	Robust to outliers and generally performs well with various data distributions.
Training Speed	Fast, as it simply relies on distance calculations.	Slow, as constructing multiple trees can be computationally expensive.
Usability in Large Datasets	Inefficient for large datasets due to the need for distance calculation between points.	Highly effective for large datasets, capable of handling them efficiently.

D. Innovation

Multi-dimensional data integration and analysis:

- Effectively combines α chain and β chain TCR data to form a multidimensional analysis method to analyse TCR data from multiple angles (α chain, β chain, α+β combination).

Cross-species TCR feature identification:

- Analysed data include mouse and human data that directly address cross-species comparisons of TCR signatures. This helps understand how TCR repertoires react in different organisms, potentially identifying universal or species-specific patterns.

CR predictive modeling based on machine learning:

- Clustering using DBSCAN and Silhouette Score to assess clustering quality is a form of unsupervised machine learning. Additionally, K-nearest neighbor (KNN) classification that predicts TCR classes based on a similarity matrix provides a predictive modeling approach. This setup facilitates the development of models that predict TCR responses based on distance matrices.

Despite the challenges, TCR sequence analysis has great potential for application in the field of immunotherapy through evolving technologies and innovative strategies. Future research should continue to optimize these technologies and expand their application in clinical diagnosis and treatment.

IV. Data Description / Preparation

We used the 'vdjdb.csv' dataset for the project. This is a dataset with over 90,000 rows of data, which includes a lot of data such as genes, CDR3 sequences, and species. To make the data clearer, we took only those columns that are important to our project like complex.id, gene, cdr3, v.segm, j.segm, species, antigen.epitope, antigen.gene, and vdjdb.score.

- The column 'complex.id' contains a number for all T-cell rows that are in the dataset. For data with complex.id equal to 0, which means we only have α or β single chain data. For data with complex.id not equal to 0, we can obtain one-to-one corresponding α and β chain data. Therefore, in data preprocessing, we classified the data based on whether complex.id is equal to 0.
- The column 'gene' contains one of 'TRA' or 'TRB', indicating that the data in this row belongs to an α or β chain.
- The column 'cdr3' records the sequence information of cdr3, which is very important for us to recognize antigen specificity.
- The column 'v.segm' stands for the Variable segment of the TCR gene. This segment is important for TCR to recognize antigens presented by the major histocompatibility complex (MHC) on antigen-presenting cells [6].
- The column 'j.segm' stands for the Joining segment of the TCR gene. It is also an important part of the TCR structure.
- The column 'species' represents the species type corresponding to this gene data. Our project mainly analyses gene data of humans and mice.
- The column 'antigen.epitope' represents epitope information of antigens, which can activate T-cells and provoke a response of the immune system [7].
- The column 'antigen.gene' refers to the genes of the antigen, which is crucial for predicting antigen specificity.
- The column 'vdjdb.score' refers to the credibility of our gene data.

For our project, we only retained the necessary columns mentioned above in the dataset for data analysis and processing. We removed all data with vdjdb.score equal to 0 to ensure the reliability and credibility of the analysed data and the conclusions drawn [8]. Due to the need to use the TCRdist3 package to calculate the distance matrix, we made some changes to the structure of the raw dataset and renamed some columns to ensure the correspondence between the dataset and the reference dataset, and to ensure that the distance matrix

can be calculated correctly. In addition, we discarded unrecognized gene data from our dataset based on the reference dataset 'alphabeta_gammadelta_db.tsv' used to calculate the distance matrix using the TCRrep method [9]. Afterwards, we discarded the missing values in the dataset. The processed dataset is shown in Table. III-Table. V. Table. III shows the dataset which is mainly used to calculate the distance matrix of all α chains. Table. IV shows the dataset which is mainly used to calculate the distance matrix of all β chains. Table. V shows the dataset which is mainly used to calculate the distance matrix of all combined α and β chains, and is used for subsequent clustering, prediction, and other tasks.

