A comparison of clustering models for inference of T cell receptor antigen specificity.

Dan Hudson[1,2], Alex Lubbock[2], Mark Basham[2], Hashem Koohy[1,3,4*]

1 MRC Human Immunology Unit, MRC Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, UK.
2 The Rosalind Franklin Institute, Didcot, UK.
3 Centre for Computational Biology, MRC Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, UK.
4 Alan Turning Fellow in Health and Medicine.

* *Corresponding author: hashem.koohy@rdm.ox.ac.uk*

Abstract

The vast potential sequence diversity of TCRs and their ligands has presented an historic barrier to computational prediction of TCR epitope specificity, a holy grail of quantitative immunology. One common approach is to cluster sequences together, on the assumption that similar receptors bind similar epitopes. Here, we provide an independent evaluation of widely used clustering algorithms for TCR specificity inference, observing some variability in predictive performance between models, and marked differences in scalability. Despite these differences, we find that different algorithms produce clusters with high degrees of similarity for receptors recognising the same epitope. Our analysis highlights an unmet need for improvement of complex models over a simple Hamming distance comparator, and strengthens the case for use of clustering models in TCR specificity inference.

Introduction

T lymphocytes recognise peptide epitopes presented at the cell surface by Major Histocompatibility Complexes (MHC) in jawed vertebrates [1]. Recognition is mediated by diverse heterodimeric α or β TCR domains positioned on the T cell surface. The chains of the more common $\alpha\beta$ TCR contain variable (V), joining (J) gene segments, constant (C) regions, and an additional diversity (D) segment in the β polypeptide. Each T cell expresses many copies of a single TCR, which bind to peptide-MHC (pMHC) via the complementarity determining regions (CDR) 1-3 of the TCR [2]. Productive TCR engagement triggers a context-dependent signalling cascade, which in turn promotes activation and differentiation of diverse immune effector cells [3].

The central role of the TCR in immune surveillance and response to disease has encouraged efforts to decode the rules of TCR-pMHC binding. Determination of specificity, or "receptor de-orphanisation", can be achieved experimentally using sequencing and repertoire analysis, or with functional, multimer binding, or TCR screening methods, reviewed in [4], [5]. However, the ability to accurately predict the cognate epitope of any TCR *in silico* could vastly accelerate our understanding of fundamental and translational T cell biology [6].

The availability of large repositories of TCR sequences and their known ligands has enabled the development of two major families of computational model for prediction of

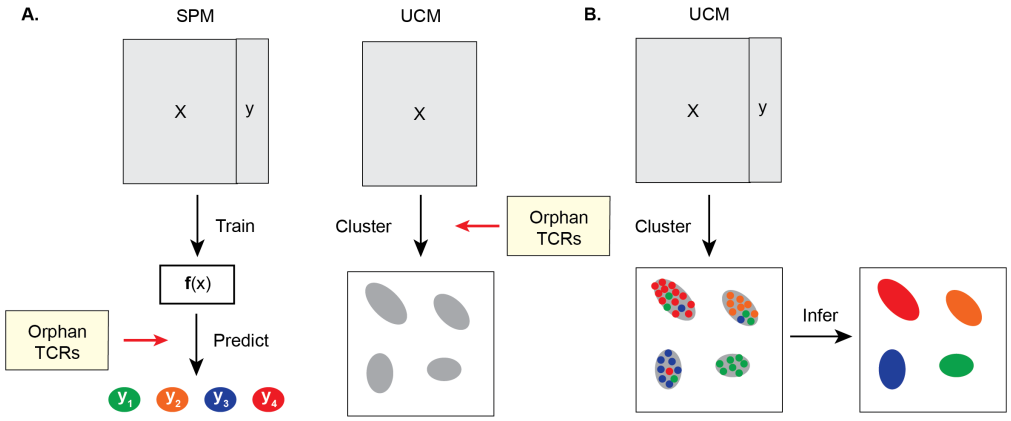

Figure 1. Supervised and unsupervised learning in T cell epitope specificity inference. A) SPMs (*left*) fit a predictive function $\mathbf{f}(x)$ to training data having an independent variable X (TCR sequences and other features) and dependent variable y (epitopes or pMHC complexes). This function may then be applied to predict the cognate epitopes of orphan TCRs. UCMs (*right*) generate a mapping from TCR sequences to a cluster allocation, such that each TCR is assigned to one or more clusters having common epitope specificity. B) When applied to datasets including full or partial epitope labels, UCMs may be used to predict TCR epitope specificity by assigning the most frequent epitope of a cluster as the predicted binder for all TCRs in that cluster.

TCR antigen specificity: Supervised Predictive Models (SPMs) and Unsupervised Clustering Models (UCMs) (**Fig. 1**)[6]. These families are representative of two distinct approaches to machine learning. In *supervised learning*, predictive models are trained on a set of input instances having a known label (in this case, the cognate epitope for a given TCR). In *unsupervised learning*, models learn the underlying statistical features or patterns of a dataset to differentiate between input TCRs, applying techniques such as clustering or dimensionality reduction.

