NAR Genomics and Bioinformatics, 2025, **7**, lqaf065
https://doi.org/10.1093/nargab/lqaf065
Standard Article

OXFORD

TCRCluster: a novel approach to T-cell receptor latent featurization and clustering using contrastive learning-guided two-stage variational autoencoders

Yat-Tsai Richie Wan * **and Morten Nielsen**

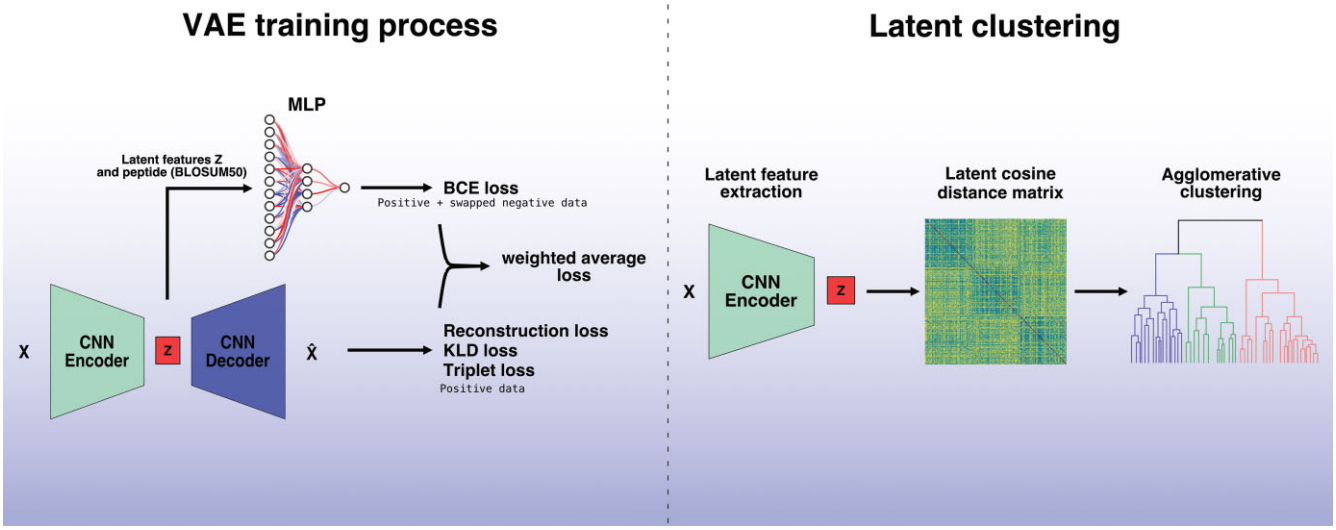
Department of Health Technology, Technical University of Denmark, Kgs Lyngby DK 28002, Denmark

*To whom correspondence should be addressed. Email: riwa@dtu.dk

Abstract

T cells play a vital role in adaptive immunity by targeting pathogen-infected or cancerous cells, but predicting their specificity remains challenging. Encoding T-cell receptor (TCR) sequences into informative feature spaces is therefore crucial for advancing specificity prediction and downstream applications. For this, we developed a variational autoencoder (VAE)-based model trained on paired TCR α–β chain data, incorporating all six complementarity-determining regions. A semi-supervised 'two-stage VAE' framework, integrating cosine triplet loss and a classifier, was found to further refine peptide-specific latent representations, outperforming sequence-based methods in specificity prediction. Clustering analyses leveraging our VAE latent space were evaluated using *K*-means, agglomerative clustering, and a novel graph-based method. Agglomerative clustering achieved the most biologically relevant results, balancing cluster purity and retention despite noise in TCR specificity annotations. We extended these insights to evaluate TCR repertoire data. Across datasets, VAE-based models outperformed sequence-based methods, particularly in retention metrics, with notable improvements in the SARS-CoV-2 repertoire dataset. Moreover, the cancer repertoire analysis highlighted the generalizability of our approach, where the model displayed high performance despite minimal similarity between the training and test data. Collectively, these results demonstrate the potential of VAE-based latent representations to offer a robust framework for prediction, clustering, and repertoire analysis.

Graphical abstract



Introduction

T cells are a fundamental component of the adaptive immune system, responsible for recognizing and eliminating pathogen-infected or cancerous cells. This recognition is mediated by the T-cell receptor (TCR), which can bind to peptides displayed on major histocompatibility complex (MHC) molecules on the surface of target cells, termed peptide–MHC complexes (pMHCs). The specificity of this interaction is determined primarily by six complementarity-determining regions (CDRs) located on the α and β chains of the TCR. Among these, the CDR3 regions are predominantly responsible for binding to the peptide, while CDR1 and CDR2 primarily interact

Received: January 24, 2025. **Revised:** May 7, 2025. **Editorial Decision:** May 9, 2025. **Accepted:** May 13, 2025

with the MHC [1, 2]. The diversity of targets recognized by TCRs lies mainly in the variability of CDR3s. Often TCRs that bind the same target pMHC share a common structural and/or amino acid motif in the CDR regions, forming specificity groups within patient TCR repertoires [3–5]. These repertoires vary largely between individuals, and analyses of its composition and diversity can provide insights into the immune system's response to infections or tumours and the overall health status of an individual [3, 4].

Considering the centrality of TCR–pMHC interaction in immune responses, accurately predicting TCR specificity and identifying specificity clusters have become key areas of research [6]. However, challenges arise due to the vast diversity of TCR sequences, the limited availability of comprehensive paired-chain datasets, as most studies focus on the CDR3β region alone, and the often high noise levels in the experimental data [7–9]. Databases such as IEDB [10], VDJdb [11], and McPas-TCR [12] provide TCR–pMHC data, focusing on positive interactions. Others such as OTS [13] and TCRdb [14] are centred around unlabelled TCR repertoire data. Despite recent advancements in single-cell sequencing technologies, enabling the analysis of paired α- and β-chain data, the scarcity and redundancy of TCR–pMHC interaction data continue to hinder comprehensive and accurate model development.

A myriad of machine learning (ML) models have been previously developed for TCR clustering (examples include ClusTCR [15], GLIPH [16], and TCRdist [17]) and antigen specificity prediction (examples include TCRMatch [18], TCRbase [19], NetTCR [19–21], and TCRex [5]). One form of models focuses on sequence distances, with some using only CDR3s such as ClusTCR [15] and GLIPH [16], while others such as tcrdist3 [17] and TCRbase [19] include all six loops. Another form extracts features from TCR sequences, using various ML architectures such as convolutional neural networks (CNNs) for NetTCR [19–21], variational autoencoders (VAEs) for DeepTCR [22], and more recently transformer-based models such as TULIP [23] and BERTrand [24]. While these methods demonstrate high predicting power in the narrow space of pHLAs with characterized TCR binders, challenges remain in generalizing TCR clustering and predictions to unseen peptides [25]. In terms of clustering, most methods used in aforementioned studies such as *K*-means or agglomerative clustering rely on hyperparameter tuning, adding another layer of complexity [15, 17, 22].

To complement these earlier methods, we here present a two-fold contribution: First, we develop a semi-supervised VAE model, capable of learning a lower-dimensional, probabilistic representation of TCRs in a latent space, capturing the variability and structure of the data. This latent representation can be effectively used for downstream tasks such as similarity comparisons, clustering, and specificity prediction. Next, we investigated different methods for TCR clustering, including *K*-means, agglomerative, and a novel graph-based method termed Top1Cut-weighted, based on the Girvan–Newman algorithm [26]. Benchmark studies of these methods demonstrate that the latter two methods are able to generalize and maintain robust performance both in a peptide-specificity setting and in patient repertoire data in three disease settings. Our study further highlights the importance of defining multiple clustering metrics, as cluster purity and retention alone fail to accurately represent performance.

Materials and methods

Training data

The training data are made up of full-length TCR sequences, with annotation of all three CDRs of both the α and β chains, as well as their known cognate peptide. TCR sequences targeting MHC-I epitopes were downloaded from IEDB [10], VDJdb [11], and 10x in October 2023. The 10x dataset was denoised with ITRAP as described by Povlsen *et al.* [9]. The raw data were first filtered to include only sequences with complete V and J gene annotations and CDR3 sequences for paired α and β chains. Filtered data were then passed through Stitchr [27] to reconstruct full TCR sequences, followed by ANARCI [28] for annotation of CDR1, CDR2, and CDR3 regions (here, CDR1 was defined as positions 27–38, CDR2 as positions 56–65, and CDR3 as positions 105–117) [21]. To address redundancy, sequences were encoded using a BLOSUM62 matrix, and a Hobohm-1 algorithm was applied with a 95% similarity threshold to reduce redundant entries. This resulted in 9769 unique TCRs interacting with 170 different epitopes.

The resulting non-redundant dataset was randomly partitioned into five partitions. Three partitions (2–4) were used for training, one partition (0) for validation during training, and one partition (1) was kept as a held-out test partition. The dataset was augmented with five negative examples per positive example by swapping TCRs between peptides with low sequence similarity (Levenshtein distance >3 between peptide sequences) as described earlier [21]. This peptide similarity threshold was imposed guided by earlier findings suggesting that cross-reactive peptides most often share two or fewer mutations [21]. Note that we here opted for the use of swapped TCRs as negatives as opposed to negatives obtained, for instance, from 10x datasets stained against a panel of HLA multimers or from healthy controls. This is because such negatives, in earlier studies, have turned out to be trivially separable from the positive TCR, without informing the method on the TCR specificities in the positive data [7, 20].

To avoid biases coming from epitopes with few TCRs, we constructed a limited dataset covering only peptides characterized by at least 20 positive TCR data points. This reduced the dataset to cover 17 peptides. All these datasets are available at https://services.healthtech.dtu.dk/services/TCRcluster-1.0/ as well as at https://doi.org/10.5281/zenodo.15350361. The datasets as well as the code used to train the models are also available at https://github.com/mnielLab/tcrcluster_backend and https://doi.org/10.5281/zenodo.15388480.

Evaluation data

Public patient TCR datasets [29–31] were downloaded from OTS [13] in July 2024. The repertoires from Garner *et al.* [30] were combined, and duplicates were removed to be used as the healthy control to sample from. The merged repertoires were further filtered to keep only TCR sequences with a count value equal to 1 (to exclude expanded clones), and sequences appearing in both the healthy and diseased repertoires were removed. For each disease repertoire, we sampled equal amounts of healthy TCR with different seeds ($n = 100$) to create mixed datasets for the patients repertoire analysis. The COVID dataset consists of 43 repertoires bootstrapped 100 times, totalling 4300 runs. Both peptide specificities (QVDYYGLYY and KSAIVTLTY) in THE cancer dataset for QVDYYGLYY-specific TCRs consist of four repertoires bootstrapped 100 times, totalling 400 runs each.