TABLE III: df1_alpha dataset

Column	Explanation
clone_id	complex.id
cdr3_a_aa	cdr3
v_a_gene	v.segm
j_a_gene	j.segm
subject	species
antigen.epitope	antigen.epitope
epitope	antigen.gene
vdjdb.score	vdjdb.score

TABLE IV: df1_beta dataset

Column	Explanation
clone_id	complex.id
cdr3_b_aa	cdr3
v_b_gene	v.segm
j_b_gene	j.segm
subject	species
antigen.epitope	antigen.epitope
epitope	antigen.gene
vdjdb.score	vdjdb.score

TABLE V: df1 dataset

Column	Explanation
clone_id	complex.id
cdr3_a_aa	cdr3 with α
cdr3_b_aa	cdr3 with β
v_a_gene	v.segm with α
v_b_gene	v.segm with β
j_a_gene	j.segm with α
j_b_gene	j.segm with β
subject	species
antigen.epitope	antigen.epitope
epitope	antigen.gene
vdjdb.score	vdjdb.score

V. Results and Discussion

A. Limitations and solutions

To predict the T-cell receptor specificity in our project, the first method we considered was one-hot encoding. It is a very common way to turn amino acid sequences of TCR into digits. However, after subsequent assessments, it quickly becomes apparent that this leads to multiple problems.

We encountered several problems or drawbacks of one-hot encoding in TCR sequences, particularly when we considered the CDR3 region. The CDR3 region shows significant variation in sequence length. Consequently, we have incompatibilities between the one hot encoding and the actual nature of sequence representation. In the one-hot encoding, all the inputs were supposed to be uniform in length. CDR3 regions in TCRs show a high degree of variability in their length, even spanning hundreds of amino acids. We believe that using one-hot encoding on CDR3 sequences might lead to models that are either inefficient or fail to capture crucial biological variations. This could ultimately make the predictive models less accurate and less effective.

To address these challenges, we proposed a simple alternative approach to the representation of TCR sequences. To better handle variable-length sequences in our models, we can adopt embedding-based methods similar to the successful approaches used in natural language processing. In these methods, we convert each amino acid sequence into a compact vector. This vector representation captures both the positional context and the overarching global context of the entire sequence and gives a more nuanced and effective alternative to the traditional one-hot encoding method. Moreover, by leveraging modern sequence processing architectures, such as recurrent neural networks (RNNs) and transformer-based models which are specifically designed to handle variable-length sequences, our model can enhance its ability to match patterns and identify dependencies within TCR sequences. This approach allows us to sophisticated analysis and understanding of the sequence elements. This approach enables the model to simultaneously address the variability in input sequence length naturally and also potentially capture the functional aspects of sequences better that's why it improves the prediction task of TCR specificity.

By implementing and evaluating these strategies or you can say as per the knowledge related to this project which we aquired by reading the various research papers related to our project, we can anticipate a significant improvement in the predictive accuracy of our model and also we can feel a more generic model that can work seamlessly across a wide range of TCR sequences. Overall, this refinement will contribute to our understanding of TCR variability and expand our knowledge of T-cell specificity.

B. Compute distance matrices

We used the TCRdist3 Python package to compute the distance matrices. It is a very useful package for us to calculate TCR gene sequences related data. We selected data from human and mouse species from the dataset for separate calculations. We used the TCRrep method to calculate the distance matrices of the α and β chains of mouse and human species using the processed df1_alpha and df1_beta datasets, respectively. Then, we used the processed df1 dataset to calculate the distance matrices of combined α and β chains for paired chains of mouse and human species, respectively. The calculation results of these distance matrices are shown below. To avoid duplication, we only display the distance matrices of mouse here.

$$\begin{bmatrix} 0 & 0 & 126 & \cdots & 151 & 156 & 151 \\ 0 & 0 & 126 & \cdots & 151 & 156 & 151 \\ 126 & 126 & 0 & \cdots & 117 & 138 & 150 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 151 & 151 & 117 & \cdots & 0 & 121 & 128 \\ 156 & 156 & 138 & \cdots & 121 & 0 & 112 \\ 151 & 151 & 150 & \cdots & 128 & 112 & 0 \end{bmatrix}$$

distance matrix – all α chains (mouse)

$$\begin{bmatrix} 0 & 132 & 133 & \cdots & 172 & 148 & 151 \\ 132 & 0 & 162 & \cdots & 195 & 147 & 173 \\ 133 & 162 & 0 & \cdots & 179 & 155 & 135 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 172 & 195 & 179 & \cdots & 0 & 105 & 137 \\ 148 & 147 & 155 & \cdots & 105 & 0 & 137 \\ 151 & 173 & 135 & \cdots & 137 & 137 & 0 \end{bmatrix}$$

distance matrix – all β chains (mouse)

$$\begin{bmatrix} 0 & 247 & 244 & \cdots & 251 & 266 & 260 \\ 247 & 0 & 111 & \cdots & 282 & 299 & 368 \\ 244 & 111 & 0 & \cdots & 276 & 292 & 340 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 251 & 282 & 276 & \cdots & 0 & 216 & 297 \\ 266 & 299 & 292 & \cdots & 216 & 0 & 192 \\ 260 & 368 & 340 & \cdots & 297 & 192 & 0 \end{bmatrix}$$

distance matrix – paired combined α and β chains (mouse)