The use of deep neural networks (DNNs) including large language models and convolutional neural networks has contributed significantly to recent improvements in UCM and SPM performance [7]–[11]. Despite these advances, no publicly available SPM is yet capable of accurately predicting the specificity of TCRs recognising "unseen" epitopes that were not encountered during model training [8], [12]. This is likely due at least in part to the limited volume of experimentally determined receptor-epitope pairs, which constitutes just a small fraction of the vast theoretical diversity of TCRs [6], [13].

Unlike SPMs, UCMs do not require receptor-ligand pairs as an input, but group similar TCRs together on the assumption that receptors having similar sequences will bind similar epitopes [14], [15]. UCMs can therefore be applied to identify clusters of similar TCRs irrespective of whether their cognate pMHC has been observed before. This is of particular use in an era when bulk and single-cell sequencing experiments can yield thousands of unique TCRs per sample, by applying UCMs to shortlist TCRs of interest for later experimental de-orphanisation. Such approaches have been successfully applied to identify and characterise TCRs associated with mycobacterial and viral infection, cancer, and autoimmune disease [15]–[20].

UCMs take as their input single or paired TCR CDR3 nucleotide or amino acid sequences, with or without V and J gene usage information, and return a mapping of sequences to unique clusters. This has historically been achieved using some form of distance measure, typically either direct sequence similarity and/ or the frequency

enrichment of short sequence snippets (*kmers*) compared to a reference dataset. Recent approaches leverage DNNs to generate a compressed numeric representation of the input TCR as a precursor to clustering [11], [21]–[23]. We dub such models DNN-UCMs to differentiate them from traditional distance-based UCMs such as GLIPH [15] and tcrdist [14], which were published simultaneously in 2017. A critical advantage of these "traditional" UCMs is that they do not require large volumes of training data, a hallmark and limitation of DNNs.

When some or all of the cognate epitopes of a given TCR dataset are known, UCMs can theoretically be used to infer epitope specificity (**Fig. 1B**). However, there has to date been no independent benchmarking study of UCMs as predictors of TCR specificity, despite their widespread use in field. In the present work, we compare the predictive performance of five commonly used UCMs on sets of known TCR-epitope pairs. We then extend our analysis to qualitative comparison of the clusters formed, practical considerations including runtime speed, and finally the impact on inference of introducing noise from synthetic background TCRs.

Results

Benchmarking analyses were performed on paired $\alpha\beta$ TCRs data drawn from VDJdb, a large, public, curated source of TCRs of known epitope specificity [24]. Model performance was analysed for data subsets generated by retaining epitopes having 10, 50, 100, 500, or 1000 cognate TCRs (datasets V10, V50, V100, V500, and V1000 respectively, **Table S1**). Instances were randomly down sampled after pre-processing, such that each experimental run was performed on the same number of TCR sequences per epitope, and all models were applied to α or β chain selections from the same set of paired TCRs. Sampling was repeated to account for sample variance between epitopes (**Table S2**). We present performance on α or β chain selections independently (see **Discussion** and **Limitations**).

Five open-source models were identified from the literature for which a python implementation was readily available: ClusTCR, GIANA, GLIPH2, iSMART, and tcrdist3 [17], [20], [25]–[27]. Three baseline models were added: A Hamming distance model that grouped together sequences having identical length and differing by not more than one amino acid; a CDR3 length-based model, and a random baseline. Details of model implementations are provided in **Methods**, and a summary of the respective methodologies in **Section S1**. As tcrdist3 generates a distance measure but does not explicitly cluster instances, a scikit-learn implementation of DBSCAN [28] was used to group distance matrices produced with tcrdist3, consistent with [25] and following comparison of model performance with different model implementations and clustering approaches (**Fig. S1**). Hamming, GLIPH2, tcrdist3, and iSMART implementations were adapted from the ClusTCR python package [25] and run using default parameters. The comparative analytical framework and model datasets are made freely available at *https://github.com/hudsondan/tcr-scapes*.

UCMs have historically been evaluated using cluster quality metrics such as purity, consistency, diversity, and retention (detailed in **Methods**). By applying each model to sets of TCRs having known specificity, we were able to combine these metrics with direct measures of predictive capacity including accuracy, precision, recall, and F1-score, following the schema depicted in **Fig. 1B**. Observing a positive correlation between performance according to cluster purity, consistency, adjusted mutual information, precision, recall, and F1-score (**Fig. S2**), we present results for F1-score alone (see **Supplementary Tables**), weighting F1-scores to account for model-specific class imbalance.

