Briefings in Bioinformatics, 2025, **26(1)**, bbaf030
https://doi.org/10.1093/bib/bbaf030
Problem Solving Protocol

A comprehensive benchmarking for evaluating TCR embeddings in modeling TCR-epitope interactions

Xikang Feng[1,*,‡], Miaozhe Huo[2,‡], He Li[1], Yongze Yang[2], Yuepeng Jiang[2], Liang He[1], Shuai Cheng Li[2,*]

[1]School of Software, Northwestern Polytechnical University, 127 West Youyi Road, Beilin District, Xi'an Shaanxi, 710072, China
[2]Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, 999077, China

[*]Corresponding authors. Xikang Feng, E-mail: fxk@nwpu.edu.cn; or Shuai Cheng Li, E-mail: shuaicli@cityu.edu.hk
[‡]The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

Abstract

The complexity of T cell receptor (TCR) sequences, particularly within the complementarity-determining region 3 (CDR3), requires efficient embedding methods for applying machine learning to immunology. While various TCR CDR3 embedding strategies have been proposed, the absence of their systematic evaluations created perplexity in the community. Here, we extracted CDR3 embedding models from 19 existing methods and benchmarked these models with four curated datasets by accessing their impact on the performance of TCR downstream tasks, including TCR-epitope binding affinity prediction, epitope-specific TCR identification, TCR clustering, and visualization analysis. We assessed these models utilizing eight downstream classifiers and five downstream clustering methods, with the performance measured by a diverse range of metrics for precision, robustness, and usability. Overall, handcrafted embeddings outperformed data-driven ones in modeling TCR-epitope interactions. To further refine our comparative findings, we developed an all-in-one TCR CDR3 embedding package comprising all evaluated embedding models. This package will assist users in easily selecting suitable embedding models for their data.

Keywords: TCR-epitope interaction; benchmarking TCR CDR3 encoding; biological relevance of embeddings; data-driven and handcrafted embeddings

Introduction

T cells, as pivotal mediators of the immune response, recognize and bind antigens through T cell receptors (TCRs). The complementarity determining region 3 (CDR3) of TCRβ chains encapsulates significant immunological diversity [1, 2]. These TCR sequences not only record an individual's immunological history but also aid in developing diagnostic tools by providing insights into immune dynamics [3–7].

The advances in machine learning have propelled the study of TCRs forward, particularly through the analysis of CDR3 sequences. For instance, Beshnova *et al.* employed a convolutional neural network (CNN) model that learned the patterns within CDR3 sequences to establish a relationship between TCR repertoires and cancer, culminating in a cancer indicator tool named DeepCAT [8]. Similarly, other computational approaches like DeepTCR and TEINet have targeted the CDR3 region to model TCRs for various immune-related predictive tasks [9–12]. These approaches transform CDR3 sequences into high-dimensional embeddings to capture essential computational features.

Encoding strategies range from handcrafted feature extraction, which utilizes amino acid characteristics or BLOSUM matrix similarity scores guided by expert-devised rules [13, 14], to data-driven methods that learn directly from sequence data [10, 15]. Although various methods claim superiority, the absence of a comprehensive benchmark hinders our ability to evaluate their effectiveness objectively, leaving significant gaps in understanding the advantages of each approach [16].

The design of CDR3 encoding methods must strive for a balance between biological relevance, robustness, and usage cost [17]. Therefore, their evaluation should cover assessments in these areas to determine each method's efficacy. Biological relevance ensures that the encoded CDR3s authentically represent immunological functions. An effective embedding should accurately cluster CDR3s with akin functions and structures, mirroring the biological principle of antigen recognition. Robustness requires encoding methods to perform well across different downstream tasks, be compatible with varied machine-learning models, and withstand diverse parameter settings. Furthermore, computational efficiency and user-friendliness are practical considerations that reflect usage costs. By addressing these criteria, benchmarks can help identify encoding techniques that are both scientifically insightful and pragmatically viable, providing a solid foundation for advancing immunological predictions.

In this study, we benchmark a range of CDR3 encoding methods and establish a comprehensive benchmark for their systematic evaluation. Our framework evaluates these methods based on biological relevance, robustness, and usage cost, providing a critical assessment tool for CDR3 encoding in immunological research. We test these methods through a series of tasks, including generic

Received: May 27, 2024. **Revised:** October 17, 2024. **Accepted:** January 13, 2025

TCR-epitope binding prediction, epitope-specific TCR identification, and TCR clustering. Extensive testing is conducted using a variety of algorithms, metrics, and parameters. We also employ visual analysis to examine the spatial distribution of embeddings generated by different methods. Additionally, we evaluate computational demands and ease of use to determine the practicality of each method. The conclusions drawn from our benchmarks help researchers select suitable encoding methods and standardize the evaluation process for future TCR encoding models. Moreover, we have integrated all methods involved in this project into an all-in-one Python package to simplify usage.

Materials and methods

Preprocessing of datasets

In this project, we utilized multiple data sources, carrying out further processing to refine the datasets for our analyses. Our filtering strategy included removing low-quality CDR3 sequences, specifically those containing stop codons, or with lengths shorter than 10 or longer than 30 amino acids, and excluding epitope sequences longer than 20 amino acids. We retained amino acid sequences that begin with cysteine (C) and end with phenylalanine (F), while excluding any sequences containing non-standard amino acids. Additionally, we excluded entries lacking epitope or antigen labels. Details of the data sources and the extra processing steps applied to the datasets are described in the Supplementary Notes 1.5.

Implementation of embedding methods

We employed 19 tools to assess a variety of CDR3 embedding strategies, extracting embedding components directly from the source codes of each tool. For handcrafted methods, we used the default parameter settings provided by the developers, adjusting parameters only when necessary. For data-driven approaches, we retrained TCRpeg, catELMo, DeepTCR, pMTnet, and Word2Vec using a designated retraining dataset, as detailed in Supplementary Notes 1.5.5 and Supplementary Fig. S20. Specifically for TCRanno, to avoid data leakage and ensure fairness, we retrained it using a hold-out dataset separate from the clustering dataset, since its embeddings are derived from a supervised learning classifier. Detailed implementation specifics and parameter settings for each method are documented in Supplementary Notes 1.3.

Design for TCR-epitope affinity prediction task

The prediction of TCR-epitope binding affinity is approached as a binary classification task, where pairs (t, e), consisting of a TCR sequence t and an epitope e, are labeled as non-binding (0) or binding (1). The negative pairs are generated using random recombination strategy [18]. The task aims to develop classifiers that can accurately predict the binding affinity $\hat{y}$ for novel TCR-epitope pairs. Training and validation data are divided using the three data-splitting strategies mentioned in the following section, and the models' generalization capabilities are evaluated using an independent test set.

We evaluated the performance of embedding methods for TCR-epitope affinity prediction under two scenarios. The first, referred to as the *Generic TCR-epitope binding affinity prediction*, involved testing various downstream classifiers on a dataset containing all epitopes. Epitopes were uniformly encoded using the BLOSUM62 encoder, and TCR sequences were encoded using different CDR3 encoding methods. For ImRex, we employed its unique encoding scheme for TCR-epitope pairs. In this scenario, we tested CDR3 encoding 18 methods, excluding TCRanno. For the second scenario, *Epitope-specific TCR identification*, we extracted epitope-specific subsets from the complete epitope binding dataset based on the most dominant epitopes present. Six subsets, each containing only one type of epitope, were formed to evaluate whether classifiers could distinguish between binding TCRs and non-binding TCRs. All methods except TCRanno were tested in this task.

Data splitting strategies for TCR-epitope affinity prediction task

We employed three dataset-splitting strategies to evaluate the models' ability to handle unseen data. The dataset was first divided using a random division approach, termed the *Random Split*, with a training-to-validation ratio of 9:1. Additionally, to assess the embedding models' performance on unseen data, we implemented two further data partitioning strategies as introduced by the ATM-TCR project [18]: *TCR Split* and *Epitope Split*.