Model

Variational autoencoder

Our VAEs follow a standard encoder–decoder architecture, utilizing Kingma and Welling's reparameterization trick [32]. To preserve the data's sequential structure, we implemented a convolutional VAE (CNNVAE) [33–35], using convolutional layers [36], to encode the input sequences to the latent space and transposed convolutional layers [37] to decode the latent vectors.

Input vectorization

Sequences for all six CDR loops were padded with negative values to a given maximum length, concatenated and encoded using a BLOSUM50 matrix into the final input. To differentiate the six loops, we used a positional vector, a one-hot encoding of dimension 6 indicating the CDR loop number within the input. This resulted in vectors of shape (73, 26) to be fed into the convolutional layers. The dimensions reflect the sum of the maximum length of each CDR loop (73) and the 20 amino acid vector + 6 positional vector (26). Peptide labels were ordinally encoded to be used in the contrastive loss. That is, by way of example, the labels were encoded as [0, 1, 2, 3, 4, 5, 6] in the case of a seven-class problem. The peptide sequence was BLOSUM50 encoded, flattened and concatenated to the TCR latent vector, and its binder status encoded with [0, 1] to use in a binary classification task for the two-stage VAE set-up (see below).

Architecture

Hyperparameter selection

All hyperparameters such as latent dimension, convolution kernel sizes, activation function, and optimal loss weights were selected based on a small-scale hyperparameter optimization. The optimal hyperparameters were chosen based on the validation set performance of the VAE model for sequence reconstruction and the area under the curve (AUC) of the retention–purity curves obtained by applying agglomerative clustering on the associated latent distance matrix.

Convolutional variational autoencoder

The model was constructed in Python 3.11.5 with PyTorch 2.0.0. The encoder compresses the input and uses two 1D convolutional layers with a kernel size of 9 and a stride of 4, to capture patterns of a TCR interacting directly with a cognate peptide–MHC complex. The convolutions are followed by linear layers to extract the mean (μ) and standard deviation (σ) of the latent distribution with a final latent dimension of 128. The decoder uses a linear layer to decompress the latent vector and transposed convolutions with a kernel size of 9 and a stride of 4 to reconstruct the input data. All convolution and transposed convolution layers use SELU as activation function and BatchNorm for regularization (see Fig. 1A).

Two-stage CVAE

We further designed a two-stage CVAE, where an MLP of three layers is trained in tandem with the CVAE by taking the latent TCR vector concatenated to a BLOSUM50-encoded peptide sequence. The MLP layers are activated by ReLU, followed by regularization with BatchNorm and the dropout with a probability of 0.2 (see Fig. 1B).

Loss function and training regimes

The models were trained by minimizing the expected lower bound loss (ELBO loss) as described by Kingma and Welling and uses the β-VAE [38] approach to constrain the KLD part of the loss with a β of 1e−2. The weight β of the KLD is controlled using a tanh annealing regime followed by a linear decrease after a plateau period (see Supplementary data).

The reconstruction loss between the input and reconstructed vector is measured using mean squared error for the BLOSUM-encoded sequence, with a 3:1 weight ratio to prioritize the CDR3 loops and a BCE loss for reconstructing the positional vector.

In the semi-supervised set-up, the two-stage loss used a BCE loss as the classifier objective with a warm-up period where the weight of the classification objective is set to 0.

A contrastive loss (triplet) loss [39] was implemented to further control the latent distances between data points. Each data point is paired with an 'anchor point', where the loss function penalizes large distances for points with the same peptide specificity, while pushing away points with different specificities (see Fig. 1C). This behaviour is shown by the equation below, where $d_{\cos}(X, Y)$ is the latent cosine distance between two data points X and Y:

$$\text{cosine triplet loss} = \max\left(|\, d_{\cos}(A,P)\,| - |\, d_{\cos}(A,N)\,| + \text{margin}, 0\right).$$

A custom batching strategy was developed to ensure that at least one anchor point was available for each peptide specificity group for the positive part of the triplet loss in each mini-batch during training.

TCRbase

TCRbase was run using a stand-alone version of the method described by Montemurro *et al.* [19]. In short, TCRbase assigns a score by comparing query data containing positive and swapped negatives TCR (see earlier) sequences for a given peptide to a database containing only positive TCRs, picking the most similar and using its similarity as score. Individual weights were used on each of the six CDR loops (1, 1, 4, 1, 1, and 4 for CDR1A, CDR2A, CDR3A, CDR1B, CDR2B, and CDR3B, respectively) to compute an aggregated sequence similarity.

Clustering methods

Distance matrices

Latent distance matrices were created by calculating the cosine distance between the latent embeddings of each pair of points in the dataset. TCRbase [19] distances were computed as described above. tcrdist3 [17] version 0.2.2 was used for computing tcrdist3 distance matrices. The package was installed from the public repository at https://github.com/kmayerb/tcrdist3.

Clustering

The agglomerative and *K*-means baseline clusterings were made using the implementations in Scikit-Learn version 1.3.0 [40] with either distance matrices or latent feature vectors.

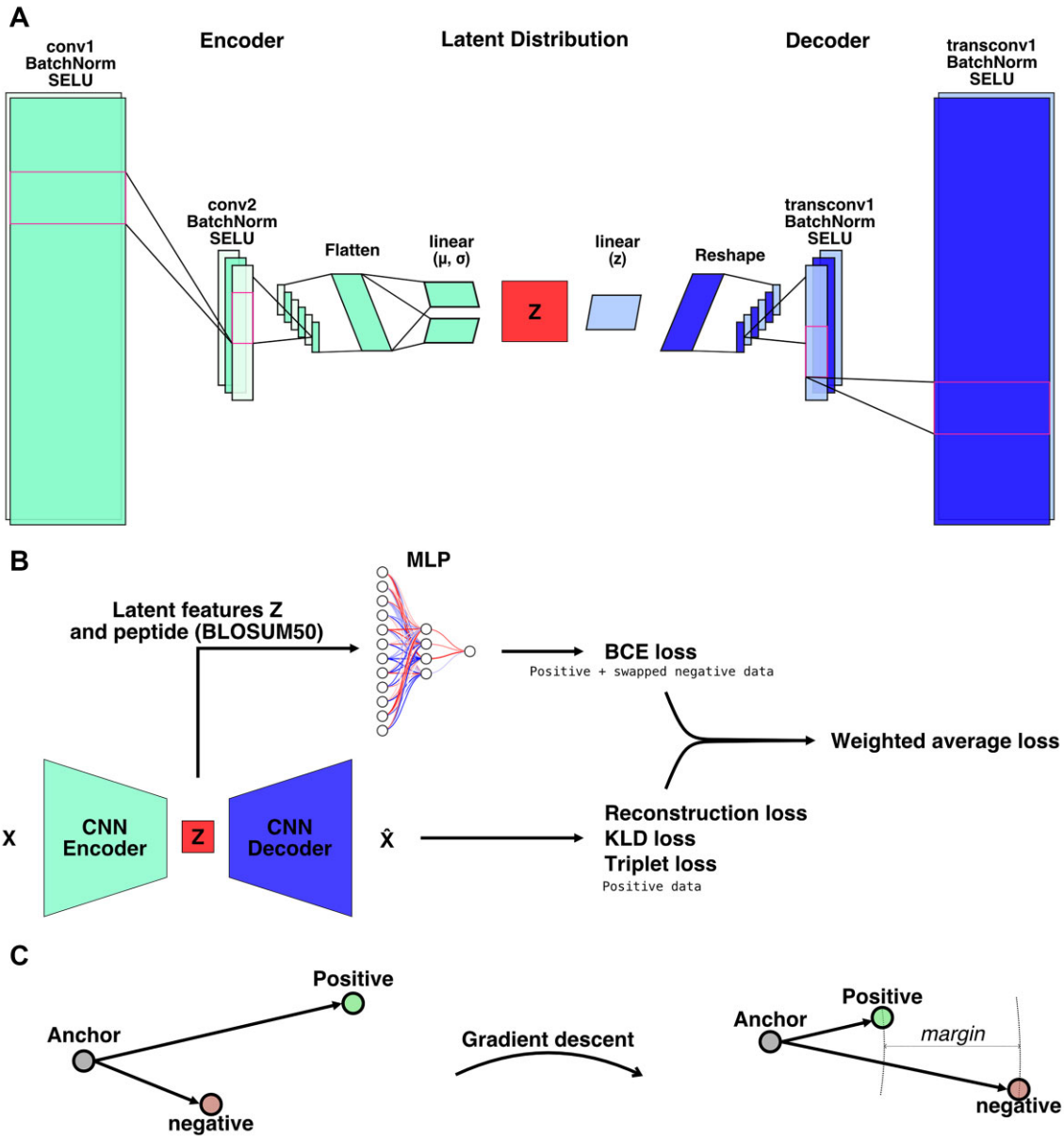


Figure 1. Architecture and training process of the models. (**A**) The architecture of the one-stage CNNVAE model. The input data consist of all six CDR loops encoded with a BLOSUM50 matrix and a positional vector (one-hot encoded) to indicate the position in the sequence. Each convolutional layer consists of either a convolution to compress or a transposed convolution to decompress the input for the encoder and decoder, respectively. BatchNorm was used for regularization and SELU as activation function. The features extracted by the encoder are flattened and passed through linear layers to derive the mean (μ) and standard deviation (σ) of the latent distribution. These parameters are then used in the reparameterization trick to generate the latent vector *Z*. (**B**) The two-stage model architecture begins with a CNNVAE in the first stage. Here, an MSE loss measures the reconstruction accuracy between the reconstructed vector and the input. Additionally, Kullback–Leibler divergence (KLD) and a cosine-distance triplet loss are applied to the latent vector *Z*. In the second stage, the latent vector *Z* is concatenated with a BLOSUM50-encoded peptide sequence. This concatenated input is then passed through a two-layer multi-layer perceptron (MLP), where a binary cross-entropy (BCE) loss is calculated between the predicted and true labels to backpropagate gradients up to the first stage. (**C**) The triplet loss starts with three points: an anchor, a positive example (same peptide specificity as the anchor), and a negative example (different peptide specificity). Through gradient descent, the model learns to increase the distance between the anchor and the negative example while reducing the distance between the anchor and the positive example to create a defined margin.