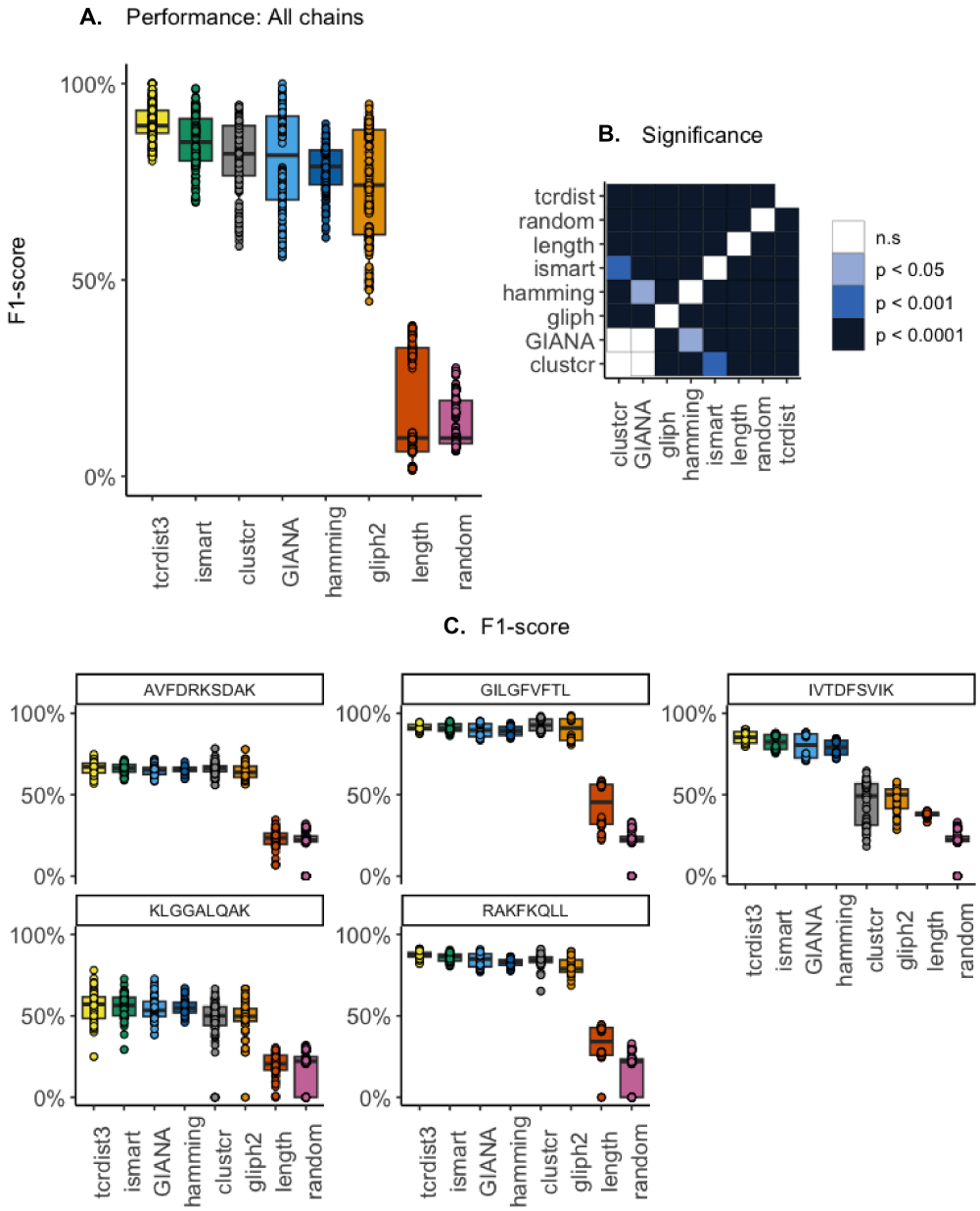

Figure 2. A-B) Comparative model performance for datasets V10-V1000, α and β chain selections combined. A) Predictive performance; B) Significance (p) values of comparisons between models following one-way ANOVA with *post hoc* Tukey's HSD test; C) F1-score per epitope, dataset V500, α and β chain selections combined

Differential model performance was first inspected at a global level for datasets V10 to V1000 combined, grouping over α and β chain selections and over all epitopes (**Fig. 2A-B** and **Table S3**). All study models outperformed length and random baselines ($p<0.0001$). However, whilst tcrdist3 generally performed well, absolute differences between many UCMs and a simple Hamming models were minimal (1-3% when accounting for 95% confidence intervals) (**Table S3**). Despite the apparent statistical