TCR Split: The dataset was divided such that all TCR sequences in the validation set were distinct from those in the training set, testing the models' extrapolation to new TCR sequences.

Epitope Split: This strategy ensured that epitope sequences in the validation set were not included in the training set, allowing evaluation of the models' performance on previously unseen epitopes.

Classifiers for generic binding prediction and TCR identification tasks

We employed a suite of eight classifiers to evaluate the performance of various TCR embedding methods. These classifiers include gradient boosting (GB), random forest (RF), decision tree (DT), k-nearest neighbors (KNN), multilayer perceptron (MLP), recurrent neural network (RNN), CNN, and multi-head self-attention model to assess the performance of various TCR embedding methods. The detailed parameter settings for each method are recorded in Supplementary Notes.

Clustering algorithms for TCR clustering tasks

In assessing the performance of various TCR embedding methods, we applied five clustering algorithms along with four distinct metrics to explore their stability and adaptability. These algorithms included hierarchical clustering, k-means clustering, spectral clustering, affinity propagation clustering, and mean-shift clustering, all implemented via the scikit-learn [19] Python package. We examined 18 different methods, excluding ImRex, which necessitates the encoding of TCR-epitope pairs. To ensure compatibility with the clustering algorithms, we transformed the high-dimensional embeddings of individual CDR3 sequences into flattened one-dimensional feature vectors. The detailed parameter setting for each algorithm is recorded in Supplementary Notes.

UMAP visualization

For visual analysis, we utilized the UMAP package [20] to achieve a low-dimensional representation of TCR sequence embeddings. We performed comparisons across various methods, excluding ImRex which requires encoding TCR-epitope pairs and hence contains epitope label information within the embeddings. Among the evaluated methods, the supervised deep learning model TCRanno was specifically trained on separate TCR-epitope paired data to avoid data leakage in this visualization [21]. Methods that transform data into vectors of shape (B, L, K)—where B represents the dataset size, L the CDR3 length, and K the dimension parameter—are first reshaped to $(B, L \times K)$ before dimensionality reduction.

UMAP reduced the dimensionality of each CDR3 embedding to two dimensions, with points colored according to their epitope labels. To ensure clarity, we filtered out CDR3 sequences that recognize multiple epitopes according to the IEDB database, retaining only those with a single functional specificity. This filtering ensured that each CDR3 had a unique label, and each label represented a specific function. Under these conditions, clear category boundaries indicate a strong embedding-function relevance.

Benchmark metrics

Predictive performance metrics included the area under the ROC curve (AUC), accuracy, sensitivity, specificity, and F1 score. Metrics for quantifying clustering performance include the adjusted Rand index (ARI), normalized mutual information (NMI), Purity, and $F_{purity>0.9}$. The detailed formula for these metrics is described in the Supplementary Notes.

Computational resource

All methods were evaluated on a server equipped with dual Intel Xeon Silver 4216 CPUs (total 32 cores running at 2.10 GHz) and 125 GB of system memory. Graphics processing was performed using a single Nvidia RTX 3090 GPU featuring 24 GB of onboard memory. The operating system was CentOS Linux release 7.9.2009.

Results

Benchmarking pipeline

Our benchmark focused on the CDR3 regions of TCRβ chains, which are key specificity determinants in antigen recognition. We reimplemented the CDR3 encoding modules following the provided instructions in the 18 recent studies: DeepRC [22], ImRex [23], Luu *et al.* [24], Word2Vec [25], SETE [26], TCRGP [27], ATM-TCR [18], NetTCR2.0 [28], iSMART [29], GIANA [14], TITAN [30], ERGO-II [31], DeepTCR [9], pMTnet [15], catELMo [32], clusTCR [13], TCRpeg [12], and TCRanno [21]. We extracted only the portions related to encoding TCR CDR3, disregarding other aspects of the original frameworks. Hereafter, the name of each project is used to specifically denote its corresponding CDR3 encoding method. Additionally, we included a universal protein model, ESM [33, 34], as a baseline for comparison.

For some methods whose TCR encoders are based on models pre-trained on extensive data, specifically TCRpeg, catELMo, DeepTCR, pMTnet, and Word2Vec, we retrained these models using a uniform dataset to ensure a fair comparison and prevent potential information leakage in downstream experiments (see Materials and methods section). The benchmarking was performed on these retrained models.

These encoding methods can be categorized into several distinct groups. When considering the dependency on data for feature learning, they can be classified into *Handcrafted* and *Data-driven* strategies. From the perspective of the encoding strategy, they can be further divided into *One-hot encoding-based, Physicochemical property-based, BLOSUM-based*, and *Deep learning-Based methods*. The characteristics of these 19 encoding methods were summarized in Fig. 1A, Table 1 and Supplementary Fig. S1. Further details regarding the above-mentioned encoding strategy were introduced in the Supplementary Notes.

Our benchmarking pipeline evaluates the performance of various CDR3 encoding strategies on three downstream experimental tasks, including the generic TCR-epitope binding affinity prediction, epitope-specific TCR identification, and TCR clustering (Fig. 1B). In the TCR-epitope binding affinity prediction and TCR identification tasks, eight predictive models were employed: GB, RF, DT, KNN, MLP, RNN, CNN, and multi-head self-attention model. In TCR clustering, five clustering methods were utilized: mean shift, k-means, affinity propagation, hierarchical, and spectral clustering. Additionally, we evaluate the usability performance of these methods based on their computation time, memory requirements, installation dependencies, and code quality. Lastly, the visualization analysis was designed to assess how well the encoding methods facilitate interpreting TCR data in the 2D space, a critical aspect for intuitive data exploration.

To undertake the benckmarking, we utilize two primary data sources. The first, employed for binding affinity prediction task, epitope-specific identification task, and UMAP visualization task, is a TCR-epitope interaction dataset merged from IEDB, VDJDB, and McPAS [36–38]. For the visualization dataset, we select TCRs from the IEDB database that uniquely pair with a single epitope clonotype, employing these epitopes as labels for the TCRs. The second, used for the clustering analysis, consists of CDR3s linked to various antigens, as collected in the GIANA project (Fig. 1C). The datasets underwent additional quality control measures, as detailed in the Materials and methods section. Ultimately, the TCR-epitope binding dataset encompassed 276 057 pairs, evenly divided between positive and negative samples. The clustering dataset consisted of 9033 TCR sequences associated with 25 different antigens, guaranteeing at least 100 sequences per antigen cluster. In the visualization dataset, there were 638 unique CDR3-epitope pairs, with each CDR3 clonotype paired with a unique epitope label.

After obtaining results from all 19 methods across all datasets, we assessed the methods' performances from multiple perspectives. This evaluation was structured around three core aspects: predictive performance, clustering performance, and usability metrics (Fig. 1D). Predictive performance metrics included the AUC, accuracy, sensitivity, specificity, and F1 score. Metrics for quantifying clustering performance include the ARI, NMI, Purity, and $F_{purity>0.9}$. Usability metrics focused on computational time, memory requirements, installation dependencies, and code quality, which are crucial for practical adoption and integration into existing workflows. The visualization analysis was assessed based on the method's ability to encode TCRs with similar functions into proximal regions while maintaining high resolution in the embedding space. Detailed definitions and rationale for each metric are provided in the Materials and methods section.

Based on our experimental results, we present a table (Fig. 2) that captures the performance of all encoding methods across the three domains.

Generic TCR-epitope binding affinity prediction task

We approached the prediction of TCR-epitope binding affinity as a binary classification task, where pairs of data consisting of a TCR sequence and an epitope are classified as binding or non-binding.