These distance matrices can effectively quantify the degree of difference between different TCR sequences, for example, the data at positions (a,b) in the matrix represents the difference between the ath and bth TCR sequences. When calculating the distance matrices of the combined α and β chains, we used to add the distance matrices of the α and β chains to obtain the result, because the distance matrices of the α and β chains have the same dimension and consistent row order, and the addition can to some extent reflect the differences in the combined chain TCR sequences [10]. Afterwards, the distance matrix of the combined α and β chains calculated here will play an important role in tasks such as antigen specific clustering and prediction.

We saved the distance matrices of the combined α and β chains of human and mouse species in CSV format, named 'tr_human_alpha_beta.csv' and 'tr_mouse_alpha_beta.csv', respectively. In addition, we saved the data that matches the reference dataset in the TCRrep method as CSV files named 'tr_human_alpha_beta_color.csv' and 'tr_mouse_alpha_beta_color.csv', respectively. These saved files allow us to observe the calculation results more clearly and facilitate the use of this data for subsequent tasks.

In general, in this section, we completed the calculation and analysis of the distance matrices for α and β single chains, as well as the combined α and β chains. But we should also pay attention to the shortcomings in our work. Due to the limited species types and data volume in the dataset, we only used data from humans and mice for calculation, but did not use more data from other species.

C. Color based on specificity

As part of our study, dimensionality reduction solutions help us describe these TCRs in a 2-D format that compares similarities and specificities. We densely constructed plots using the Uniform Manifold Approximation and Projection (UMAP) technique for human data based on both α and β chains and those of mouse data for α and β together [11].

The UMAP graphs clearly showed the way a TCR clustered accordingly towards the antigen specificity. We utilized a color that coded differently to separate all epitopes. Regarding mouse TCRs, visualizations of both α and β chains appeared to have distinct grouping which denotes that TCR molecules of the same epitope affinity direct toward each cluster. This finding was further enhanced by the combination of the α and β chains that showed the role of chain interaction in the peak discrimination capacity of TCR against various antigens [11].

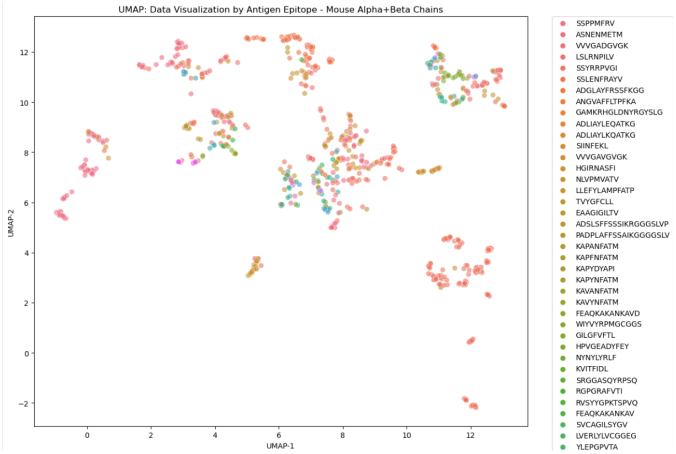

Fig. 1: UMAP-mouse

Similarly, human TCR visualizations demonstrated clustering based on TCR specificity, emphasizing the differences between various epitopes due to the distinct nature of interactions between the TCR and the antigen. These relationships were further highlighted by the combined α-β chain plots, which demonstrate the high contribution of chain interaction to TCR specificity [11].