Minimum spanning tree and Top1Cut

Minimum spanning trees (MSTs) were generated from distance matrices using NetworkX version 3.1 with the Kruskal algorithm [41] (see Fig. 2A). The Top1Cut algorithm was developed based on a distance-weighted adaptation of the Girvan–Newman betweenness centrality algorithm [26]. Here, each edge's betweenness centrality is weighted by the distance (e.g. weight) of that edge and used as a score to cut the MST into clusters. At each iteration, the highest ranking edge is cut, and a silhouette (SI) score is computed. The optimal clustering is selected based on the iteration with the highest SI score (see Fig. 2B).

Retention–purity curves

Retention–purity curves were generated through iterations (for the Top1Cut methods) or by iterating through hyperparameters (distance threshold for aggregation and *K*) for agglomerative and *K*-means clustering, respectively. The optimal

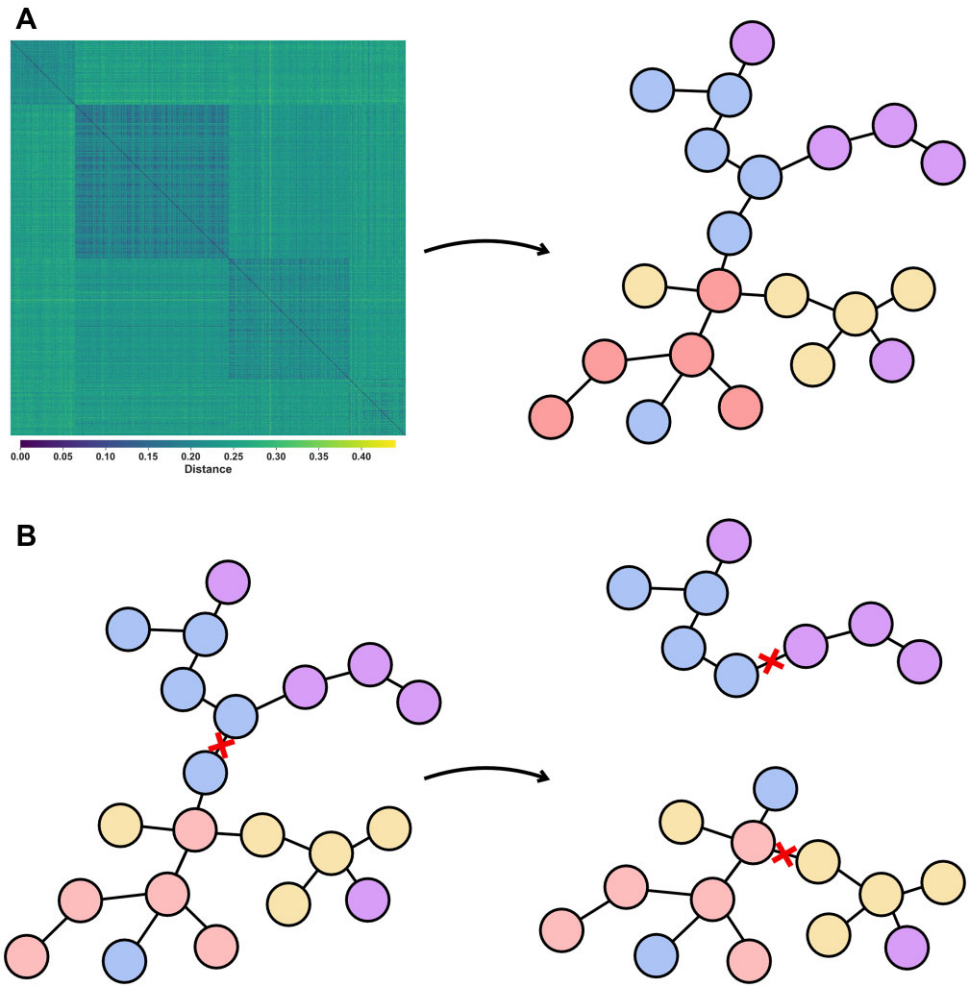


Figure 2. (**A**) An MST is generated from a distance matrix (latent space or sequence space). This sample distance matrix was sorted by peptide specificity. Each colour in the cartoon graph model represents a given label (peptide specificity). (**B**) The edge with the largest distance-weighted betweenness centrality is selected and cut at each iteration, repeating until the optimal SI score is reached.

'iteration' was defined as the one resulting in the highest SI index.

Results

Our main goal was to design a model that accurately encodes TCR sequences in a latent feature space, motivated by the need for precise feature extraction to support downstream applications and analyses. To achieve this, we opted for VAEs [32], owing to their unsupervised nature and their ability to learn an underlying latent distribution of the data. Our models were trained on paired TCR α–β chain data, interacting with 170 different epitopes (for details refer to the 'Materials and methods' section). Latent spaces between models trained on different partitions are not trivially aggregated. For instance, extracting a 'mean latent vector' from models trained in a cross-validation setting is not feasible, as each model may have compressed different aspects of the data along each dimension. As such, we instead opted for a training, validation, and test split of our dataset into five partitions, using three for training, one for validation, one for testing (see the 'Materials and methods' section). Though the contact between a TCR and its cognate peptide–MHC complex occurs mainly through the CDR3β loops [42–45], previous studies have demonstrated the importance of including the three CDR loops from both the α and β chains in order to assess and predict the TCR specificity [19, 21, 46, 47]. Therefore, our models were trained including all six CDRs.

Generating a tractable latent space

To assess the latent representation of the TCR sequences, we first investigated the model's ability to accurately reconstruct the TCR input. We compared multiple hyperparameters, picking the best model based on validation accuracy. The results show that using a lower weight for the KLD term with an annealing schedule on the weight of the KLD term improved reconstruction from the latent vector, achieving up to 99.3% average sequence reconstruction accuracy for the best model on both the validation (data not shown) and test datasets (see Fig. 3).

Previous studies have noted the importance of 'disentangling' the latent space to improve its usefulness for downstream tasks as well as the relationship between a disentangled representation and a poorer reconstruction accuracy [38, 48]. Seeking to tune our latent space for specific downstream tasks, we introduced a semi-supervised approach with multiple levels of semi-supervision. This involved using a cosine-distance triplet loss on the latent dimensions. Additionally, we trained a

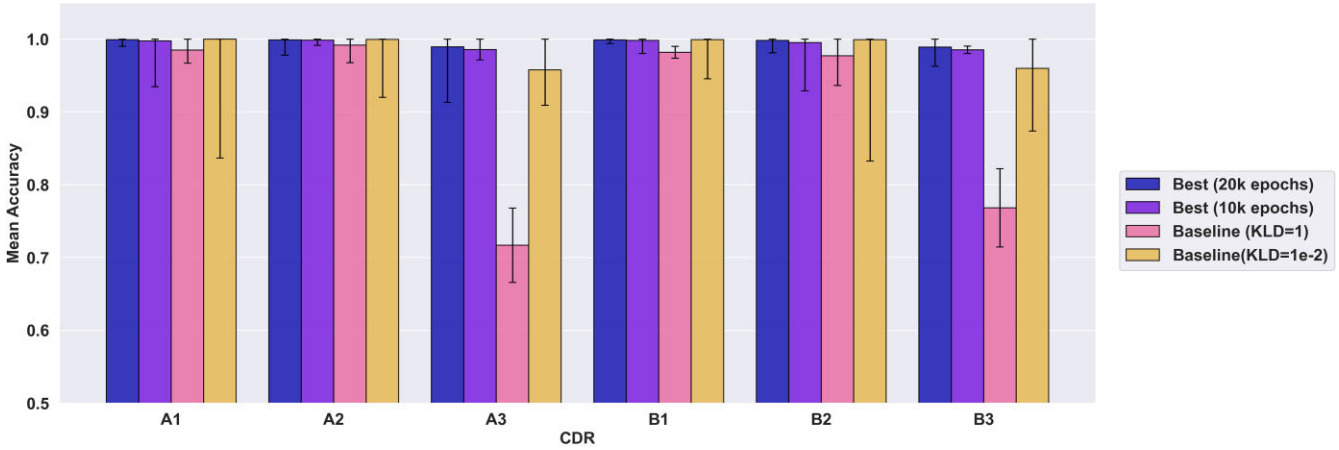


Figure 3. Reconstruction accuracies for all models tested. The models labelled 'Best' were trained using additional tricks such as a custom loss regime and weights for the reconstruction loss of individual CDRs (increased weights for CDR3a and CDR3b). The baseline models were trained without any specific changes except the weight of the KLD. The 'Best' models were trained with a custom regime for the weight of the KLD and an increased weight placed on the reconstruction loss on the CDR3a and CDR3b chains. The 'Best' models were selected as the models with the highest reconstruction accuracy at 10 000 and 20 000 epochs, respectively. The baseline methods were trained until 10 000 epochs using a weight on the KLD of either 1 or 1e−2.

simple one-hidden layer FFNN in tandem with the VAE using the latent features, creating what we term a 'two-stage VAE' to force the latent space to learn patterns based on peptide specificities (see the 'Materials and methods' section). Our model using both the cosine triplet loss and the two-stage framework uses 'two levels' of semi-supervision.

As a first benchmark, we benchmarked the power of the latent space representation of the VAE models against the sequence similarity-based method TCRbase. In short, we compare query sequences to a database of positive TCRs to a given peptide, assigning a prediction score from the similarity to the most similar hit. Next, these scores can be used to compute a receiver operating characteristic AUC for the query dataset for the given peptide. Partial AUCs for a maximum false positive rate of 10% (AUC 0.1) were also computed. The latter calculated using McClish standardization [49]. For more details on the original TCRbase implementation, refer to [19]. Here, we used this approach with our VAEs by using cosine similarities between latent space vectors as the similarity metric. We further included in the benchmark a version of the VAE trained using only the CDR3 loops (α and β). To avoid AUC distributions being skewed by peptides with few known TCR binders, we only report the AUC for the peptides with at least 20 known binders in the test partition, leaving TCRs specific to 17 peptides out of the 170 in the dataset. The AUCs of this benchmark are summarized in Fig. 4A. In line with previous studies, the model using only CDR3 loops performed worse than TCRbase using all six CDR loops. Next, including all loops in our model resulted in a substantial gain in performance, surpassing TCRbase. Following this, we included a cosine triplet loss to our VAE. This resulted in a minor drop in performance. However, when integrated in the semi-supervised 'two-stage VAE' model, the cosine triplet loss improves the AUC, performing better overall than all other models as well as the baseline. This trend also holds true when comparing AUC 0.1 values (see Supplementary Fig. S1). This suggests that the latent space of the 'two-stage VAE cosine triplet loss' model is better defined for TCR specificity prediction, with an improved separation of the distance distributions between binders and swapped negatives. This is further supported by the similarity distribution, where the bottom right panel shows an improved separation between the two classes (binders and non-binders) when altering the VAE architecture from the simple CDR3 model to the model using all six CDRs and two-stage learning combined with cosine triple loss (see Fig. 4B). For details on the 17 peptides and distribution shift plots for all models on the 17-peptide test set, refer to Supplementary Fig. S2.