Classifier robustness to unseen data was evaluated using three dataset-splitting strategies: random split, TCR split, and epitope split. The random split assesses the model's predictive ability across randomly chosen TCR-epitope pairings. The TCR split measures model success in predicting new, unseen TCR sequences, and the epitope split determines predictability concerning completely novel epitopes. Figure 3, Supplementary Figs S2–S9, and Supplementary Table 1 display a side-by-side evaluation of different TCR CDR3 embedding and classification techniques under these dataset partitioning strategies. We further utilize an independent test set from the pMTnet project to validate the

Table 1. Summary of TCR embedding tools

Method	Encoding strategy	Encoding principle	Embedding shape	Feature in the last dim
ImRex	Handcrafted	Selected physicochemical properties	(B, L, 20, 5)	Handcrafted physical and chemical properties
Luu et al.	Handcrafted	Atchley factor + mask indicator	(B, L, 6)	Five atchley factors with a mask indicator
clusTCR	Handcrafted	Selected physicochemical properties	(B, L × properties_num)	User-selected physical and chemical properties
ATM-TCR	Handcrafted	BLOSUM45	(B, L, 25)	BLOSUM matrix vector
NetTCR2.0	Handcrafted	BLOSUM50	(B, L, 20)	BLOSUM matrix vector
TITAN	Handcrafted	BLOSUM62	(B, L, 20)	BLOSUM matrix vector
TCRGP	Handcrafted	BLOSUM62 + alignment + matrix transformation	(B, L × 21)	Transformed BLOSUM feature
GIANA	Handcrafted	BLOSUM62 + matrix transformation	(B, 96)	Transformed BLOSUM feature
iSMART	Handcrafted	BLOSUM62 + MDS	(B, 96)	Sequence isometric coordinates
SETE	Handcrafted	K-mers + PCA	(B, K)	Dimensionality-reduced k-mer feature
DeepRC	Handcrafted	One-hot	(B, L, 23)	Animal acid type with position information
DeepTCR	Data-driven	Autoencoder	(B, K), defalt K=256	Deep learning representation
pMTnet	Data-driven	Autoencoder	(B, K), defalt K=30	Deep learning representation
ERGO-II	Data-driven	Autoencoder	(B, K), defalt K=100	Deep learning representation
TCRanno	Data-driven	MLP	(B, K), defalt K=32	Deep learning representation
TCRpeg	Data-driven	LSTM	(B, K), defalt K=192	Deep learning representation
catELMo	Data-driven	ELMo-based architecture [35]	(B, K), defalt K=1024	Deep learning representation
Word2Vec	Data-driven	K-mers + deep neural networks	(B, K), defalt K=16	Sum of learned k-mer features
ESM	Data-driven	Transformer + structure information	(B, K), defalt K=1280	Deep learning representation

Note: This table showcases all of the embedding tools evaluated in this project. The “Embedding Shape” column defines the dimensions of the embeddings. “B” represents the number of input sequences or the batch size. “L” is the sequence length, which can be the actual length, the maximum length, or a predetermined fixed length. “K” denotes the embedding dimensionality controlled by hyperparameter; the table provides the default values. “properties_num” refers to the number of user-selected features, with a default of four. The Word2Vec encoder is part of the immuneML package [25]. Among the data-driven methods, TCRanno is trained with supervised learning on antigen-label data, while other methods are trained with unsupervised learning using the architecture shown in the table.

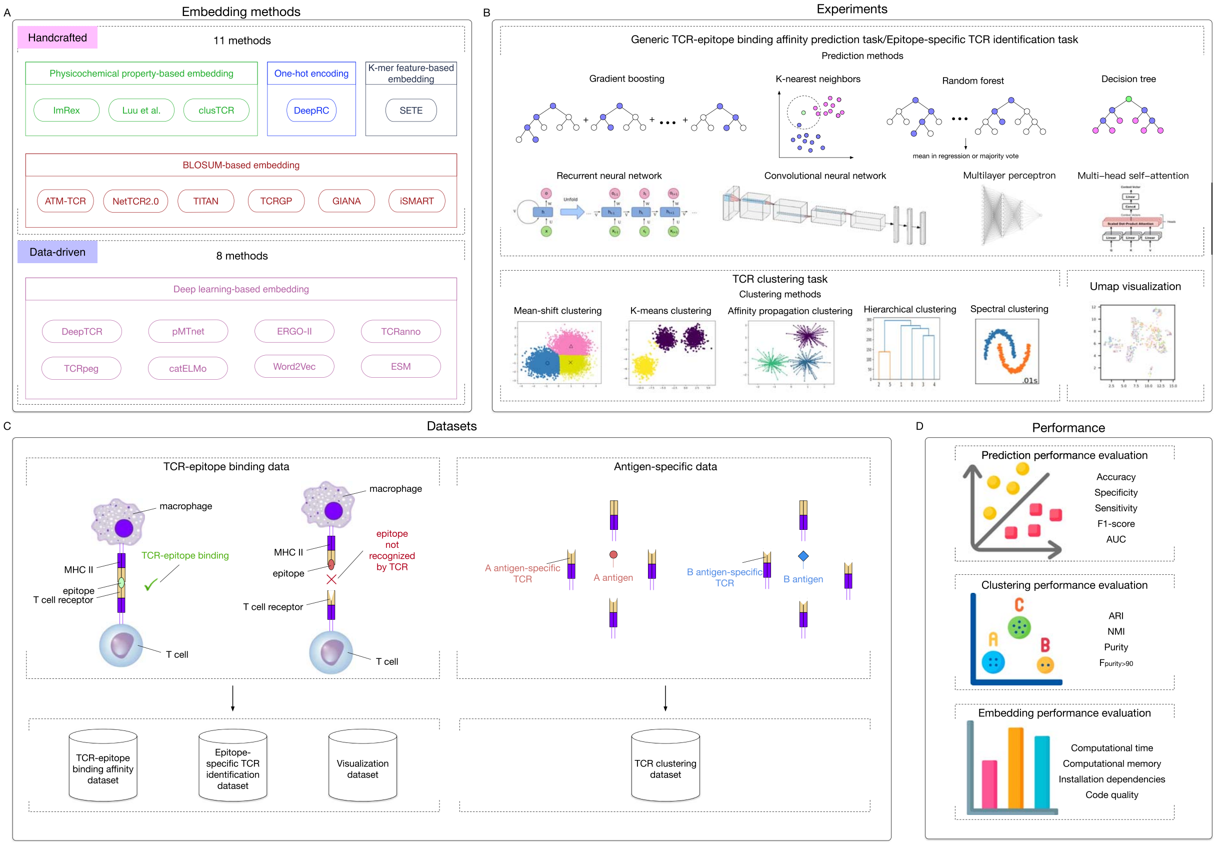

Figure 1. Overview of the benchmarking pipeline. (A) Eighteen tools plus one universal protein model (ESM) are assessed. These tools are categorized according to the design principles of the encoder. (B) These embeddings are then utilized in four experimental tasks designed to assess the impact of encoding strategies on TCR epitope specificity: generic TCR-epitope binding affinity prediction, epitope-specific TCR identification, TCR clustering, and TCR visualization analysis. Various predictive and clustering methods are applied within each task to ensure the robustness of our assessment. (C) The TCR-epitope dataset for the binding and identification tasks consists of pairs from IEDB, VDJDB, and McPAS. The visualization dataset is a curated selection from IEDB. The clustering dataset comprises antigen-specific TCR sequences from the GIANA project [14]. (D) Benchmarking results are quantitatively scored across three dimensions: predictive performance, clustering performance, and usability metrics.