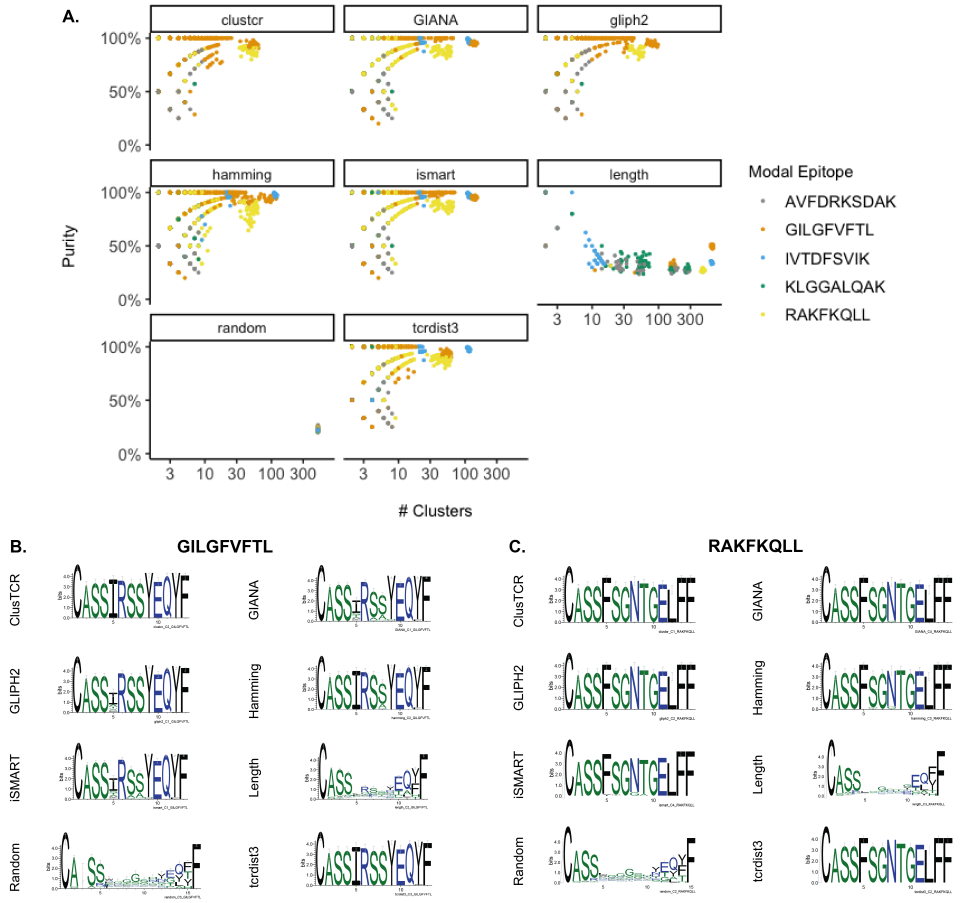

Figure 3. Qualitative analysis of UCM model outputs, for dataset V500 (β chain selection). A): Cluster purity and size distributions by modal epitope. B-C): Sequence logos for the largest clusters produced by a given model for a given epitope. B): GILGFVFTL; C): RAKFKQLL. Logos were produced with WebLogo [29] for TCRs in the largest cluster produced for a given model following sequence alignment with MUSCLE [30].

significance of many comparisons between models (**Fig. 2B**), the relative performance of each model was sensitive to the TCR chain selection (**Fig. S3** and **Table S3**). Furthermore, rankings did not hold across epitopes (**Fig. 2C** and **Table S4**), nor when applied to a discrete set of 509 TCRs drawn from McPas-TCR (**Fig. S4** and **Table S5**).

An inspection of the size and purity of clusters generated by each model revealed high levels of similarity in the patterns produced by ClusTCR, GIANA, GLIPH2, iSMART, tcrdist3, and a Hamming model, which were in turn strikingly different from those produced by CDR3 length and random models (**Fig. 3A**). Notably, these six models produced large, pure clusters of thirty or more members in which the most frequent epitope was GILGFVFTL (Influenza A M-Protein) or RAKFKQLL (EBV BLZF1). β chain CDR3 motifs for the largest clusters associated with each of these epitopes were also near-identical except for length and random baseline models (**Fig. 3B**). Clusters in which AVFDRKSDAK (EBV EBNA-4), IVTDFSVIK (EBV EBNA-4), and KLQQALQAK (hCMV IE1) was the most common epitope were rarer, smaller, and

less pure, producing less consistent CDR3 motifs (**Fig. S5**).