Clustering in a multi-peptide specificity space

To expand from a per-peptide evaluation basis, we next sought to evaluate various clustering strategies as a means to identify antigen specificity groups in disease settings [4, 16, 50]. Here, we investigated several methods including *K*-means, agglomerative, and a novel graph-based approach termed Top1Cut based on MSTs [41] combined with an algorithm based on edge scores to define cuts to trim the resulting graph into clusters [26, 51]. Here, the edge score is defined from its betweenness centrality, calculated as the total proportion of all shortest paths between node pairs that traverse that edge. This score is potentially weighted (Top1Cut-weighted) by the distance (defined by latent cosine or sequence similarity) between two TCRs (nodes), and in each iteration the top scoring edge is removed. For all methods, the optimal clustering was selected by use of the SI score [52, 53] (refer to the ''Materials and methods' section). For *K*-means clustering, the latent vector representation of the TCR was used as input. From both the agglomerative and Top1Cut clustering, the distance matrices were generated using cosine distances between the VAE latent vector representations of the TCRs.

In this benchmark, the same subset of TCRs binding to 17 peptides defined in the previous section was used. We first evaluated the latent space of our two-stage cosine VAE model on this data subset using t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) (see Supplementary Fig. S3). We observe that while some peptide specificities (GILGFVFTL, LLWNGPMAV, RAKFKQLL, and YLQPRTFLL) appear to form clusters in the latent space, for most parts the TCRs lie scat-

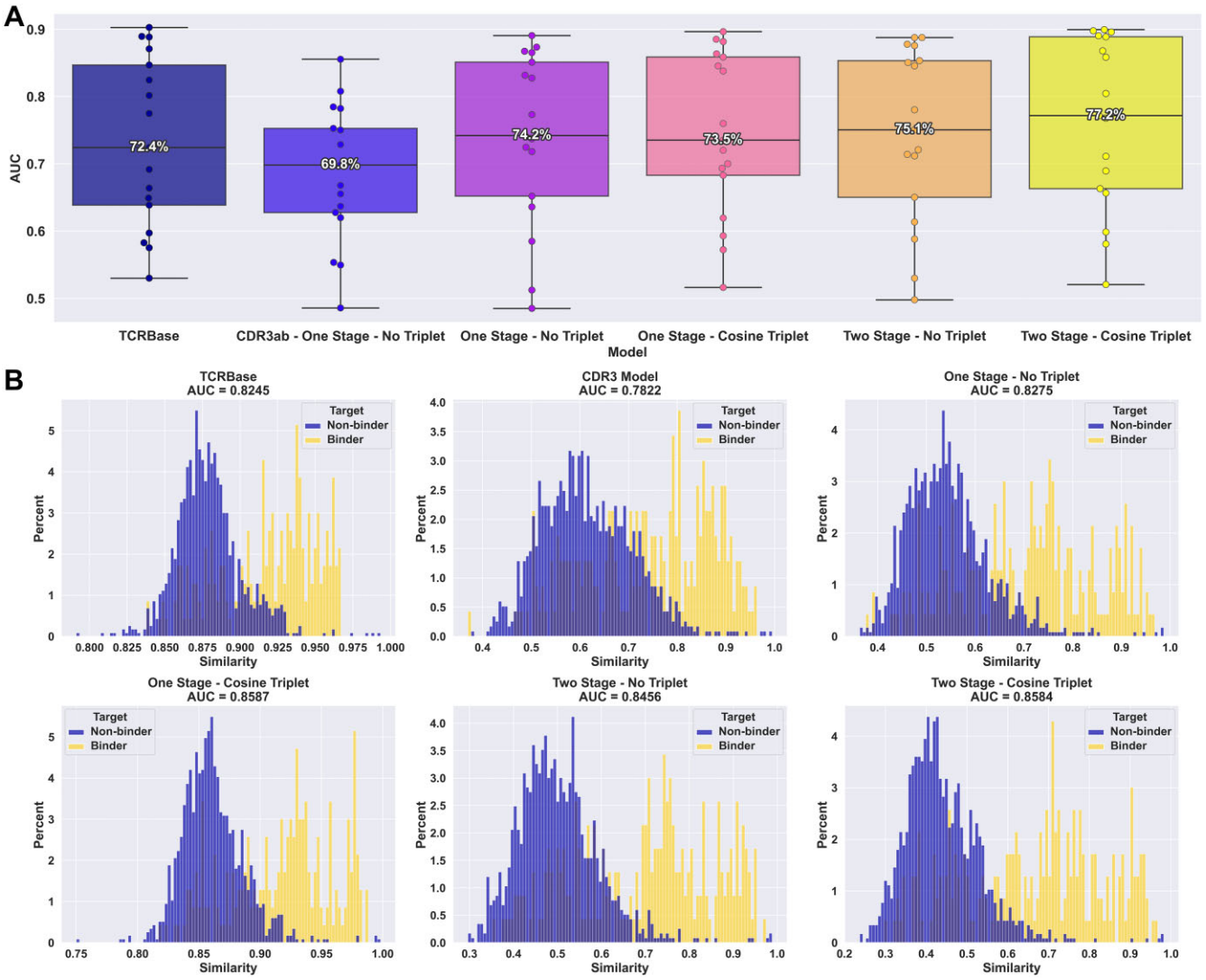


Figure 4. Comparison of similarity methods with TCRbase's approach on the 17-peptide test set. TCRbase or a VAE was used to generate either the sequence or latent cosine similarity, respectively. (**A**) Boxplot of AUCs on the test set for 17 peptide specificities. Each point shows the AUC value for TCRs to a given peptide specificity. (**B**) 'Hit' distance distributions for each class label for GILGFVFTL. The similarity between the query (binder and non-binder) and the database (binders only) was computed and a hit is defined as the data point with the highest similarity (sequence or latent cosine).

tered around. However, such visualizations can be misleading due to several inherent limitations. Both t-SNE and UMAP prioritize preserving local neighbourhood structures, often at the expense of accurately representing global relationships within the data and are sensitive to hyperparameters and initialization [54, 55]. This focus can distort the true topology of the latent space, leading to misinterpretations of cluster proximities and separations. Following this, we focused on the latent cosine distance matrices and compared the performance of the different clustering methods using the two-stage cosine VAE latent space representation. The clustering performance was evaluated in terms of the mean cluster purity (defined as the proportion of TCRs sharing the majority specificity) and retention (i.e. the proportion of TCR placed in clusters of size 2 and above). Curves of average purity, retention, and SI scores for the different methods at each clustering iteration (or hyperparameter) are shown in Fig. 5A and B, from which retention–purity curves were generated accordingly (see Fig. 5C; refer to the ''Materials and methods' section for details).

In these retention–purity curves, at high retention (in the early part of the clustering iterations), the cluster sizes are large and the purity is overall low. Figure 5A–C shows the inherent trade-off between cluster purity and retention. That is, as the cut iterations proceed, the purity increases, resulting in lower retention (e.g. fewer, smaller clusters). Figure 3C shows that all methods (with the exception of Top1Cut-unweighted) fare similarly in the retention range of 0%–50%, while Top1Cut-weighted achieves the highest performance for retention above 50%. When looking at the optimal clustering obtained (based on maximizing the SI score), the Top1Cut-weighted method is the only algorithm with an average purity nearing 70% (see marks on the curves in Fig. 5C). However, despite finding pure clusters and retaining over 70% of the dataset using this method, a key observation is that the solutions with optimal SI value for both the agglomerative and Top1Cut methods are formed by clusters of very small size (see Fig. 5A and B). In fact, the mean cluster sizes of both the agglomerative clustering and Top1Cut methods share an average cluster size around 3 at optimal SI (see Fig. 5A and