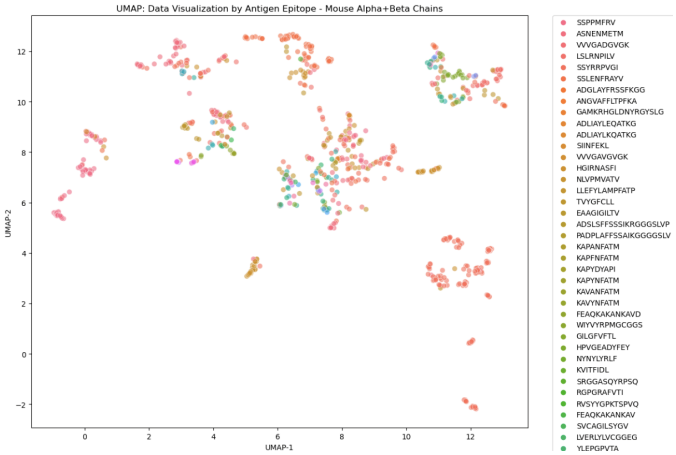

Fig. 2: UMAP-human

These visual representations of TCR repertoire are therefore key in allowing us to understand complex aspects of structure and selectivity. These constantly observed outcomes confirm

the idea of TCRs' fine-tuning to epitopes. The visualizations from such experiments give the models used in formulating computational algorithms that are crucial in the process of making predictions of the final TCR specificity. Such models make it possible to tailor the systems of T-cell-based immunity therapies [11].

D. Cluster based on specificity

In this part, we clustered TCRs based on specificity. We clustered the distance matrices of combined α and β chains for paired chains of mouse and human species. We tried Kmeans clustering, Kmediods clustering and DBSCAN clustering [12] respectively. Compared with the other two methods, the DBSCAN clustering algorithm had better performance. In the final cluster visualization diagram, there were more obvious clusters. Therefore, we used the DBSCAN clustering algorithm for clustering. In addition, for visualization, we used the MDS multidimensional scaling technique to reduce the distance matrix to a two-dimensional space. Finally, we removed the noise points, calculated the silhouette coefficients of the distance matrices of combined α and β chains for paired chains of mouse and human species, and performed visual representation. The calculated silhouette coefficients and the effect of visual clustering are shown in the Fig.3 and Fig.4.

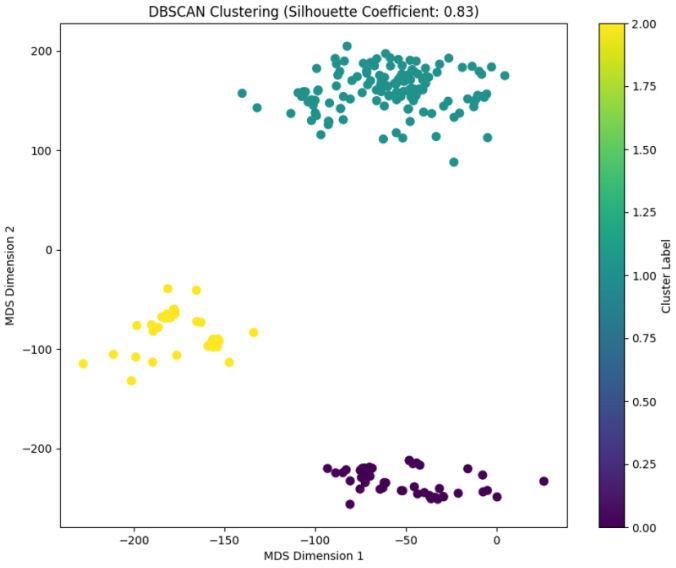

Fig. 3: cluster-mouse

As an index used to evaluate the quality of clustering results, the silhouette coefficient takes into account the cohesion and separation of the sample points. The cohesion reflects the closeness of the sample points to the elements in the cluster, while the separation reflects the degree of isolation of the sample points from elements outside the cluster. Silhouette coefficient takes these two points into consideration. We can see that in the mouse data set, the silhouette coefficient is 0.83, while in the human data set, the silhouette coefficient is 0.60, which shows that TCRs with similar specificities can be effectively clustered. And the difference in silhouette