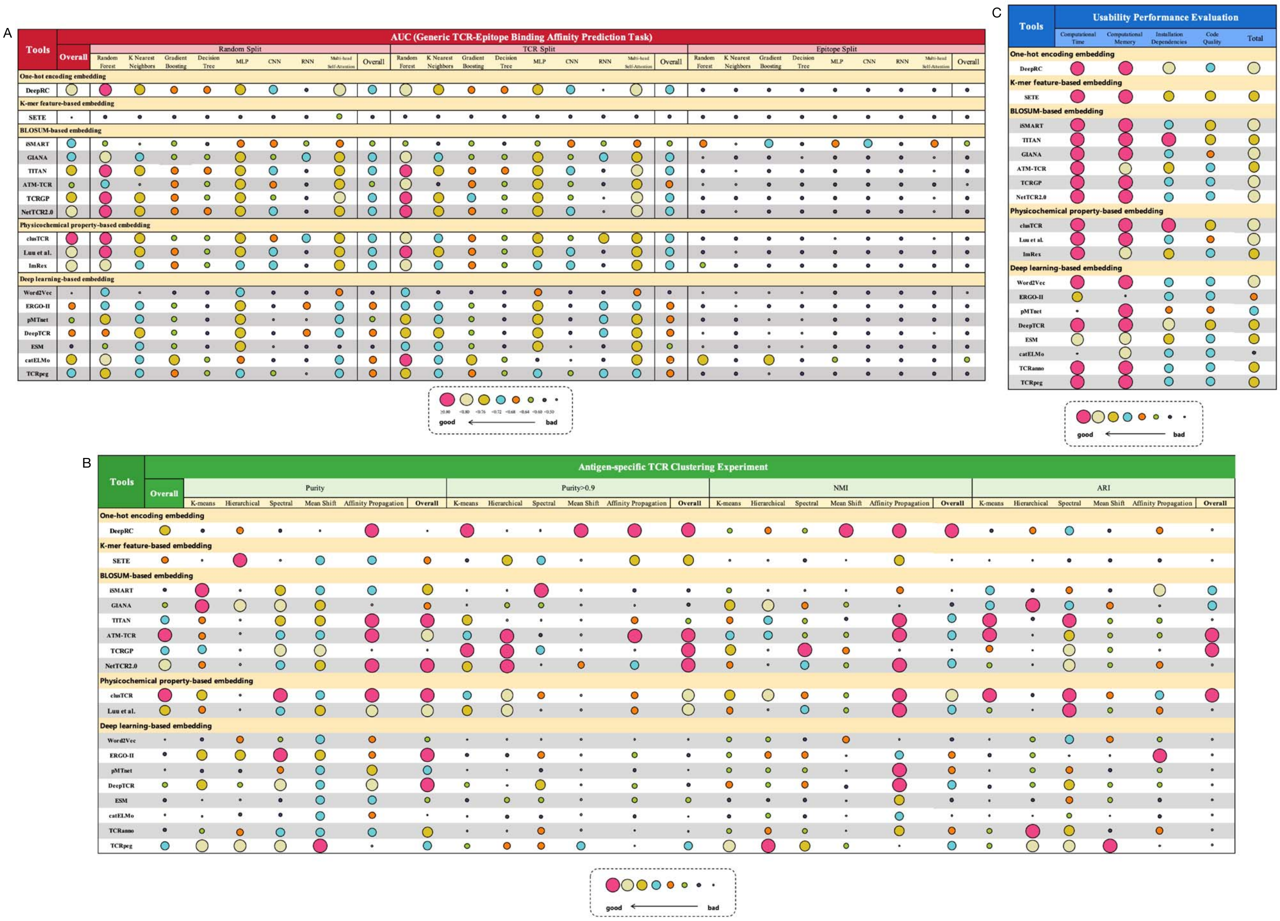

Figure 2. The summary table outlines the performance of all embedding methods in experiments related to generic TCR-epitope binding affinity prediction task, TCR clustering, and usability performance. (A) For TCR-epitope binding affinity prediction experiments, the AUC metric was employed, with results categorized by dataset split strategies and prediction classifiers. The source data are in Supplementary Table 1. (B) In the antigen-specific TCR clustering experiment, the results were assessed using a range of downstream clustering methods and diverse evaluation metrics. The source data are provided in Supplementary Table 2. (C) Assessment of usability performance based on computational time, memory requirements, installation dependencies, and code quality, resulting in an overall score reflecting practical deployment considerations for each method.

generalization capability of these embedding methods. The results are presented in Supplementary Fig. S10.

The results revealed variance in classifier performance when employing different splitting strategies, with no singular approach uniformly superior. When evaluating the random and TCR split strategies, which assess the model's predictive ability within a known epitope range, clusTCR—a physicochemical property-based embedding—demonstrated the best overall performance. It achieved the highest average AUC scores across all eight classifiers, recording mean AUCs of 0.707 and 0.705, respectively. Following DeepRC, based on the one-hot encoding method, is the second-best performing method, achieving the second-highest average AUC of 0.697 under both split strategies. From the perspective of embedding strategies, the performance results favor BLOSUM-based embedding methods (TCRGP, NetTCR2.0, TITAN), which maintained high performance across both random and TCR split strategies, consistently achieving some of the highest or second-highest AUC scores across eight classifiers. This robustness reflects the ability of BLOSUM-based methods to capture essential features for TCR-epitope binding, enabling accurate predictions even for TCR sequences not present during training.

In contrast, SETE, a k-mer feature-based embedding method, demonstrated the poorest predictive performance under both split strategies, with average AUCs of 0.556 and 0.552, respectively. Similarly, other embedding methods that also rely on k-mer features, such as Word2Vec, demonstrated the second least effective performance in the random split, suggesting intrinsic limitations in their ability to capture the complexities required for TCR-epitope binding prediction.

In terms of classifiers, RF, MLP, and the multi-head self-attention model all produced good predictions for all embedding data, with average AUCs exceeding 0.7 under both split strategies. However, the prediction performance of more complex models like CNN and RNN was relatively poor. This indicates that different classifiers have a significant impact on the results of TCR-epitope binding affinity prediction experiments.

When employing the epitope split strategy, which excludes epitopes in the validation set from the training set, the purpose is to test a classifier's ability to discern binding relationships with unseen epitopes. All embedding methods across different classifiers yielded AUC scores near the threshold of 0.5, which is equivalent to random guessing. This uniformly poor performance suggests that all the embeddings fail to enable classifiers to capture the general molecular binding characteristics necessary for the effective recognition of unseen epitopes. The ability to predict TCR recognition of novel epitopes is of substantial biological significance, as it mirrors the adaptive immune system's capacity to respond to new pathogens. However, our results reveal the general