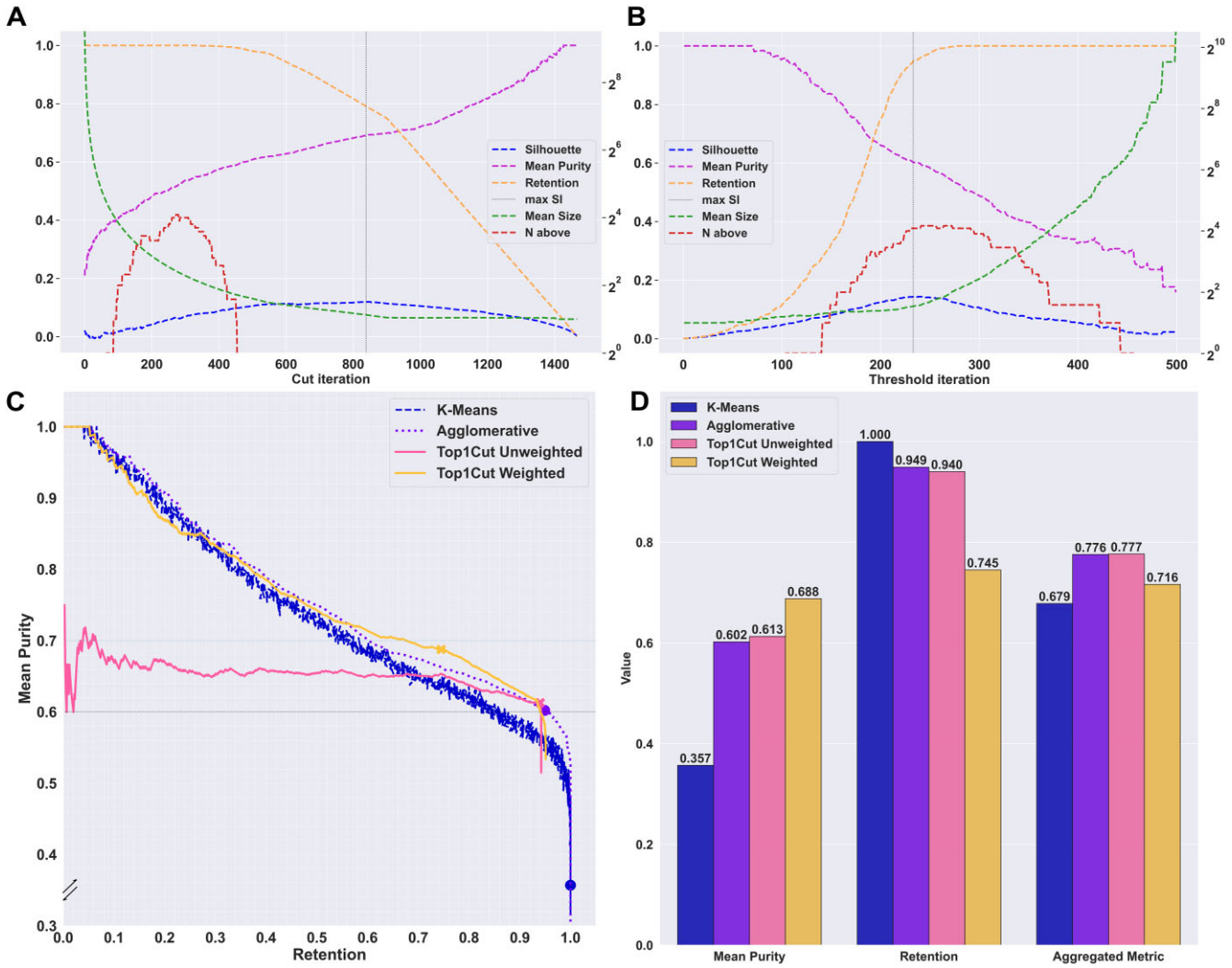


Figure 5. Benchmark of clustering methods on the 17-peptide subset. (**A**) Metrics at each iteration for the Top1Cut-weighted algorithm. Values on the main *Y*-axis (SI, mean purity, and retention) range from 0 to 1. Values on the secondary *Y*-axis (mean size and *N* above) range from 1 to 2^8 on a $\log_2$ scale. Max SI indicates the optimal clustering iteration. *N* above indicates the number of clusters with a purity above 75% and size of 5 or above. (**B**) Metrics at each distance threshold used for the agglomerative algorithm. The agglomerative algorithm is a bottom-up approach; hence, the progression of mean purity and retention is inverted with respect to iterations compared to Top1Cut-weighted. (**C**) Retention–purity curves for each of the methods benchmarked. The markers indicate the optimal SI solution. (**D**) Metrics for the optimal SI solution for each of the methods benchmarked. Aggregated metric is defined as the mean of mean purity and retention.

B). Further, looking at the number of clusters with a purity above 75% and size of 5 and above (red curves in Fig. 5A and B), we observe that the Top1Cut-weighted method maintains no such clusters at the optimal SI solution, and similar conclusions can be reached for the Top1Cut-unweighted and *K*-means methods (see Supplementary Fig. S4). In contrast, for the agglomerative clustering, the optimal SI solution falls very close to the peak number of such clusters. Together, these observations suggest that given the current data, the agglomerative method achieves the most biologically relevant clustering solution despite resulting in a slightly decreased cluster purity.

Analysis of top clusters and benchmark of distance matrices

A key observation from the cluster size analysis was the overall small size of the clusters in the optimal SI solution for all the investigated methods. We speculate that noise of mislabelled and/or wrongly annotated TCR data would be the main source leading to this behaviour. The presence of such noise in TCR specificity data is a known and well-documented phenomenon [7–9]. With the 17-peptide subset, we observe clear signals of such noise when using both the VAE and TCRbase distance matrices sorted by peptide specificities (see Supplementary Fig. S5). That is, we observe TCRs annotated to the same peptide often found to share very low mutual similarity, as evidenced by the lack of dense, low distance clusters along the diagonal. And likewise, TCRs annotated to a given peptide were found to show high similarity to TCRs specific to another peptide, as apparent from the occurrence of low-distance cluster regions away from the diagonal. We hypothesize that these properties overall will result in the formation of small clusters when using the SI score as clustering metric.

To verify the hypothesis, we conducted an analysis using synthetic distance matrices with different degrees of defined noise and data from the 17-peptide subset denoised to different degrees of purity (for details on the experiment and its results, refer to Supplementary data). In short, the outcome of

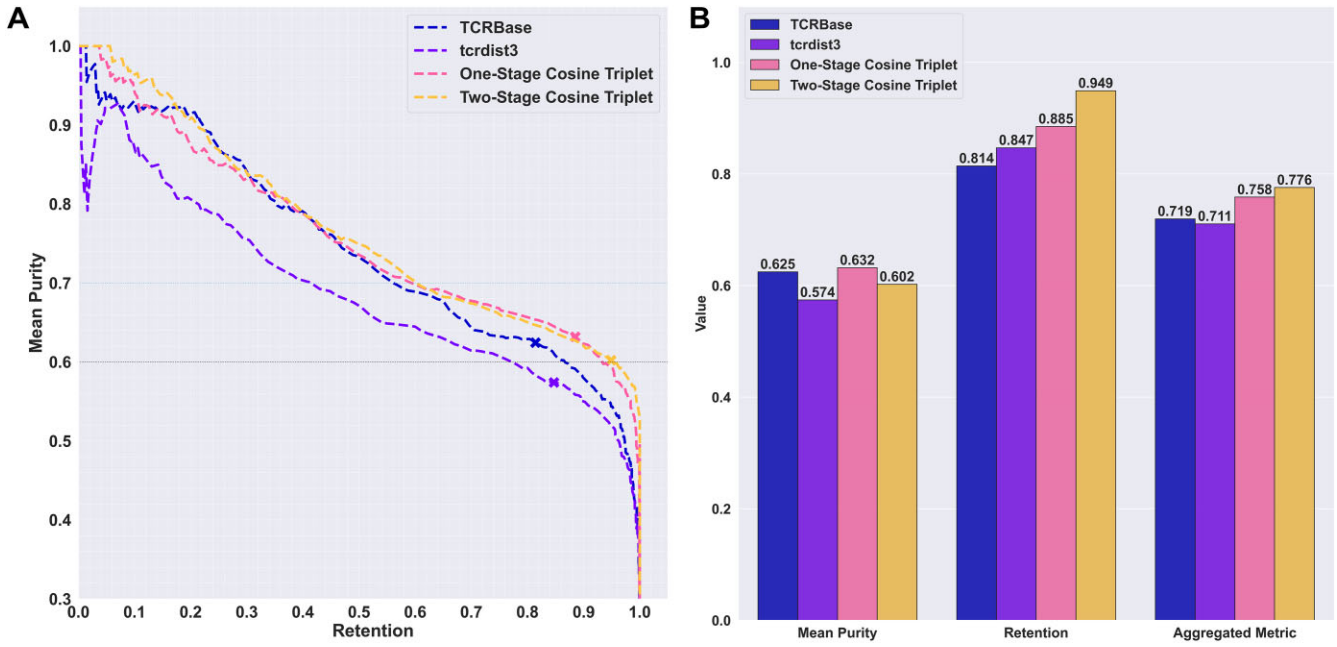


Figure 6. Benchmark of distance matrices on the 17-peptide subset using the agglomerative algorithm. (**A**) Retention–purity curves for each of the methods benchmarked. The markers indicate the optimal SI solution. (**B**) Metrics for the optimal SI solution for each of the distance matrices benchmarked. Aggregated metric is defined as the mean of mean purity and retention.

these experiments confirmed our hypothesis. We found that while both the agglomerative and Top1Cut-weighted methods could achieve perfect clustering results in cases with no or limited noise, they both were challenged when noise (and in particular mislabelling and cross-reactivity) was introduced. In these cases, both methods resulted in solutions with small clusters. It is, however, worth noticing that in these experiments while the Top1Cut-weighted method was demonstrated resistant to normal noise, its clustering performance falls off in a scenario with biological noise (mislabelling, cross-reactivity). In contrast, while the agglomerative method struggles under normal noise, it achieved similar performance in terms of average purity and retention as the Top1Cut-weighted method in the biological setting but with overall larger clusters. These observations are thus in agreement with the findings in Fig. 5, and based on these results, we opted for using the agglomerative method for the remaining part of the analysis.

To further investigate the source of this noise and potential mislabelling, we next investigated properties of the 'best clusters', defined by a minimum purity of 75% and size of 5 or larger, as obtained by the agglomerative clustering with our best VAE model for the 17-peptide test set. This dataset consists of TCRs from three sources, 10x Genomics (filtered with ITRAP-1.0), IEDB, and VDJdb, with a relative proportion of 54.6%, 6.3%, and 39.1%, respectively. Of these, in particular the 10x Genomics data have previously been suggested to share substantial issues with mislabelling and cross-reactivity [7, 9]. Among the 'best clusters', the corresponding proportions of TCRs were 65.3%, 6.9%, and 27.8%, respectively. These numbers are approximately in line with the proportions found across these entire data test set, with a slight enrichment for TCRs originating from 10x data and depletion for VDJdb. Additionally, investigating the annotated pMHC targets for the TCRs in these 'best clusters', we find that these were enriched in specificity towards LLWNGPMAV, GILGFVFTL, and RAKFKQLL and depleted towards KLGGALQAK (see Supplementary Table S1). These findings are in line with the prediction performances previously reported for these same peptides in, for instance, [21], with the enriched peptides sharing a high and the depleted peptide a low predictive performance. However, we observed a surprisingly small proportion of clusters specific to ELAGIGILTV, despite its high performance reported in [21] and high degree of separation in the latent space (see Supplementary Table S1, sheet ELAGIGILTV). We hypothesize that this is likely due to the noise present between TCRs specific to ELAGIGILTV and LLWNGPMAV (see Supplementary Fig. S2). This is further supported by the clusters found for LLWNGPMAV (Supplementary Table S2, Cluster ID 440), where the cluster of size 18 and purity of 78% had four misclustered TCRs, all specific to ELAGIGILTV. While these results suggest that the 'noise' is not particularly driven by one data source, they also illustrate that the noise present is likely non-random and that our models and evaluation results could very likely be improved by use of cleaner data.

Having thus selected an optimal clustering method, we next benchmarked it in the context of the 17-peptide subset in terms of distance matrices generated using the cosine similarities of the latent space of the VAEs trained as one or two stages with triplet loss and the distance matrices generated using TCRbase and tcrdist3. The benchmark was conducted in a similar manner as in Fig. 5 by performing iterative clusterings to generate purity versus retention curves, and indicating the optimal solution as defined from the max SI value. The results of this analysis are included in Fig. 6A and demonstrate that both VAEs learned useful patterns in the latent space, as both outperformed the two sequence-based methods on the aggregated metric. Further, the results support the results from Fig. 4, with a superior performance in terms of both purity and retention of the two-stage model over both the one-stage and the two sequence-based models (Fig. 6B).