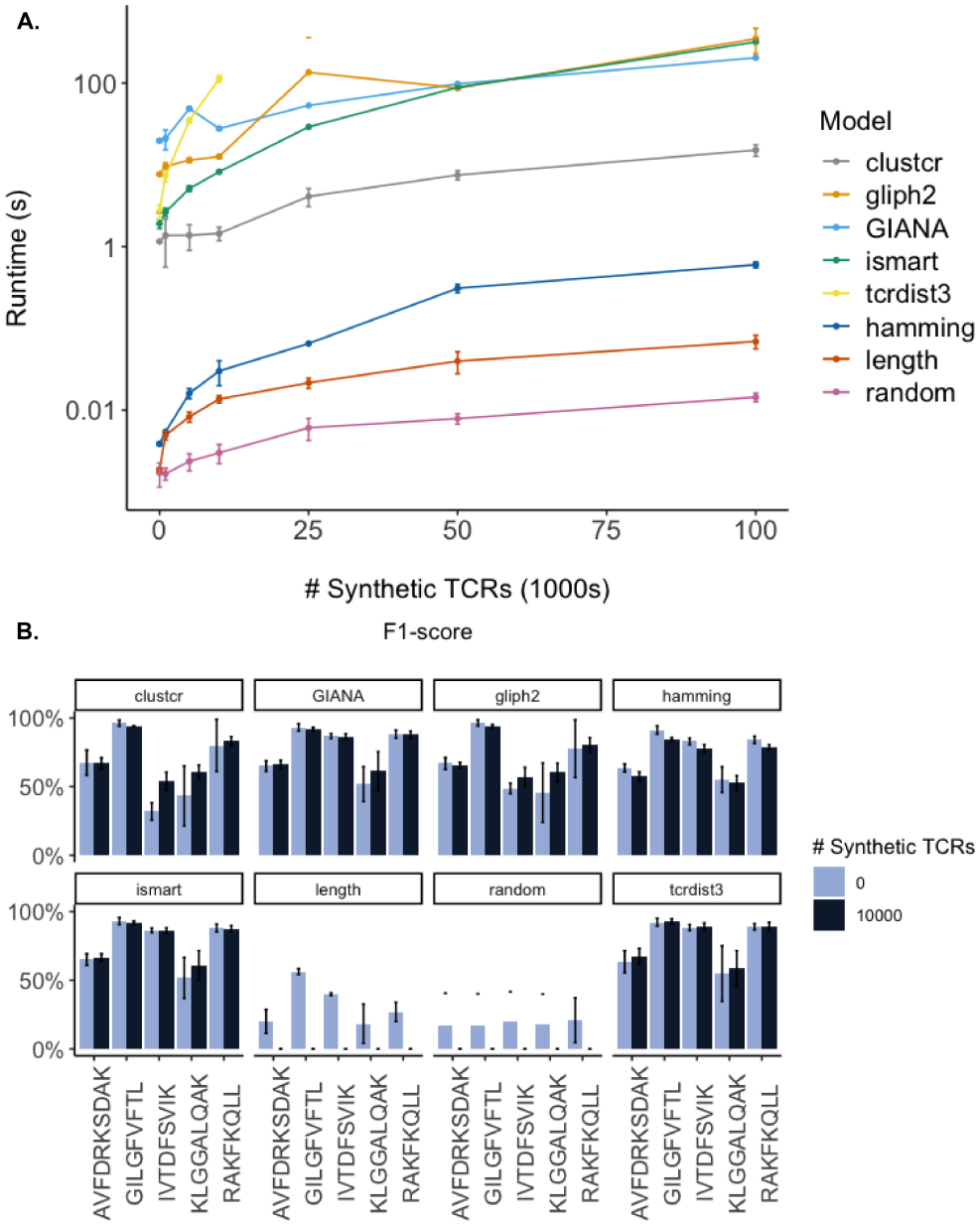

Figure 4. Investigating model scalability, comparing model runtimes as a function of the number of synthetic TCR sequences introduced with OLGA (Sethna et al., 2019). A) Runtimes. B): Epitope-specific F1-scores in the presence or absence of 10,000 synthetic TCR sequences. All experiments conducted on dataset V500 (β chain selection, 5 repeats).

We next compared UCM computational speed by adding synthetic TCRs, produced with OLGA [31], to our input data (**Fig. 4A**). Here, the Hamming distance comparator outstripped the other models by a significant margin, slower only than length and

random baselines. tcrdist3 scaled poorly: with memory constraints preventing use with repertoires greater than 10,000 TCRs for a single CPU and the majority of runtime being contributed by the computation of TCR distances (**Fig. S6**). Predictive performance was not materially impacted by the addition of 10,000 synthetic TCR sequences (**Fig. 4B** and **Table S6**). Taken together, these results suggest firstly that the variable performance gains observed for some models over a Hamming distance comparator come at the cost of a significant increase in runtime, at least when deployed over a single CPU. Secondly, all models were able to infer TCR epitope specificity materially better than a random comparator, even when labelled TCRs were diluted with synthetic TCRs at a ratio of 3:1.

Discussion

Despite the exponential growth of orphan TCR datasets, and the widespread use of UCMs in de-orphanisation pipelines, an independent comparison off their predictive capacity has thus far been missing from the field. Here, we present a first, modest attempt to address this need. Our findings suggest that five commonly used UCMs show some variation in their ability to infer the specificity of a given set of TCRs, but that the best-performing models produce similar clusters to a simple Hamming distance model at considerably slower speeds. We first explore the possible factors underlying the observed performance differences, before reviewing the implications of these results for the challenging task of TCR specificity inference.

ClusTCR, GIANA, GLIPH2, iSMART and tcrdist3 consistently outperformed length and random baselines, however the relative rankings were sensitive to the chain selection, epitope, and dataset used (**Fig. 2, Fig. S3-4**). Intriguingly, global performance was generally within 1-3% of a simple Hamming distance model when accounting for variance (**Table S3**). One exception was tcrdist3, which achieved a 5% improvement in mean F1-score over the next best model when grouping analyses across datasets, epitopes and chain selections (**Table S3**). However, we cannot rule out that the observed performance gain for tcrdist is a product of the pre-processing and sampling strategy used, and we encourage further independent comparisons on other datasets. If real, one possible explanation of this improved performance is the use by tcrdist of inferred sequences for germline-encoded CDR1, CDR2, and pMHC-facing CDR2.5 regions of the TCR, instead of categorical representations of the corresponding gene code [14]. Indeed, a simple predictive model combining tcrdist and a K-nearest neighbour model achieved superior performance to many DNN-SPMs using gene codes in a recent benchmarking exercise [12].

Although recent structural [32], statistical [33], and predictive [9] analyses suggest that both polypeptide chains play an important role in epitope recognition, we observed consistently lower F1-scores for α compared to β chain selections (**Table S3**). One possible explanation is that the default model hyperparameters have been optimised for β chain data, which make up the majority of published TCR-epitope pairs [8]. Alternatively, the β chain may simply contribute more to determination of overall epitope specificity, as a product of its increased diversity relative to α chains, however this warrants further investigation. For example, modelling strategies that permit integration of α andβ chain pairing with transcriptomic and phenotypic information, including graph network approaches such as CoNGA [34], may help efforts to decode the relative contribution of chain pairing to epitope specificity at single cell resolution.