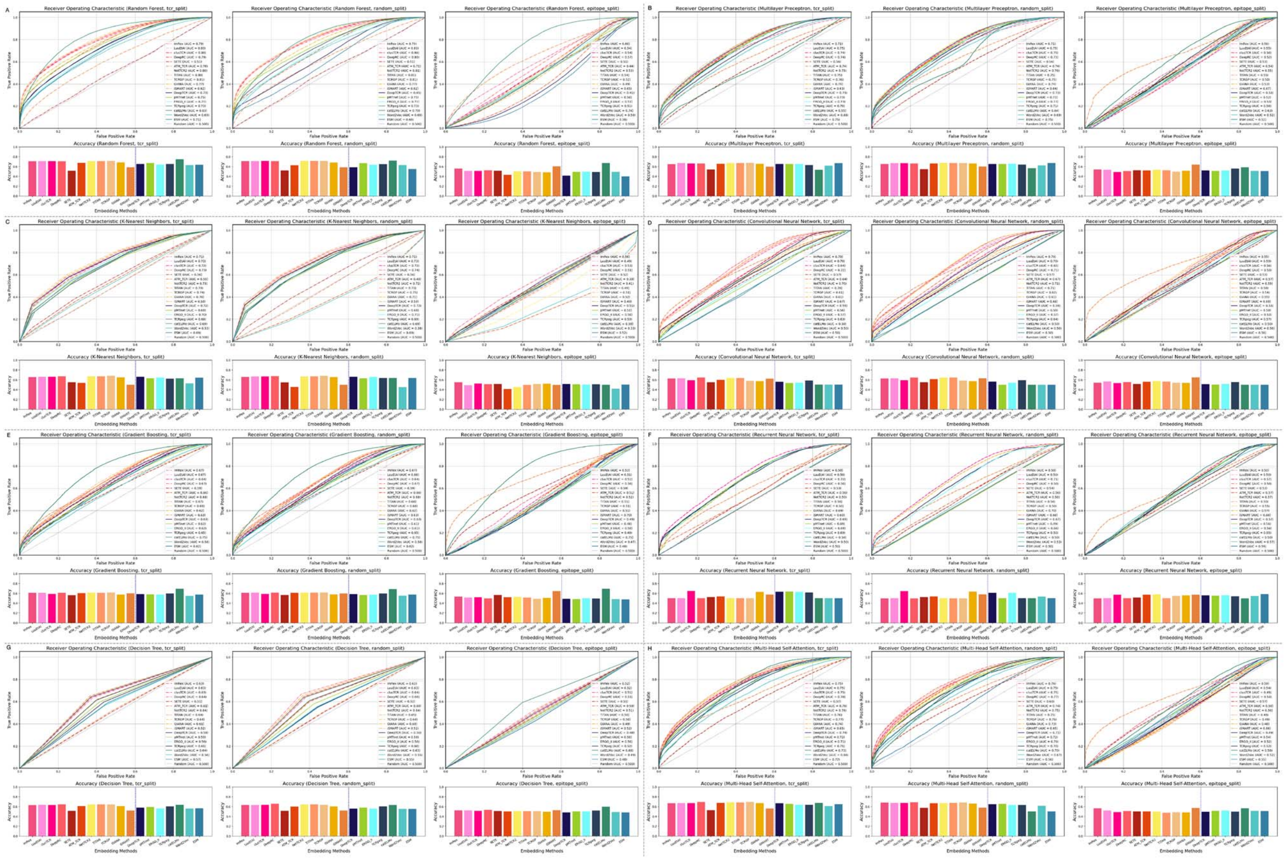

Figure 3. Evaluation of CDR3 embedding strategies across GB, RF, DT, KNN, MLP, RNN, CNN, and self-attention classifiers for TCR-epitope binding affinity classification task. Performance is assessed using the AUC and accuracy metrics under three dataset split strategies: random, TCR-centric, and epitope-centric splits. Overall, classifiers exhibit variable performance across different splits, with random splits generally yielding better results and epitope-centric splits showing notably poorer performance, in some cases falling below the AUC threshold of 0.5. Within the same split strategy, the impact of the chosen classifier on the performance of embedding methods is discernible.

deficiency of existing methods in this scenario (Supplementary Notes 1.10 and Supplementary Fig. S22), indicating the need to develop more effective encoding strategies to address this limitation.

Epitope-specific TCR identification task

In the epitope-specific TCR identification task, we selected subsets of the six most dominant epitopes. For each subset, TCRs binding to a specific epitope were labeled as positive samples, and TCR classification was employed to evaluate the performance of various encoding methods. The objective was to assess the discriminative capability of these methods within datasets defined by single epitopes. The performance measured by multiple metrics, including AUC, showed significant variation across these epitope-specific subsets (see Fig.4, and Supplementary Figs S11–S18).

Overall, no single encoding method demonstrated consistent superiority across all conditions; different methods excelled in different scenarios. Handcrafted-based and data-driven methods generally performed well, indicating their strengths in various classification tasks. SETE performed poorly on traditional classifiers. According to the results in Supplementary Figs S11–S14, SETE achieved AUC and accuracy values close to 0.5 on DT-based classifiers and KNN classifiers under our experimental conditions. On classifiers based on neural networks, ImRex often underperformed compared to other methods. For instance, in the MLP classifier, as shown in Supplementary Fig. S15, other tested methods achieved AUC values greater than 0.9 across all epitope-specific subsets, whereas ImRex only achieved AUC values ranging from 0.62 to 0.89.

TCR clustering task

In our exploration of TCR clustering, we evaluated the efficacy of multiple encoding methods in grouping antigen-labeled CDR3 sequences. Overall, methods employing a handcrafted encoding strategy outperformed those utilizing a data-driven encoding strategy in clustering experiments. Specifically within the handcrafted encoding strategy, both BLOSUM-based and physicochemical property-based embedding methods consistently exhibited superior performance compared to the other three categories across various metrics (Figs 2, 5, Supplementary Fig. S19 and Supplementary Table 2). Despite this trend, the results varied across different evaluation metrics, with no single encoding method consistently leading.

ATM-TCR, TCRGP, and NetTCR2.0, employing BLOSUM-based strategies, were particularly effective, attaining high scores in multiple evaluation metrics. iSMART, another method based on BLOSUM62 encoding, showed subpar performance compared to ATM-TCR, suggesting that the particular handcraft matrix transformations applied in ATM-TCR are more conducive to clustering TCR sequences. Method clusTCR [13], employing predefined physicochemical properties like amino acid mutation

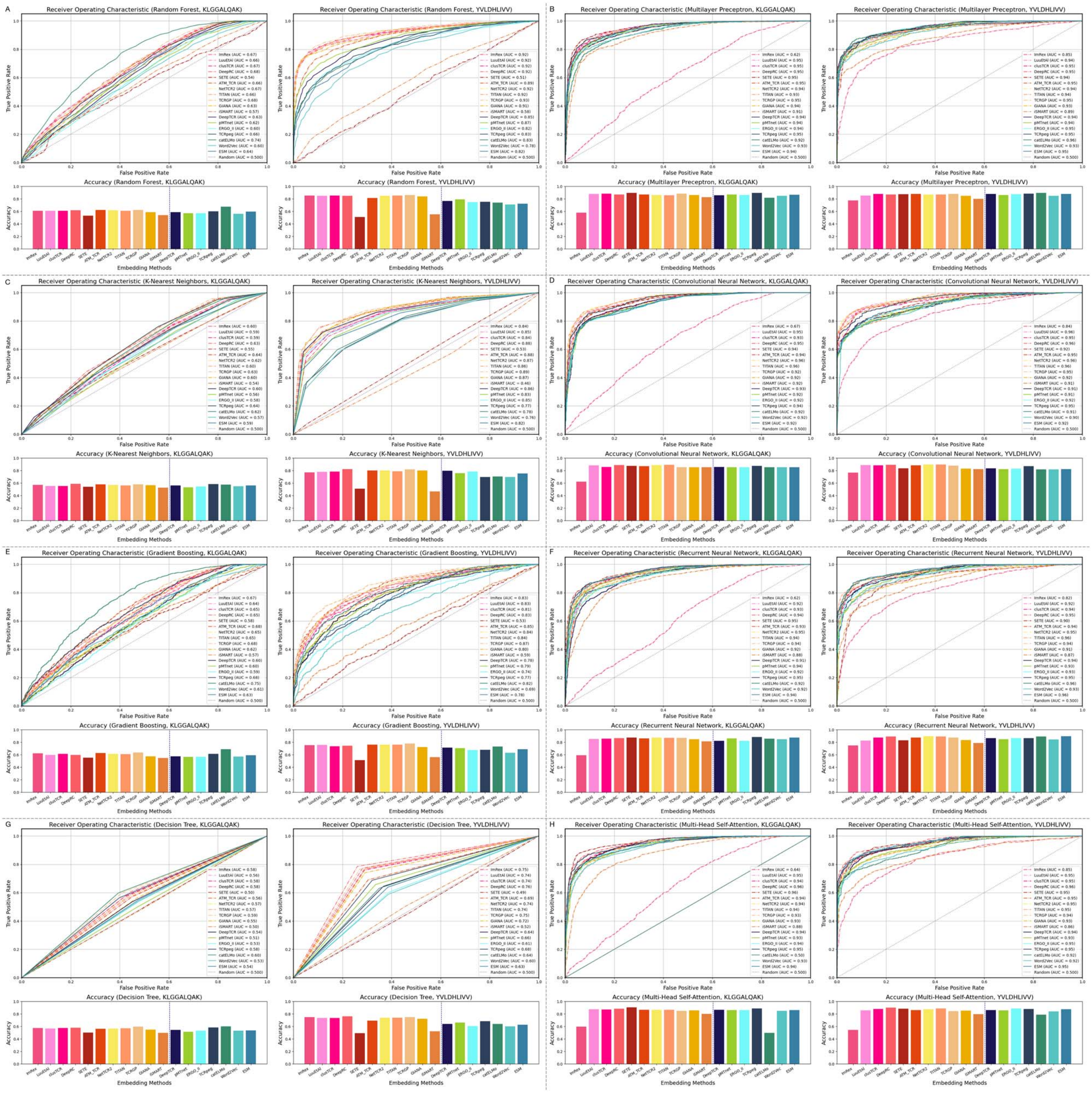

Figure 4. Comparative analysis of CDR3 embedding approaches on TCR-KLGGALQAK and TCR-YVLDHLIVV binding subset datasets using multiple classifiers. The performance of 18 embedding strategies was evaluated via TCR-epitope binding affinity prediction task on two distinct epitope-specific subsets. The AUC metric was utilized to measure classifier efficacy, highlighting the differential impact of embedding methods on predicting binding affinity within these epitope-focused datasets.