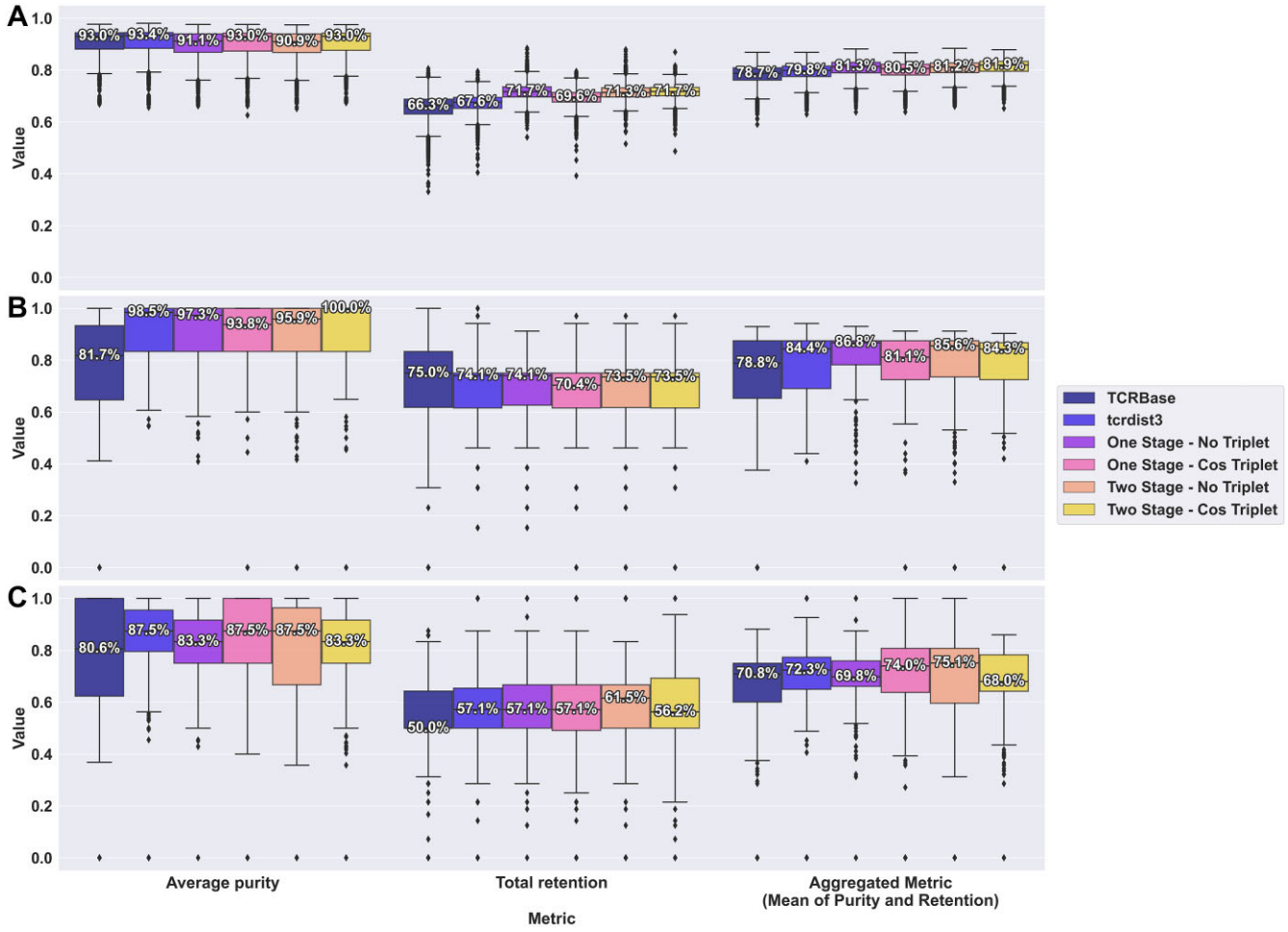


Figure 7. Repertoire analyses. Boxplots representing the bootstrapped distributions of average purity, retention, and aggregated metrics for the target class (excluding TCRs clustered with sequences from the healthy background) across the repertoire datasets. Random samples from a healthy background were combined with disease-specific repertoires in a 1:1 ratio. This process was repeated 100 times with different seeds for bootstrapping. (**A**) Boxplot of clustering metrics for the COVID dataset, consisting of 43 repertoires bootstrapped 100 times, totalling 4300 runs. (**B**) Boxplot of clustering metrics for the cancer dataset for QVDYYGLYY-specific TCRs, consisting of four repertoires bootstrapped 100 times, totalling 400 runs. (**C**) Boxplot of clustering metrics of the cancer dataset for KSAIVTLTY-specific TCRs, consisting of four repertoires each bootstrapped 100 times, totalling 400 runs.

Generalizing to unseen patient repertoires

Building on the insights gained from our clustering evaluation, we next investigated our TCR clustering approach on unseen data beyond the semi-supervised context. To this end, we conducted an analysis on immune repertoires from SARS-CoV-2 [31] and cancer patients [29] incorporating data from multiple studies to evaluate our approach in a real-world scenario. In all cases, we generated a TCR background from healthy control from the OTS database [13], keeping only non-expanded TCRs (i.e. TCRs with a count value of 1) that do not overlap with the diseased repertoires (for details refer to the ''Materials and methods' section). Data from the disease settings were merged with subsampled data from the healthy background with 100 different seeds in a bootstrap framework. This resulted in 4300 mixed repertoires for the COVID dataset and 400 mixed repertoires for the cancer dataset. The agglomerative algorithm was applied to the distance matrices generated by our VAE (trained on the 170-peptide-specific training set) using the cosine of the latent TCR representation, or sequence-based matrices from TCRbase and tcrdist3. The results of this analysis are detailed in Fig. 7, with each panel representing a different dataset: SARS-CoV-2, cancer (QVDYYGLYY-specific), and cancer (KSAIVTLTY-specific), respectively.

Focusing first on the SARS-CoV-2 data (Fig. 7A), all distance matrices exhibited comparable performance in terms of average cluster purity, while the VAE models display an overall higher retention rate. Performing the same cluster size analysis as for Fig. 5, we found a similar trend with respect to iterations and optimal SI here (see Supplementary Fig. S6).

When accounting for both metrics, the two-stage cosine triplet model performs best overall outperforming both TCRbase and TCRdist. This boost in performance for the VAE models can likely at least in part be attributed to the large presence of COVID epitopes and TCRs being present on the training data for the VAE (give some values for the number and proportion of peptides and TCR from COVID in the training data). In summary, in the SARS-CoV-2 setting, we observe a similar trend as in the peptide setting, with performance (represented by the aggregated metric) improving from no semi-supervision (one-stage) towards the two-stage VAE with two levels of semi-supervision. In addition, the mean cluster size at the optimal SI solution was larger on average compared to

the Top1Cut-weighted method (data not shown), further reinforcing our method's choice.

On the other hand, in the cancer setting (Fig. 7B and C), the overall best-performing VAE method was found to be the two-stage model trained without triplet loss. This can likely be explained by the very low similarity of two investigated neo-epitopes KSAIVTLTY and QVDYYGLYY to the epitopes in the training data, the most similar being VAANIVLTV and FTSDYYQLY with mutual peptide Levenshtein distances of 5 and 5, respectively. For more details, refer to Supplementary Fig. S7. However, here the VAE method outperforms TCRbase and performs at least on par with if not better than TCRdist. In the cancer setting, the performance gain by including semi-supervision is not as noticeable, with all VAEs faring similarly with regards to the aggregated metric, however, with the two-stage no-triplet model performing best when accounting for both peptide subsets (KSAIVTLTY and QVDYYGLYY).

One thing to note is the larger spread in performance. We speculate that it can partly be attributed to inherent differences in the distribution of the data. First, there are much fewer unique sequences for each repertoire in the cancer setting versus the SARS-CoV-2 setting. Second, there are fewer repertoires available ($n = 4$) compared to the COVID dataset ($n = 43$). In another analysis, we observed that the variability in clustering metrics came from the sampling of the healthy control ($n = 100$, sampling with no replacement), as well as from between repertoires themselves (data not shown). With only four repertoires and fewer sequences available, the variability from sampling may become more apparent than in the COVID case. Finally, the clustering task itself is more challenging, as a given patient repertoire contains sequences that have been matched in either a tetramer-KSAIVTLTY or tetramer-QVDYYGLYY sample, whereas the COVID setting only had two classes to cluster from.

Post-hoc analysis of COVID repertoire clusters

We speculated that TCRs clustered in COVID repertoires may represent TCR specific to common infections (e.g. influenza, EBV). Also, we wanted to investigate to what degree the identification of COVID-specific clusters were driven by similarities to the data used from training of the VAE mode. To investigate these two points, we conducted a post-hoc analysis on the cluster results from all the COVID repertoires.

Specifically, for each repertoire ($n = 43$), we looked at all repetitions ($s = 100$) and grouped TCRs based on how many times they appeared in a 'best cluster' (defined as purity >75%, size ≥5) out of 100 runs, defining six groups (percentile-based): Group 1 (≤50%), Group 2 ($50\% < p \leq 60\%$), Group 3 ($60\% < p \leq 70\%$), Group 4 ($70\% < p \leq 80\%$), Group 5 ($80\% < p \leq 90\%$), and Group 6 ($90\% < p \leq 100\%$). Then, we defined four reference datasets: COVID-specific TCRs in IEDB, TCR in our training set, expanded TCRs in the healthy repertoires (TCRs with a count value >1, excluded in the sampling process and analysis above), and background TCRs in healthy repertoire (TCRs with a count value = 1, included in the sampling process and analysis above). Next, we compared the TCR from each of the six groups to the different reference datasets to identify to what degree TCRs found in a given cluster group would also be found in each reference dataset. Here, the expectation would be that the overlap between the IEDB (and potentially the training data) should increase as the cluster group level increases, and overlap to the healthy control background remains low. With respect to the set of expanded healthy control TCR, we would expect an increased overlap if the clusters identified in the COVID repertoire were non-COVID-specific (i.e. specific to common background infections), and a constant low overlap if the clusters were COVID-specific. The results of this analysis are summarized in Table 1.