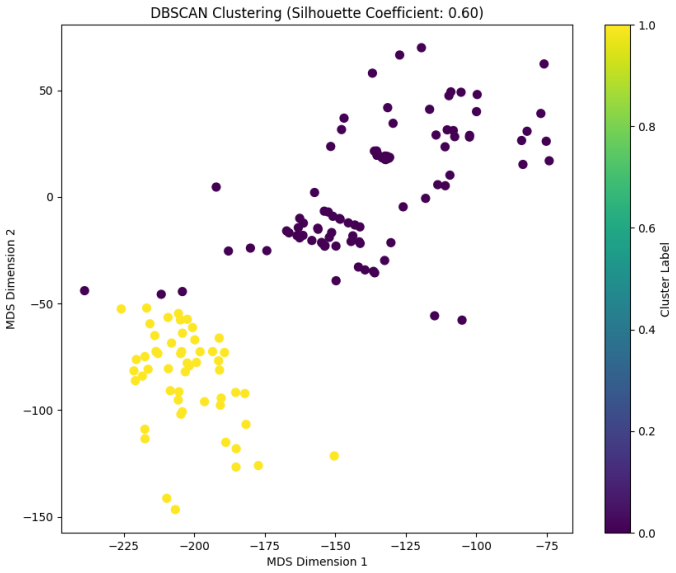

Fig. 4: cluster-human

coefficients occurs because the human data set has more data and the data is messier, so the clustering effect is not as obvious as the mouse data set.

The clustering effect shows that TCRs with similar specificities may have common structural characteristics, causing them to cluster in the feature space. At the same time, we also need to further consider how to reduce the impact of noise points.

E. Predict antigen specificity

This part aims to predict the epitope fragment of an antibody using tcr sequence. Due to its simplicity and interpretability, the K-nearest neighbor (KNN) classification method is a good choice. This algorithm was chosen to handle the prediction task because it is good at classifying data points according to the most common categories among their nearest neighbors. Especially when the distance measure between samples can well reflect the similarity between samples, KNN performs better [13]. Random forest is also a good method. Compared with KNN, its prediction accuracy is higher, but the relative task will become more complex and the running time will be longer.

The data uses a similarity matrix between humans and mice. This step greatly reduces the amount and complexity of data, and effectively improves the calculation efficiency and the interpretability of the model. The distribution of sample counts between different antigen classes revealed heavy tails, with many classes having small numbers of samples, which makes training robust models challenging. Data visualization using histograms provides a good representation of the sample number distribution of antigens in human and mouse data sets. Setting the filtering threshold between 90 and 95 percent (25 for human data and 40 for mouse data) can make the data more representative and effectively alleviate the problem of uneven data. Considering that mice have fewer data samples, more data samples are retained for them during filtering.After

processing the data, perform feature standardization. This step is very important for the KNN algorithm. It ensures that different features have the same weight in the model, and then converts the antigen name into a numerical code for model training.

The KNN algorithm is applied to predict humans and mice respectively. The final output results show that the prediction accuracy of humans is 0.66 and the precision is 0.7. The mouse's accuracy was 0.8, and the accuracy and other indicators also reached 0.8. Next, a confusion matrix is drawn to further visualize the output results. For humans, the model can recognize and classify correctly to a certain extent, but there are many wrong categories. The number of wrong categories for mice was significantly reduced. Only the recognition effect in column 5 was not good, and the correct classification was basically achieved. Through visual observation, it is obvious that the model training effect of mice is better than that of humans. This may be due to the fact that human TCRs are more complex and diverse, whereas mice are generally less heritable. In addition, data processing and filtering may have an impact on the final training results. The predicted results are shown in the Table.VI, Fig.5 and Fig.6.

TABLE VI: Performance Metrics

Species	Accuracy	Precision	Recall	F1 Score
Human	0.66	0.70	0.66	0.66
Mouse	0.80	0.81	0.80	0.79

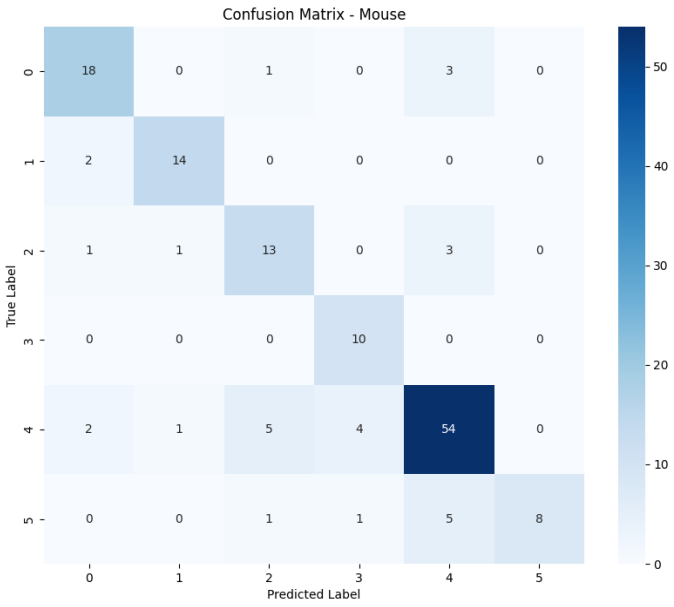

Fig. 5: predict-mouse

It is essential to consider that the KNN algorithm's performance is highly dependent on the chosen hyperparameters, distance metrics, and the scale of the features. Although KNN is intuitive, it might not capture the complexities of antigen specificity, which can involve non-linear and high - order interactions. Advanced models, such as deep learning