stability and hydrophobicity, also showed strong performance, underscoring the effectiveness of these selected features in encoding CDR3 regions. Luu *et al.*'s method [24], based on five-dimensional Atchley factors with a mask indicator, was effective but less effective than clusTCR. This observation suggests that while the Atchley factors successfully capture the physicochemical properties of amino acids for encoding TCR CDR3 regions, integrating additional selected properties could potentially enhance their functional encoding.

Data-driven methods did not perform as well as expected, often matching or underperforming compared to the baseline method ESM. This could be due to the typically short length of CDR3 data or suboptimal model architectures, suggesting that the existing deep learning models might not effectively capture the underlying functional information within the TCR sequences. However, TCRpeg distinguished itself within this category, achieving the highest scores in 7 metrics and the second-highest in 2 of the 12 evaluated metrics. This superior performance suggests that the pre-training process employed in the TCRpeg project effectively captured the complexities of TCR data, leading to a more accurate and robust embedding method.

In summary, while certain encoding methods showed promise, the overall performance across the encoding strategies was moderate. The average ARI, NMI, purity, and $F_{purity>0.9}$ values indicate no universally superior embedding method for CDR3 clustering tasks.

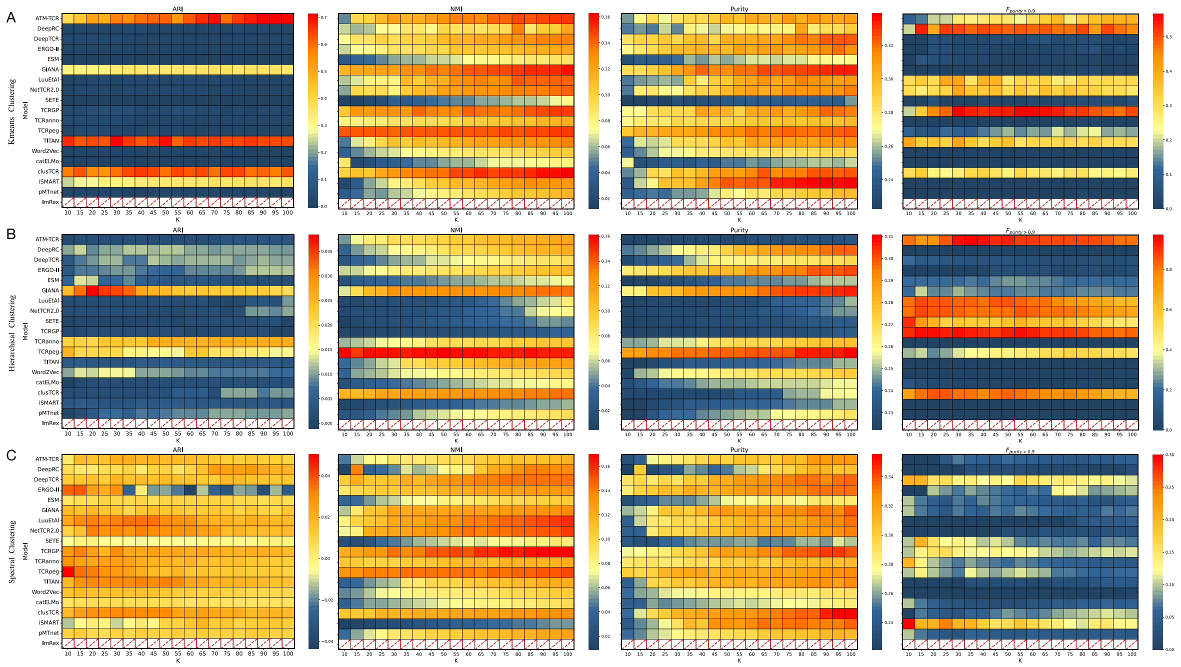

Figure 5. Evaluation of CDR3 encoding strategies across k-means, spectral, and hierarchical clustering models for antigen-specific TCR clustering task. The performance of 18 encoding strategies was evaluated via antigen-specific TCR clustering task and four metrics, including ARI, NMI, purity, and $F_{purity>0.9}$, were employed to assess the clustering performance. To alleviate the influence of the target number of clusters on the clustering results, a continuous strategy for K values was adopted, ranging from 10 to 100 with a step size of 5.

Visual analysis explains model performance

To intuitively understand the characteristics of each encoding method, we conducted a visual analysis using the UMAP tool, as depicted in Fig. 6, with further details provided in the Materials and methods section. This analysis enables direct observation of the embeddings' characteristics in the feature space, revealing how well they can group TCRs according to their binding affinity to the epitopes.

Our observations from the visual results align with the conclusions drawn from previous quantitative analysis experiments. For instance, clusTCR emerges as one of the superior encoding methods in our experiments, effectively predicting the binding affinity to epitopes of TCRs. Visual analysis confirms that clusTCR captures critical variations important for distinguishing TCRs binding to different epitopes, representing meaningful differences in the embedding space. Concurrently, such an encoding strategy may tolerate minor sequence alterations that do not impact the TCR sequences' local structure or functional properties, a factor crucial for successfully applying CDR3 embeddings in predictive models of antigen specificity. Conversely, less effective methods produced embeddings with a non-clustered distribution, which can be categorized into two types of scenarios. One scenario involves the embeddings from iSMART and ESM, which are dispersed across the feature space and fail to cluster TCRs with similar functionalities effectively. The other scenario is observed with TCRpeg and Word2vec, where these methods generate overly similar embeddings for a diverse range of TCRs, thereby lacking the granularity needed to discern subtle variations crucial for epitope specificity.

Usability

Figure 2C and Supplementary Fig. S21 detailed our usability assessment, considering computational efficiency and user experience.

Our assessment revealed that NetTCR2.0, Luu *et al.*, and TCRGP are the most efficient, capable of embedding 200 000 TCR sequences within 6 s. Additionally, the running times for most methods generally stabilized below 1000 s. In contrast, even when not accounting for model training time, pMTnet and catELMo were considerably slower, requiring over 20 000 s to encode 200 000 sequences. Detailed running time data are available in Supplementary Table 3.

Memory usage assessments revealed that clusTCR, Word2Vec, and iSMART were the most economical, requiring 100 to 500MB, while methods like ERGO-II demanded over 90 000MB for 20 000 sequences. Despite higher computational demands for certain methods, the data-driven methods evaluated in this study typically featured clear documentation, original datasets, and readable code, collectively contributing to execution and secondary development. Detailed memory usage data can be found in Supplementary Table 4.

Discussion

In our study, we conducted a comprehensive evaluation of various TCR sequence encoding methods across multiple tasks, datasets, and conditions. Our results indicate that no single method consistently outperforms others across all scenarios. This variability highlights the variability in the effectiveness of encoding strategies, which can depend significantly on factors such as dataset generation, model parameter settings, and the specific combinations of algorithms used. Therefore, users should select tools or strategies that best fit their specific application contexts.