Table 1. Counts and percentage found in the respective databases for each percentile-based cluster groups: Group 1 (≤50%), Group 2 ($50\% < p \leq 60\%$), Group 3 ($60\% < p \leq 70\%$), Group 4 ($70\% < p \leq 80\%$), Group 5 ($80\% < p \leq 90\%$), and Group 6 ($90\% < p \leq 100\%$)

	Count	% IEDB	% Train	% Healthy (background)	% Healthy (expanded)
Group 1	17 262	5.31	3.20	0.00	0.00
Group 2	352	11.36	10.23	0.00	0.00
Group 3	195	11.28	10.26	0.00	0.00
Group 4	108	16.67	14.82	0.00	0.93
Group 5	107	19.63	27.10	0.00	3.74
Group 6	192	21.35	18.75	2.08	2.08

Given that only a minority of the data in the IEDB dataset contained a CDR3 alpha chain (4.74% of $n = 92\,482$), a match was defined using only the CDR3 beta chain for the IEDB and training datasets. A match against the healthy datasets (background and expanded) was defined using a combined match (CDR3 alpha and beta).

Looking first at the distribution of TCRs among the six groups, the majority of sequences fall into Group 1 (17 262 TCRs), indicating that most TCRs appear in a 'good' clusters (e.g. clusters with high purity and adequate size) <50% of the time in repeated sampling. The sequences in this group also show only minimal overlap with the IEDB COVID reference set or the training data (5.31% and 3.20%, respectively), which suggests that a large proportion of these TCRs are not strongly associated with known COVID epitope-specific TCR in our references, at least based on the current clustering criteria. In contrast, TCRs that pass higher thresholds (Groups 2–6) exhibit increasing overlap with IEDB, reflecting a stronger potential link to COVID-specific TCRs. At the same time, this increase is similarly reflected in the training set, suggesting that these TCRs to some degree overlap with the COVID-specific TCRs seen during training. In contrast, TCRs found in good clusters did not appear in either the background or expanded healthy references for the first three clustering levels, and only minimally for levels 4–6 (all values below 4%, and all much below the overlap to the IEDB dataset). Together, these results suggest that the good clusters found in this re-sampling experiment do not simply occur due to artefacts from overlap with the training data or TCRs to other infections, and that they likely reflect TCRs specific to COVID-related epitopes.

The TCRCluster method

The developed VAE model and clustering procedure is made available at a tool called TCRCluster through a web server and stand-alone tool hosted at https://services.healthtech.dtu.dk/services/TCRcluster-1.0/. The data used to train, validate, and test the models, as well as the code used to run the backend and train the VAE models of the web server, can be found at https://github.com/mnielLab/tcrcluster_backend.

Input data must be provided in a comma-separated format, with each data point being the sequence of all six CDR loops (CDR1a, CDR2a, CDR3a, CDR1b, CDR2b, and CDR3b), and optionally a label for each TCR sequence. Here, the CDRs must be defined according to IMGT nomenclature defined ear-

lier, i.e. with CDR1 defined as positions 27–38, CDR2 as positions 56–65, and CDR3 as positions 105–117. The user may choose which of the four VAE models to use and whether to run the optimization process, e.g. picking the optimal cluster solution based on the SI score, or provide a distance threshold with which to cluster. The program first uses the selected VAE model to extract the latent features. Then, the features are converted into a latent cosine distance matrix that serves as input for the clustering part. Finally, if the optimization process was selected, the agglomerative algorithm will be applied with a set of distance thresholds, and the optimal threshold will be selected based on the maximum SI score. If a custom distance threshold is given, the cluster solution for that threshold will be returned.

The program's output provides the extracted latent features with the corresponding predicted cluster labels and distance matrix. If the optimization process was used, clustering metrics at each distance threshold tested in the optimization process can also be downloaded in a summary table. This table includes metrics such as the SI score, the number of clusters, and mean cluster purity, allowing users to evaluate the clustering performance and select alternative solutions if desired. Additionally, users can download visualizations of the optimization curve and other relevant outputs, facilitating an in-depth analysis of the clustering results. These outputs ensure transparency and allow for further downstream analysis, while giving the user the option to re-run a clustering with a manual distance threshold based on the optimization outputs.

Discussion

In this work, we have presented TCRCluster, an ML framework for TCR featurization and clustering. The TCR featurization is performed utilizing VAEs, capable of learning and generating a meaningful latent representation of TCRs by leveraging paired α–β chain data.

We first evaluated the latent space by assessing reconstruction accuracy, achieving up to 99.3% sequence reconstruction on validation and test datasets for our best models using six CDR loops. Fine-tuning hyperparameters, including the use of an annealing schedule for the KLD term, was critical in enhancing reconstruction fidelity. Incorporating tailored loss regimes, such as weighted reconstruction losses for CDR3α and CDR3β, further improved performance. These findings underscore the potential of VAEs to generate latent spaces that faithfully represent TCR sequences.

To enhance the latent space's utility for downstream tasks, we introduced a semi-supervised learning approach, incorporating cosine-distance triplet loss and a two-stage VAE framework. These modifications resulted in improved the latent space representation encoding specificity-related patterns, with improvements reflected in the separation of distance distributions between binders and non-binders. We found that using all six CDR loops, combined with semi-supervised approaches, improved the tractability of the latent space, thus making the latent space representations better suited for prediction and clustering tasks. In particular, employing both a triplet loss and a two-stage model during training boosted the model's ability to generate meaningful latent vector representations. Moreover, they were shown to share enriched information that could be used to compare latent similarities between TCRs, which outperformed traditional sequence-based methods such as tcrdist3 and TCRbase in clustering tasks. While some studies have included HLA-typing into their models [23, 56], others have found limited benefit to adding such an encoding in their models [19, 20]. Furthermore, our experiments with encoding V and J genes categorically did not yield any performance gains (data not shown), and we speculate that categorically encoding HLA would similarly offer little benefit given that a few HLA alleles dominate our training data.

Extending beyond single-peptide evaluations, we next explored clustering methods to group TCRs by antigen specificity. Benchmarking *K*-means, agglomerative clustering, and a novel graph-based Top1Cut algorithm revealed distinct trade-offs between cluster purity and retention. Top1Cut-weighted clustering demonstrated superior retention at higher purity thresholds, although its optimal solutions often comprised small clusters. In contrast, agglomerative clustering achieved biologically relevant groupings, maintaining larger cluster sizes with acceptable purity levels. These outcomes suggest agglomerative clustering's suitability for datasets with inherent biological noise, such as mislabelling or cross-reactivity. Although our VAE-based distance metrics and clustering methods successfully identified enriched clusters for a subset of peptides, the overall cluster sizes remained small, suggesting that noisy or incorrectly annotated TCR data can disrupt otherwise promising results. Collectively, this analysis emphasizes the need for cleaner training data and more robust methods that can mitigate noise effects, ensuring that models can train on and accurately capture the underlying biology of TCR specificity.

Finally, we evaluated our model's ability to generalize to unseen immune repertoires from SARS-CoV-2 and cancer patients. In SARS-CoV-2 data, the two-stage VAE outperformed both TCRbase and tcrdist3, likely benefiting from the overlap of epitopes and TCRs with the training data. In cancer datasets, the VAE models also surpassed TCRbase and performed comparably or better than tcrdist3. Notably, the two-stage model trained without triplet loss performed best in these settings, likely reflecting the unique challenges posed by neo-epitope data. In contrast to the peptide-specific case, the improvement from using semi-supervised models varies for all models across datasets, with no clear pattern emerging in the repertoire task. While the level 2 semi-supervised model showed notable advantages in the first two tasks, its performance in the third generalization task did not show significant improvement.

Despite our semi-supervised strategy, the ability of our combined framework to generalize to unseen patient repertoires underscores its potential for real-world applications in immunotherapy and diagnostics. When applied to datasets from SARS-CoV-2 and cancer repertoires, our models demonstrated robust clustering performance, with the two-stage cosine triplet VAE consistently outperforming baseline methods. However, variability across datasets indicates room for improvement in adapting the latent space to diverse repertoires. This suggests that even though the model was optimized for semi-supervised tasks, it retains a reasonable level of flexibility in adapting to unseen data, reinforcing its value in scenarios requiring both specificity and generalization. Together, these contributions emphasize the need for continued refinement of both latent representation learning and clustering methodologies to address noise, cross-reactivity, and data scarcity. Our results on the COVID dataset further show that our models

gain information through semi-supervision, paving the way for potential model refinement using disease-specific data. By repeatedly clustering 43 repertoires and assigning each TCR to one of six clustering groups each with increasing clustering occurrence likelihoods, we found that the majority of sequences (17 262) in Group 0 (lowest clustering occurrence likelihood) showed little overlap with either the IEDB COVID reference or our training data, implying that the unclustered TCRs likely did not have any specificity for COVID epitopes. In contrast, sequences in groups with higher clustering likelihood demonstrated greater overlap with IEDB, suggesting a stronger potential for COVID-specific TCRs, while remaining mostly absent from both expanded and background healthy references. This pattern indicates that the observed clusters are unlikely to be mere artefacts arising from training data overlap or cross-reactivity with other common infections, but rather reflect an enrichment of TCRs relevant to COVID-related epitopes.

In conclusion, our findings demonstrate the versatility and accuracy of VAEs in accurate featurization and representation of TCRs, highlighting their potential for applications in immune repertoire analysis and antigen-specific clustering and specificity prediction. The semi-supervised enhancements and clustering benchmarks provide a framework for analysing complex immune datasets, with implications for advancing our understanding of immune responses in disease contexts. Future work should focus on mitigating the effects of biological noise and further refining clustering strategies to improve applicability to broader datasets.

Acknowledgements

We thank Simon Rasmussen, Svetlana Kutuzova, and Jakob Nybo Nissen for their expertise and assistance in the development of the VAE models.

Author contributions: Yat-tsai Richie Wan (Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing—original draft, Writing—review & editing) and Morten Nielsen (Funding acquisition, Data curation, Investigation, Project administration, Supervision, Writing—original draft, Writing—review & editing).

Supplementary data

Supplementary data is available at NAR Genomics & Bioinformatics online.

Conflict of interest

None declared.

Funding

Research reported in this publication was supported in part by the National Cancer Institute (NCI), under award number U24CA248138, the National Institute of Allergy and Infectious Diseases (NIAID), under award number 75N93019C00001, and from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement no. 101007799 (Inno4Vac).

Data availability

Data used to train and validate the VAE models are available online at https://services.healthtech.dtu.dk/services/TCRcluster-1.0/. The datasets as well as the code used to train the models are also available at https://github.com/mnielLab/tcrcluster_backend. The datasets and code are also hosted at the following DOIs on Zenodo: https://doi.org/10.5281/zenodo.15350361 and https://doi.org/10.5281/zenodo.15388480, respectively.