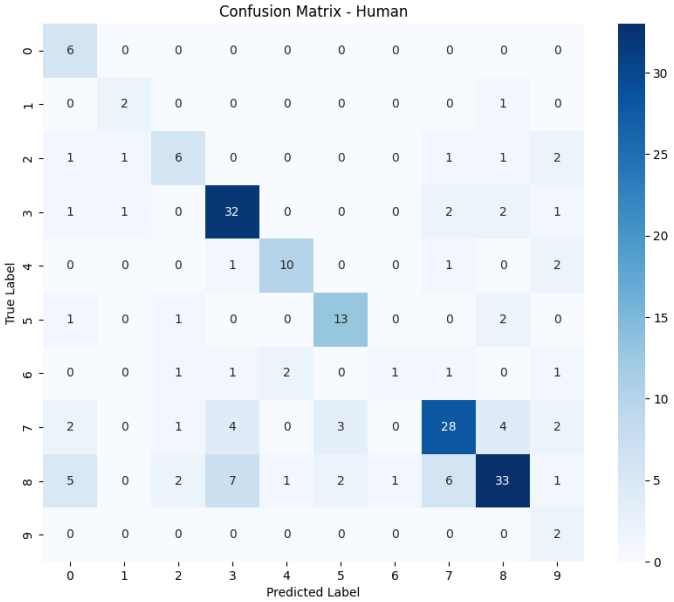

Fig. 6: predict-human

approaches, could potentially unveil these intricate patterns, provided sufficient data quality and quantity.

In general, KNN can initially achieve task prediction. Random forest has more advantages in prediction accuracy, but its efficiency is slower, so knn may be a more suitable choice. The next step can be to consider applying feature engineering for more detailed data processing. In addition, deep learning is also a direction worth exploring, which can further improve the accuracy of the model.

VI. Further Work and Improvement

Based on completing the above work, we hope to continue working in the future to further improve and perfect this project.

There are still some shortcomings in certain aspects of the project, which is also the direction of our future efforts. This project only clusters and predicts the genes of humans and mice. In the future, efforts can be made to collect gene data from more species and predict their antigen specificity. Besides, we only had V and J gene segments data from the dataset for analysis, but did not have D segments data. In the future, we will try to include D segments as part of our analysis and prediction. In addition, only the combined α and β chains were used for clustering and prediction, the potential impact of α and β single chains on clustering and prediction was not considered. The future research on the impact of α and β single chains on clustering and prediction results is very meaningful. Apart from that, we should consider more about how to reduce the number of noise points to make clustering more general. What's more, due to limitations in computer resources, this project used the KNN algorithm to predict adversarial specificity, which may result in the prediction accuracy, recall, and f1 score not reaching the highest level. In the future, we will try to obtain better prediction results

by using Amazon Web Services to run other algorithms, such as the neural network algorithm, which is also the work we have started. Further adjusting the parameters of clustering and prediction algorithms in the project to obtain better results is also one of our efforts. We have used some methods to obtain better parameters, but more methods can be adopted for validation.

VII. Conclusion

This report mainly introduces the background, objectives, methodology, data preparation, and results of the TCR specificity prediction task, and provides prospects for further work and improvement.

This project aims to use machine learning to predict TCR specificity. We first preprocessed the dataset and selected humans and mice as our research subjects, respectively. Then, the distance matrices of the α chains, β chains, and the combined α and β chains were calculated using the TCRrep method. Afterward, we used UMAP dimensionality reduction methods to plot and color TCR according to their specificity. In addition, we also used the DBSCAN method to cluster TCR based on specificity and visualized the results. Finally, we used the KNN algorithm to predict antigenic specificity and calculated the accuracy, recall, and f1 score of the prediction results.