If model performance is sensitive to the choice of dataset and pre-processing strategy, and the five UCMs produce similar clusters to a Hamming distance model, how then can one decide which UCM to use for analysis and/or co-clustering of large TCR sequence datasets? One important lens is scalability, but here again a Hamming distance model

performed best, apart from length and random baselines (**Fig. 4A**). Another consideration is accessibility. We note for example that at present only GLIPH2 is currently available as both a web tool and command line executable, the other models requiring some familiarity with programming languages such as Python and R.

Finally, what do these results tell us about the relative capacity of UCMs for inference of orphan TCR specificity? Combining labelled instances and synthetic sequences produced with OLGA (**Fig 4B** and **Table S5**) provides a window into model performance in co-clustering of reference and orphan TCRs. Encouragingly, we see that UCMs are able to successfully cluster TCRs of common specificity in the presence of a synthetic background of 10,000 TCRs. Our results therefore support the continued use of these models in de-orphanisation pipelines, by co-clustering labelled and unlabelled TCRs, an approach that could theoretically be applied to both seen and unseen epitopes.

Study Limitations

The significant scientific and economic potential of a generalisable solution to prediction of TCR epitope specificity has encouraged the development of a multitude of new SPMs and UCMs, summarised in [6]. However, the scope of the present study is limited to a handful of commonly used UCMs, on the basis of their widespread use and relative freedom from training data bias as compared to DNN-UCMs and SPMs. Nonetheless, an independent comparison of UCMs, DNN-UCMs and DNN-SPMs would be of great use to the community. There is also growing evidence that inclusion of both α and β chains improves predictive performance in SPMs and DNN-SPMs. However, whilst ClusTCR, tcrdist3, and GLIPH2 may all theoretically be applied to paired chain data, cluster assignments are produced for a given CDR3 independent of the other chain for all but tcrdist3. An investigation of whether the integration of α and β chain information improves performance equally across models might reveal the relative merits of each, when applied to large scale single-cell experiments. Time and technical limitations prevented extension of the present analysis to comparative performance over parallel CPUs, and to the GPU-enabled versions of ClusTCR and GIANA. Finally, whilst we have made efforts to investigate relative model performance under a variety of pre-processing conditions, predictive power is sensitive to the dataset and pre-processing methodology. Therefore, an extension to other public datasets, as well as to complete repertoire data from large population studies such as [35], would add certainty to the conclusions drawn.

Acknowledgments

Our comparative framework used borrowed heavily from that developed by the Meysman group for ClusTCR (https://github.com/svalkiers/clusTCR), to whom the authors express their gratitude. Thanks are also due to Dr Ricardo A. Fernandes and Sam Farrar for critical review. H.K. is supported by funding from the UK Medical Research Council grant number MC_UU_12010/3. D.H. receives administrative and financial support from the Biotechnology and Biological Sciences Research Council (BBSRC) (grant number BB_T008784_1) and from the Rosalind Franklin Institute.

Ethics statement

D.H. provides consultancy services to companies active in T cell antigen discovery and vaccine development. The other authors declare no competing interests.

References

[1] M. M. Davis and P. J. Bjorkman, "T-cell antigen receptor genes and t-cell recognition," *Nature*, vol. 334, no. 6181, pp. 395–402, 1988.

[2] R. Bosselut, "T cell antigen recognition: Evolution-driven affinities," *PNAS*, vol. 116, no. 44, pp. 21 969–21 971, 2019.

[3] G. D. Sckisel, M. N. Bouchlaka, A. M. Monjazeb, *et al.*, "Out-of-sequence signal 3 paralyzes primary cd4(+) t-cell-dependent immunity.," *Immunity*, vol. 43, no. 2, pp. 240–250, 2015.

[4] A. V. Joglekar and G. Li, "T cell antigen discovery," *Nat. Methods*, vol. 18, no. 8, pp. 873–880, 2021.

[5] S. Valkiers, N. de Vrij, S. Gielis, *et al.*, "Recent advances in t-cell receptor repertoire analysis: Bridging the gap with multimodal single-cell rna sequencing," *Immunoinformatics*, vol. 5, p. 100 009, 2022.

[6] D. Hudson, R. A. Fernandes, M. Basham, G. Ogg, and H. Koohy, "Can we predict t cell specificity with digital biology and machine learning?" *Nat. Rev. Immunol.*, 2023.

[7] A. Weber, J. Born, and M. a. Rodriguez Martínez, "Titan: T cell receptor specificity prediction with bimodal attention networks," *Bioinformatics*, vol. 37, no. S1, pp. I237–I244, 2021.

[8] P. Moris, A. Postovskaya, S. Gielis, *et al.*, "Current challenges for unseen-epitope tcr interaction prediction and a new perspective derived from image classification," *Brief. Bioinformatics*, vol. 22, no. 4, pp. 1–12, 2021.