Among the evaluated methods, clusTCR, deepRC, and NetTCR2.0 emerged relatively robust, offering consistent reliability across various experimental contexts. As we delve deeper into the performance of specific encoding strategies, it becomes evident that the choice of encoding strategies and feature types

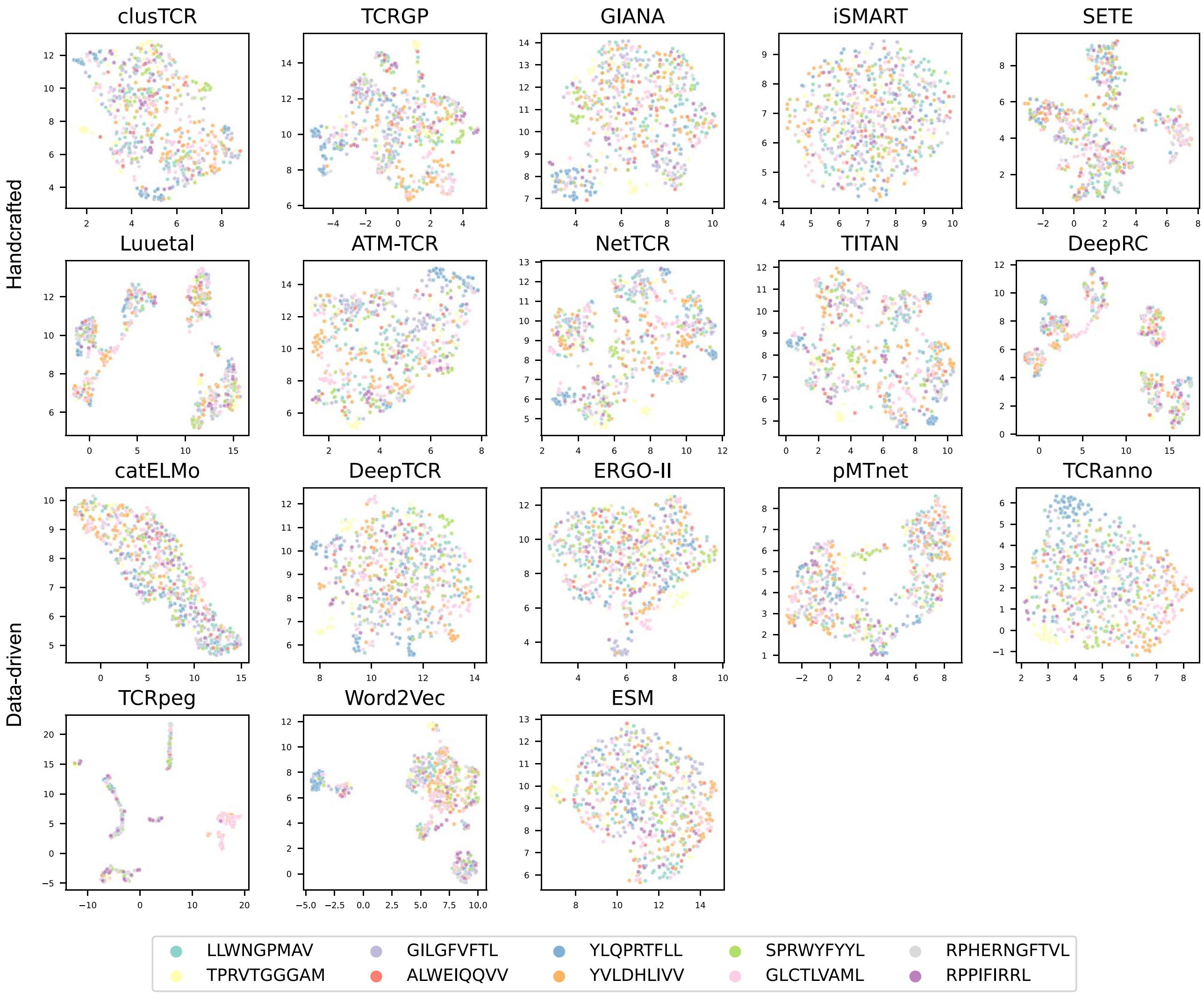

Figure 6. UMAP dimensionality reduction of CDR3 embeddings for the unique epitope dataset. UMAP projections of TCR sequence embeddings reveal distinct clusters corresponding to unique epitope bindings. The embeddings, derived from 18 distinct computational methods, illustrate the aggregation of TCRs by epitope affinity, with each cluster suggesting shared binding characteristics within the dataset. This clustering underscores the potential of these methods to discern the complex patterns of TCR-epitope interactions.

can significantly influence the outcomes of TCR-related tasks. This leads us to thoroughly examine handcrafted and data-driven strategies, assessing their advantages and limitations.

Handcrafted strategies, particularly those that utilize features such as BLOSUM matrices and physicochemical properties, outperformed data-driven strategies in our evaluations. While data-driven strategies, which leverage deep learning frameworks to discern patterns in extensive datasets, might seem advantageous, they did not show superior performance in our observations and incurred higher computational costs. This highlights the effectiveness of handcrafted features in CDR3 encoding, where the complexity of biological data can be captured more efficiently through tailored approaches.

Furthermore, our findings emphasize the need for specifically designed models to handle the unique challenges of CDR3 encoding effectively. General-purpose protein models, such as ESM, do not perform adequately in TCR sequence-related downstream tasks. Although widely recognized in the protein structure encoding field, such models encounter limitations when applied to short protein sequences like CDR3s, which contain subtle and complex sequence variations.

Key Points

- We systematically evaluated 19 TCR CDR3 embedding models across various downstream tasks related to TCR-epitope interaction analysis.
- The findings underscore that handcrafted embeddings surpassed data-driven ones in modeling TCR-epitope interactions.
- We developed an all-in-one TCR CDR3 embedding package comprising all evaluated embedding models.

Acknowledgments

S.C.L. conceived and initiated this study. X.F., M.H., H.L., Y.Y., and Y.J. collected and processed all datasets. X.F., M.H., H.L., and Y.Y. conducted all the experiments. X.F., M.H., H.L., and Y.Y. outputted the figures and tables. X.F., M.H., and Y.J. wrote the manuscript under the supervision of S.C.L. S.C.L. and L.H.

polished the manuscript writing. H.L. and Y.Y. extracted and developed the code of embedding methods. All authors are involved in the discussion and finalization of the manuscript.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Funding

This study was funded by the RGC Healthy Longevity Catalyst Awards (CityU9080002, HLCA/E-107/23), the National Natural Science Foundation of China (32300527), and the Natural Science Basic Research Program of Shaanxi Province (2022JQ-644). The publication fee of this article is covered by the National Natural Science Foundation of China (32300527).

Data availability

The datasets used in this project, including TCR-epitope binding pairs and TCR repertoire data, were obtained from the IEDB [36], VDJdb [37], and McPAS [38] in July 2023, as well as the GIANA project [14] and the catELMo project [32].

All processed datasets are accessible for download at https://github.com/deepomicslab/TCREmbedding/tree/main/dataset.

Code availability

The code is accessible through GitHub via https://github.com/deepomicslab/TCREmbedding.