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. Garcia KC, Adams EJ. How the T cell receptor sees antigen—a structural view. *Cell* 2005;**122**:333–6. https://doi.org/10.1016/j.cell.2005.07.015
3. Glanville J, Huang H, Nau A *et al.* Identifying specificity groups in the T cell receptor repertoire. *Nature* 2017;**547**:94–8. https://doi.org/10.1038/nature22976
4. Emerson RO, DeWitt WS, Vignali M *et al.* Immunosequencing identifies signatures of cytomegalovirus exposure history and HLA-mediated effects on the T cell repertoire. *Nat Genet* 2017;**49**:659–65. https://doi.org/10.1038/ng.3822
5. Gielis S, Moris P, Bittremieux W *et al.* Detection of enriched T cell epitope specificity in full T cell receptor sequence repertoires. *Front Immunol* 2019;**10**:2820. https://doi.org/10.3389/fimmu.2019.02820
6. 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
7. Montemurro A, Povlsen HR, Jessen LE *et al.* Benchmarking data-driven filtering for denoising of TCRpMHC single-cell data. *Sci Rep* 2023;**13**:16147. https://doi.org/10.1038/s41598-023-43048-3
8. Zhang W, Hawkins PG, He J *et al.* A framework for highly multiplexed dextramer mapping and prediction of T cell receptor sequences to antigen specificity. *Sci Adv* 2021;**7**:eabf5835. https://doi.org/10.1126/sciadv.abf5835
9. Povlsen HR, Bentzen AK, Kadivar M *et al.* Improved T cell receptor antigen pairing through data-driven filtering of sequencing information from single cells. *eLife* 2023;**12**:e81810. https://doi.org/10.7554/eLife.81810
10. 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
11. 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
12. 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
13. Raybould MIJ, Greenshields-Watson A, Agarwal P *et al.* The observed T cell receptor space database enables paired-chain repertoire mining, coherence analysis and language modelling. *Cell Rep* 2024;**43**:114704. https://doi.org/10.1101/2024.05.20.594960
14. Chen S-Y, Yue T, Lei Q *et al.* TCRdb: a comprehensive database for T-cell receptor sequences with powerful search function. *Nucleic Acids Res* 2021;**49**:D468–74. https://doi.org/10.1093/nar/gkaa796

15. 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
16. Huang H, Wang C, Rubelt F *et al.* Analyzing the *Mycobacterium tuberculosis* immune response by T-cell receptor clustering with GLIPH2 and genome-wide antigen screening. *Nat Biotechnol* 2020;**38**:1194–202. https://doi.org/10.1038/s41587-020-0505-4
17. Mayer-Blackwell K, Schattgen S, Cohen-Lavi L *et al.* TCR meta-clonotypes 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
18. Chronister WD, Crinklaw A, Mahajan S *et al.* TCRMatch: predicting T-cell receptor specificity based on sequence similarity to previously characterized receptors. *Front Immunol* 2021;**12**:640725. https://doi.org/10.3389/fimmu.2021.640725
19. Montemurro A, Jessen LE, Nielsen M. NetTCR-2.1: lessons and guidance on how to develop models for TCR specificity predictions. *Front Immunol* 2022;**13**:1055151. https://doi.org/10.3389/fimmu.2022.1055151
20. 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
21. Jensen MF, Nielsen M. Enhancing TCR specificity predictions by combined pan- and peptide-specific training, loss-scaling, and sequence similarity integration. *eLife* 2024;**12**:RP93934. https://doi.org/10.7554/eLife.93934
22. Sidhom J-W, Larman HB, 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
23. Meynard-Piganeau B, Feinauer C, Weigt M *et al.* TULIP: a transformer-based unsupervised language model for interacting peptides and T cell receptors that generalizes to unseen epitopes. *Proc Natl Acad Sci USA* 2024;**121**:e2316401121. https://doi.org/10.1073/pnas.2316401121
24. Myronov A, Mazzocco G, Król P *et al.* BERTrand-peptide: TCR binding prediction using bidirectional encoder representations from transformers augmented with random TCR pairing. *Bioinformatics* 2023;**39**:btad468. https://doi.org/10.1093/bioinformatics/btad468
25. Nielsen M, Eugster A, Jensen MF *et al.* Lessons learned from the IMMREP23 TCR-epitope prediction challenge. *ImmunoInformatics* 2024;**16**:100045. https://doi.org/10.1016/j.immuno.2024.100045
26. Girvan M, Newman MEJ. Community structure in social and biological networks. *Proc Natl Acad Sci USA* 2002;**99**:7821–6. https://doi.org/10.1073/pnas.122653799
27. Heather JM, Spindler MJ, Alonso MH *et al.* Stitchr: stitching coding TCR nucleotide sequences from V/J/CDR3 information. *Nucleic Acids Res* 2022;**50**:e68. https://doi.org/10.1093/nar/gkac190
28. Dunbar J, Deane CM. ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics* 2016;**32**:298–300. https://doi.org/10.1093/bioinformatics/btv552
29. Eberhardt CS, Kissick HT, Patel MR *et al.* Functional HPV-specific PD-1$^+$ stem-like CD8 T cells in head and neck cancer. *Nature* 2021;**597**:279–84. https://doi.org/10.1038/s41586-021-03862-z
30. Garner LC, Amini A, FitzPatrick MEB *et al.* Single-cell analysis of human MAIT cell transcriptional, functional and clonal diversity. *Nat Immunol* 2023;**24**:1565–78. https://doi.org/10.1038/s41590-023-01575-1
31. Francis JM, Leistritz-Edwards D, Dunn A *et al.* Allelic variation in class I HLA determines CD8$^+$ T cell repertoire shape and cross-reactive memory responses to SARS-CoV-2. *Sci Immunol* 2022;**7**:eabk3070.
32. Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv, https://arxiv.org/abs/1312.6114, 20 December 2013, preprint: not peer reviewed.
33. Gulrajani I, Kumar K, Ahmed F *et al.* PixelVAE: A latent variable model for natural images. arXiv, https://arxiv.org/abs/1611.05013, 15 November 2016, preprint: not peer reviewed.
34. Semeniuta S, Severyn A, Barth E. A hybrid convolutional variational autoencoder for text generation. arXiv, https://arxiv.org/abs/1702.02390, 8 February 2017, preprint: not peer reviewed.
35. Pu Y, Gan Z, Henao R *et al.* Variational autoencoder for deep learning of images, labels and captions. arXiv, https://arxiv.org/abs/1609.08976, 28 September 2016, preprint: not peer reviewed.
36. LeCun Y, Boser B, Denker JS *et al.* Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1989;**1**:541–51. https://doi.org/10.1162/neco.1989.1.4.541
37. Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning. arXiv, https://arxiv.org/abs/1603.07285, 11 January 2018, preprint: not peer reviewed.
38. Higgins I, Matthey L, Pal A *et al.* beta-VAE: learning basic visual concepts with a constrained variational framework. In: *International Conference on Learning Representations*. Toulon, France, 2017. https://openreview.net/forum?id=Sy2fzU9gl
39. Khosla P, Teterwak P, Wang C *et al.* Supervised contrastive learning. arXiv, https://arxiv.org/abs/2004.11362, 10 March 2021, preprint: not peer reviewed.
40. Pedregosa F, Varoquaux G, Gramfort A *et al.* Scikit-learn: machine learning in Python. arXiv, https://arxiv.org/abs/1201.0490, 5 June 2018, preprint: not peer reviewed.
41. Kruskal JB. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc Am Math Soc* 1956;**7**:48–50. https://doi.org/10.1090/S0002-9939-1956-0078686-7
42. Danska JS, Livingstone AM, Paragas V *et al.* The presumptive CDR3 regions of both T cell receptor alpha and beta chains determine T cell specificity for myoglobin peptides. *J Exp Med* 1990;**172**:27–33. https://doi.org/10.1084/jem.172.1.27
43. Reinherz EL, Tan K, Tang L *et al.* The crystal structure of a T cell receptor in complex with peptide and MHC class II. *Science* 1999;**286**:1913–21. https://doi.org/10.1126/science.286.5446.1913
44. Reiser J-B, Grégoire C, Darnault C *et al.* A T cell receptor CDR3β loop undergoes conformational changes of unprecedented magnitude upon binding to a peptide/MHC class I complex. *Immunity* 2002;**16**:345–54. https://doi.org/10.1016/S1074-7613(02)00288-1
45. Jones LL, Colf LA, Stone JD *et al.* Distinct CDR3 conformations in TCRs determine the level of cross-reactivity for diverse antigens, but not the docking orientation. *J Immunol* 2008;**181**:6255–64. https://doi.org/10.4049/jimmunol.181.9.6255
46. Leary AY, Scott D, Gupta NT *et al.* Designing meaningful continuous representations of T cell receptor sequences with deep generative models. *Nat Commun* 2024;**15**:4271. https://doi.org/10.1038/s41467-024-48198-0
47. Croce G, Bobisse S, Moreno DL *et al.* Deep learning predictions of TCR-epitope interactions reveal epitope-specific chains in dual alpha T cells. *Nat Commun* 2024;**15**:3211. https://doi.org/10.1038/s41467-024-47461-8
48. Burgess CP, Higgins I, Pal A *et al.* Understanding disentangling in β-VAE. arXiv, https://arxiv.org/abs/1804.03599, 10 April 2018, preprint: not peer reviewed.
49. McClish DK. Analyzing a portion of the ROC curve. *Med Decis Making* 1989;**9**:190–5. https://doi.org/10.1177/0272989X8900900307
50. 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

51. Wang H, Hernandez JM, Van Mieghem P. Betweenness centrality in a weighted network. *Phys Rev E Stat Nonlin Soft Matter Phys* 2008;**77**:046105. https://doi.org/10.1103/PhysRevE.77.046105
52. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65. https://doi.org/10.1016/0377-0427(87)90125-7
53. Pavlopoulos J, Vardakas G, Likas A. Revisiting silhouette aggregation. arXiv, https://arxiv.org/abs/2401.05831, 22 June 2024, preprint: not peer reviewed.
54. 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
55. Kobak D, Linderman GC. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nat Biotechnol* 2021;**39**:156–7. https://doi.org/10.1038/s41587-020-00809-z
56. 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

Received: January 24, 2025. **Revised:** May 7, 2025. **Editorial Decision:** May 9, 2025. **Accepted:** May 13, 2025