This report analyses the results obtained from the project after completing these studies. We believe that overall, the project achieved the expected results, but there are still some shortcomings in certain aspects. For example, more data from different species can be used to analyse their antigen specificity separately, further research can be conducted on the potential impact of α and β single chains on clustering and prediction results, and other algorithms can be used to improve prediction accuracy. These are the goals for our future work and improvement.

In summary, this is a very meaningful and challenging project. This project refers to some existing research results and strives to try and innovate multiple algorithms to complete the work and achieve the best results on this basis. With the continuous progress of future work, we hope that this project can achieve more valuable results.

The code we wrote is available at https://github.com/UoB-DSMP-2023-24/dsmp-2024-group09.git.

References

[1] M. Goncharov, D. Bagaev, D. Shcherbinin, I. Zvyagin, D. Bolotin, P. G. Thomas, A. A. Minervina, M. V. Pogorelyy, K. Ladell, J. E. McLaren *et al.*, "Vdjdb in the pandemic era: a compendium of t cell receptors specific for sars-cov-2," *Nature methods*, vol. 19, no. 9, pp. 1017–1019, 2022.

[2] M. Vujovic, K. F. Degn, F. I. Marin, A.-L. Schaap-Johansen, B. Chain, T. L. Andresen, J. Kaplinsky, and P. Marcatili, "T cell receptor sequence clustering and antigen specificity," *Computational and Structural Biotechnology Journal*, vol. 18, pp. 2166–2173, 2020.

[3] K. Mayer-Blackwell, S. Schattgen, L. Cohen-Lavi, J. C. Crawford, A. Souquette, J. A. Gaevert, T. Hertz, P. G. Thomas, P. Bradley, and A. Fiore-Gartland, "Tcr meta-clonotypes for biomarker discovery with tcrdist3: identification of public, hla-restricted sars-cov-2 associated tcr features," *BioRxiv*, 2021.

[4] H. Huang, C. Wang, F. Rubelt, T. J. Scriba, and M. M. Davis, "Analyzing the mycobacterium tuberculosis immune response by t-cell receptor clustering with gliph2 and genome-wide antigen screening," *Nature biotechnology*, vol. 38, no. 10, pp. 1194–1202, 2020.

[5] H. Zhang, X. Zhan, and B. Li, "Giana allows computationally-efficient tcr clustering and multi-disease repertoire classification by isometric transformation," *Nature communications*, vol. 12, no. 1, p. 4699, 2021.

[6] M. Shugay, D. V. Bagaev, I. V. Zvyagin, R. M. Vroomans, J. C. Crawford, G. Dolton, E. A. Komech, A. L. Sycheva, A. E. Koneva, E. S. Egorov *et al.*, "Vdjdb: a curated database of t-cell receptor sequences with known antigen specificity," *Nucleic acids research*, vol. 46, no. D1, pp. D419–D427, 2018.

[7] E. Jokinen, A. Dumitrescu, J. Huuhtanen, V. Gligorijević, S. Mustjoki, R. Bonneau, M. Heinonen, and H. Lähdesmäki, "Tcrconv: predicting recognition between t cell receptors and epitopes using contextualized motifs," *Bioinformatics*, vol. 39, no. 1, p. btac788, 2023.

[8] M. Shugay, D. V. Bagaev, I. V. Zvyagin, R. M. Vroomans, J. C. Crawford, G. Dolton, E. A. Komech, A. L. Sycheva, A. E. Koneva, E. S. Egorov *et al.*, "Vdjdb: a curated database of t-cell receptor sequences with known antigen specificity," *Nucleic acids research*, vol. 46, no. D1, pp. D419–D427, 2018.

[9] P. B. Koshlan, Mayer-Blackwell and A. Fiore-Gartland, "Tcr distances," 2020, https://tcrdist3.readthedocs.io/en/latest/tcrdistances.html, Last accessed on 2024-4-30.

[10] K. Mayer-Blackwell, A. Fiore-Gartland, and P. G. Thomas, "Flexible distance-based tcr analysis in python with tcrdist3," in *T-Cell Repertoire Characterization*. Springer, 2022, pp. 309–366.

[11] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell, "Dimensionality reduction for visualizing single-cell data using umap," *Nature biotechnology*, vol. 37, no. 1, pp. 38–44, 2019.

[12] D. Deng, "Dbscan clustering algorithm based on density," in *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, 2020, pp. 949–953.

[13] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of k-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning," *Decision Analytics Journal*, vol. 3, p. 100071, 2022.