References

1. Davis MM, Bjorkman PJ. T-cell antigen receptor genes and T-cell recognition. *Nature* 1988;**334**:395–402. https://doi.org/10.1038/334395a0.
2. Turner SJ, La Gruta NL, Kedzierska K. *et al.* Functional implications of T cell receptor diversity. *Curr Opin Immunol* 2009;**21**: 286–90. https://doi.org/10.1016/j.coi.2009.05.004.
3. Wang C, Sanders CM, Yang Q. *et al.* High throughput sequencing reveals a complex pattern of dynamic interrelationships among human T cell subsets. *Proc Natl Acad Sci* 2010;**107**:1518–23. https://doi.org/10.1073/pnas.0913939107.
4. DeWitt III WS, Smith A, Schoch G. *et al.* Human T cell receptor occurrence patterns encode immune history, genetic background, and receptor specificity. *Elife* 2018;**7**:e38358. https://doi.org/10.7554/eLife.38358.
5. Han J, Duan J, Bai H. *et al.* TCR repertoire diversity of peripheral PD-1+ CD8+ T cells predicts clinical outcomes after immunotherapy in patients with non–small cell lung cancer. *Cancer Immunol Res* 2020;**8**:146–54. https://doi.org/10.1158/2326-6066.CIR-19-0398.
6. Cui J-H, Lin K-R, Yuan S-H. *et al.* TCR repertoire as a novel indicator for immune monitoring and prognosis assessment of patients with cervical cancer. *Front Immunol* 2018;**9**:2729. https://doi.org/10.3389/fimmu.2018.02729.
7. Mayer-Blackwell K, Schattgen S, Cohen-Lavi L. *et al.* TCR metaclonotypes for biomarker discovery with tcrdist3 enabled identification of public, HLA-restricted clusters of SARS-CoV-2 TCRs. *Elife* 2021;**10**:e68605. https://doi.org/10.7554/eLife.68605.
8. Beshnova D, Ye J, Onabolu O. *et al.* De novo prediction of cancer-associated T cell receptors for noninvasive cancer detection. *Sci Transl Med* 2020;**12**:eaaz3738. https://doi.org/10.1126/scitranslmed.aaz3738.
9. John-William Sidhom H, Larman B, Pardoll DM. *et al.* Deeptcr is a deep learning framework for revealing sequence concepts within T-cell repertoires. *Nat Commun* 2021;**12**:1605. https://doi.org/10.1038/s41467-021-21879-w.
10. Sidhom J-W, Oliveira G, Ross-MacDonald P. *et al.* Deep learning reveals predictive sequence concepts within immune repertoires to immunotherapy. *Sci Adv* 2022;**8**:eabq5089. https://doi.org/10.1126/sciadv.abq5089.
11. Jiang Y, Huo M, Li SC. TEINet: a deep learning framework for prediction of TCR–epitope binding specificity. *Brief Bioinform* 2023;**24**:bbad086. https://doi.org/10.1093/bib/bbad086.
12. Jiang Y, Li SC. Deep autoregressive generative models capture the intrinsics embedded in T-cell receptor repertoires. *Brief Bioinform* 2023;**24**:bbad038. https://doi.org/10.1093/bib/bbad038.
13. Valkiers S, Van Houcke M, Laukens K. *et al.* ClusTCR: a Python interface for rapid clustering of large sets of CDR3 sequences with unknown antigen specificity. *Bioinformatics* 2021;**37**:4865–7. https://doi.org/10.1093/bioinformatics/btab446.
14. Zhang H, Zhan X, Li B. GIANA allows computationally-efficient TCR clustering and multi-disease repertoire classification by isometric transformation. *Nat Commun* 2021;**12**:4699. https://doi.org/10.1038/s41467-021-25006-7.
15. Tianshi L, Zhang Z, Zhu J. *et al.* Deep learning-based prediction of the T cell receptor–antigen binding specificity. *Nat Mach Intell* 2021;**3**:864–75. https://doi.org/10.1038/s42256-021-00383-2.
16. Hudson D, Fernandes RA, Basham M. *et al.* Can we predict T cell specificity with digital biology and machine learning? *Nat Rev Immunol* 2023;**23**:511–21. https://doi.org/10.1038/s41577-023-00835-3.
17. Bepler T, Berger B. Learning the protein language: evolution, structure, and function. *Cell Syst* 2021;**12**:654–669.e3. https://doi.org/10.1016/j.cels.2021.05.017.
18. Cai M, Bang S, Zhang P. *et al.* ATM-TCR: TCR-epitope binding affinity prediction using a multi-head self-attention model. *Front Immunol* 2022;**13**:893247. https://doi.org/10.3389/fimmu.2022.893247.
19. Pedregosa F, Varoquaux G, Gramfort A. *et al.* Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;**12**: 2825–30.
20. Becht E, McInnes L, Healy J. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol* 2019;**37**: 38–44. https://doi.org/10.1038/nbt.4314.
21. Luo J, Wang X, Zou Y. *et al.* Quantitative annotations of T-cell repertoire specificity. *Brief Bioinform* 2023;**24**:bbad175. https://doi.org/10.1093/bib/bbad175.
22. Widrich M, Schäfl B, Pavlović M. *et al.* Modern Hopfield networks and attention for immune repertoire classification. *Adv Neural Inf Process Syst* 2020;**33**:18832–45.
23. Moris P, De Pauw J, Postovskaya A. *et al.* Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. *Brief Bioinform* 2021;**22**:bbaa318. https://doi.org/10.1093/bib/bbaa318.
24. Luu AM, Leistico JR, Miller T. *et al.* Predicting TCR-epitope binding specificity using deep metric learning and multimodal learning. *Genes* 2021;**12**:572. https://doi.org/10.3390/genes12040572.
25. Pavlović M, Scheffer L, Motwani K. *et al.* The immuneML ecosystem for machine learning analysis of adaptive immune receptor repertoires. *Nat Mach Intell* 2021;**3**:936–44. https://doi.org/10.1038/s42256-021-00413-z.

26. Tong Y, Wang J, Zheng T. *et al*. SETE: sequence-based ensemble learning approach for TCR epitope binding prediction. *Comput Biol Chem* 2020;**87**:107281. https://doi.org/10.1016/j.compbiolchem.2020.107281.
27. Jokinen E, Huuhtanen J, Mustjoki S. *et al*. Predicting recognition between T cell receptors and epitopes with TCRGP. *PLoS Comput Biol* 2021;**17**:e1008814. https://doi.org/10.1371/journal.pcbi.1008814.
28. Montemurro A, Schuster V, Povlsen HR. *et al*. NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. *Commun Biol* 2021;**4**:1060. https://doi.org/10.1038/s42003-021-02610-3.
29. Zhang H, Liu L, Zhang J. *et al*. Investigation of antigen-specific T-cell receptor clusters in human cancers. *Clin Cancer Res* 2020;**26**:1359–71. https://doi.org/10.1158/1078-0432.CCR-19-3249.
30. Weber A, Born J, Martínez MR. Titan: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics* 2021;**37**:i237–44. https://doi.org/10.1093/bioinformatics/btab294.
31. Springer I, Tickotsky N, Louzoun Y. Contribution of T cell receptor alpha and beta CDR3, MHC typing, V and J genes to peptide binding prediction. *Front Immunol* 2021;**12**:664514. https://doi.org/10.3389/fimmu.2021.664514.
32. Zhang P, Bang S, Cai M. *et al*. Context-aware amino acid embedding advances analysis of TCR-epitope interactions. *eLife* 2023;**12**:RP88837. https://doi.org/10.7554/eLife.88837.2.
33. Rives A, Meier J, Sercu T. *et al*. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS* 2021;**118**:e2016239118. https://doi.org/10.1073/pnas.2016239118.
34. Rao RM, Meier J, Sercu T. *et al*. Transformer protein language models are unsupervised structure learners. *International Conference on Learning Representations* 2021. https://openreview.net/forum?id=fylclEqgvgd.
35. Peters ME, Neumann M, Iyyer M. *et al*. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 2018;**1**:2227–37. https://doi.org/10.18653/v1/N18-1202.
36. Vita R, Mahajan S, Overton JA. *et al*. The immune epitope database (IEDB): 2018 update. *Nucleic Acids Res* 2019;**47**:D339–43. https://doi.org/10.1093/nar/gky1006.
37. Goncharov M, Bagaev D, Shcherbinin D. *et al*. VDJdb in the pandemic era: a compendium of T cell receptors specific for SARS-CoV-2. *Nat Methods* 2022;**19**:1017–9. https://doi.org/10.1038/s41592-022-01578-0.
38. Tickotsky N, Sagiv T, Prilusky J. *et al*. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics* 2017;**33**:2924–9. https://doi.org/10.1093/bioinformatics/btx286.

Briefings in Bioinformatics, 2025, **26(1)**, bbaf030
https://doi.org/10.1093/bib/bbaf030
Problem Solving Protocol