[9] A. Montemurro, V. Schuster, H. R. Povlsen, *et al.*, "Nettcr-2.0 enables accurate prediction of tcr-peptide binding by using paired alpha and beta sequence data," *Nat. Commun. Bio*, vol. 4, no. 1, 2021.

[10] K. Wu, K. E. Yost, B. Daniel, *et al.*, "Tcr-bert: Learning the grammar of t-cell receptors for flexible antigen-xbinding analyses," *Preprint at https://www.biorxiv.org/content/10.1101/2021.11.18.469186v1*, 2021.

[11] W. Zhang, P. G. Hawkins, J. He, *et al.*, "A framework for highly multiplexed dextramer mapping and prediction of t cell receptor sequences to antigen specificity," *Sci Adv*, vol. 7, no. 20, eabf5835, 2021.

[12] P. Meysman, J. Barton, B. Bravi, *et al.*, "Benchmarking solutions to the t-cell receptor epitope prediction problem: Immrep22 workshop report," *ImmunoInformatics*, vol. 9, p. 100 024, 2023.

[13] T. P. Arstila, A. Casrouge, V. Baron, J. Even, J. Kanellopoulos, and P. a. Kourilsky, "A direct estimate of the human alphabeta t cell receptor diversity," *Science*, vol. 286, pp. 958–961, 1999.

[14] P. Dash, A. J. Fiore-Gartland, T. Hertz, *et al.*, "Quantifiable predictive features define epitope-specific t cell receptor repertoires," *Nature*, vol. 547, no. 7661, pp. 89–93, 2017.

[15] J. Glanville, H. Huang, A. Nau, *et al.*, "Identifying specificity groups in the t cell receptor repertoire," *Nature*, vol. 547, no. 7661, pp. 94–98, 2017.

[16] F. Hayashi, N. Isobe, J. Glanville, *et al.*, "A new clustering method identifies multiple sclerosis-specific t-cell receptors," *Ann Clin Transl Neurol*, vol. 8, no. 1, pp. 163–176, 2021.

[17] H. Huang, C. Wang, F. Rubelt, T. J. Scriba, and M. M. a. Davis, "Analyzing the mycobacterium tuberculosis immune response by t-cell receptor clustering with gliph2 and genome-wide antigen screening," *Nat Biotechnol*, vol. 38, pp. 1194–1202, 2020.

[18] M. V. Pogorelyy, E. Rosati, A. A. Minervina, *et al.*, "Resolving sars-cov-2 cd4+ t cell specificity via reverse epitope discovery," *Cell. Rep. Med.*, vol. 3, no. 8, p. 100 697, 2022.

[19] Y. Wang, F. Duan, Z. Zhu, *et al.*, "Analysis of tcr repertoire by high-throughput sequencing indicates the feature of t cell immune response after sars-cov-2 infection," *Cells*, vol. 11, no. 1, p. 68, 2021.

[20] H. Zhang, L. Liu, J. Zhang, *et al.*, "Investigation of antigen-specific t-cell receptor clusters in human cancers," *Clin. Canc. Res.*, vol. 26, no. 6, pp. 1359–1371, 2020.

[21] J.-W. Sidhom, H. B. Larman, D. M. Pardoll, and A. S. Baras, "Deeptcr is a deep learning framework for revealing sequence concepts within t-cell repertoires," *Nat. Commun.*, vol. 12, no. 1, 2021.

[22] I. Springer, N. Tickotsky, and Y. a. Louzoun, "Contribution of t cell receptor alpha and beta cdr3, mhc typing, v and j genes to peptide binding prediction," *Front. Immunol*, vol. 12, p. 1436, 2021.

[23] F. Drost, Y. An, L. M. Dratva, *et al.*, "Integrating t-cell receptor and transcriptome for large-scale single-cell immune profiling analysis," *Preprint at https://www.biorxiv.org/content/10.1101/2021.06.24.449733v2*, 2021.

[24] D. V. Bagaev, R. M. Vroomans, J. Samir, *et al.*, "Vdjdb in 2019: Database extension, new analysis infrastructure and a t-cell receptor motif compendium," *Nucleic Acids Res*, vol. 48, no. D1, pp. D1057–D1062, 2020.

[25] S. Valkiers, M. Van Houcke, K. Laukens, and P. a. Meysman, "Clustcr: A python interface for rapid clustering of large sets of cdr3 sequences with unknown antigen specificity," *Bioinformatics*, vol. 37, no. 24, pp. 4865–4867, 2021.

[26] H. Zhang, X. Zhan, and B. a. Li, "Giana allows computationally-efficient tcr clustering and multi-disease repertoire classification by isometric transformation," *Nat. Commun.*, vol. 12, no. 4699, pp. 1–11, 2021.

[27] K. Mayer-Blackwell, S. Schattgen, L. Cohen-Lavi, *et al.*, "Tcr meta-clonotypes for biomarker discovery with tcrdist3 enabled identification of public, hla-restricted clusters of sars-cov-2 tcrs," *eLife*, vol. 10, e68605, 2021.

[28] F. Pedregosa, V. Michel, O. Grisel, *et al.*, "Scikit-learn: Machine learning in python," *JMLR*, vol. 12, pp. 2825–2830, 2011.

[29] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner, "Weblogo: A sequence logo generator," *Genome Res.*, vol. 14, no. 6, pp. 1188–1190, 2004.

[30] R. C. Edgar, "Muscle: Multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Res*, vol. 32, no. 5, pp. 1792–1797, 2004.

[31] Z. Sethna, Y. Elhanati, C. G. Callan, A. M. Walczak, and T. a. Mora, "Olga: Fast computation of generation probabilities of b-and t-cell receptor amino acid sequences and motifs," *Bioinformatics*, vol. 35, no. 17, pp. 2974–2981, 2019.

[32] M. I. J. Raybould, D. A. Nissley, S. Kumar, and C. M. Deane, "Computationally profiling peptide:mhc recognition by t-cell receptors and t-cell receptor-mimetic antibodies," *Front Immunol.*, vol. 13, 2023.

[33] A. Mayer and C. G. Callan, "Measures of epitope binding degeneracy from t cell receptor repertoires," *PNAS*, vol. 120, no. 4, e2213264120, 2023.

[34] S. A. Schattgen, K. Guion, J. C. Crawford, *et al.*, "Integrating t cell receptor sequences and transcriptional profiles by clonotype neighbor graph analysis (conga)," *Nat. Biotechnol.*, vol. 40, no. 1, pp. 54–63, 2022.

[35] R. O. Emerson, W. S. Dewitt, M. Vignali, *et al.*, "Immunosequencing identifies signatures of cytomegalovirus exposure history and hla-mediated effects on the t cell repertoire," *Nat Genet.*, vol. 49, no. 5, pp. 659–665, 2017.

[36] N. Tickotsky, T. Sagiv, J. Prilusky, E. Shifrut, and N. a. Friedman, "Mcpas-tcr: A manually curated catalogue of pathology-associated t cell receptor sequences," *Bioinformatics*, vol. 33, no. 18, pp. 2924–2929, 2017.

[37] M. P. Lefranc, V. Giudicelli, P. Duroux, *et al.*, "Imgt, the international immunogenetics information system 25 years on," *Nucleic Acids Res*, vol. 43, no. D1, pp. D413–D422, 2015.

[38] M. V. Pogorelyy, A. A. Minervina, M. Shugay, *et al.*, "Detecting t cell receptors involved in immune responses from single repertoire snapshots," *PLoS Biol*, vol. 17, no. 6, e3000314, 2019.

[39] P. Meysman, N. De Neuter, S. Gielis, D. Bui Thi, B. Ogunjimi, and K. Laukens, "On the viability of unsupervised t-cell receptor sequence clustering for epitope preference," *Bioinformatics*, vol. 35, no. 9, pp. 1461–1468, 2018.

[40] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016, ISBN: 978-3-319-24277-4. [Online]. Available: `https://ggplot2.tidyverse.org`.

Methods

Datasets

A consolidated dataset of paired TCR amino acid sequences of human origin was developed using instances drawn from VDJdb [24] and from McPas-TCR [36] as a separate test set. Sequences derived from a 10X study of healthy donors, and CDR3 sequences containing non amino acid symbols, were removed from the input data. V and J gene codes were processed for consistency with IMGT reference sequences [37]. Duplicates were removed within and between datasets using CDR3-V-J bio-identities for both α and β chains, such that a given TCR was encoded in the format CDR3α_TRAV_TRAJ_CDR3β_TRBV_TRBJ. Only those TCRs having TRA or TRB genes included in the reference IMGT alleles of the tcrdist module of the CoNGA conda package (v.0.1.1) were retained, to ensure that consistent numbers of sequences were provided to each model. Benchmarking experiments were performed on VDJdb data after selection and down sampling as described in **Results**.

Models

A systematic review of the literature was conducted to identify studies presenting novel methods for prediction of antigen specificity from TCR sequences. ClusTCR [25], GIANA [26], GLIPH2 [17], iSMART [20], and tcrdist3 [27] were shortlisted for analysis based on the availability of open-source python packages or executable files. ALICE [38] was excluded as more appropriately applied to the identification of expanded clones in individual patient repertoire data. Background methodological detail is included for each of the selected algorithms in **Section S1**. The analytical framework developed to accompany the ClusTCR package was adapted to permit comparison of each of the models described below. All benchmarking experiments were run on a single remote Intel(R) Xeon(R) CPU (E7-8891 v3 @ 2.80GHz) to ensure fair comparison of algorithms with and without parallel processing capability.

The ClusTCR python package (v1.0.2) was imported with Anaconda and implemented using default settings. Benchmarking of ClusTCR was conducted with the CPU version for fair comparison with non-parallelisable models. GLIPH2 was downloaded from the developers' website and run using a combined CD4/CD8 reference, otherwise using default parameters. Where a given sequence was assigned to more than one putative cluster, absolute cluster assignments were made to the cluster having the greatest probability in the output. A Hamming distance model was adapted from a version published in the ClusTCR repository which makes use of sequence hashing for efficient CDR3 comparison, first grouping CDR3 sequences by length and then sorting these superclusters into subclusters with a Hamming distance of 1. iSMART was implemented as in [25] except that V gene usage was included by default. GIANAv4.1 was downloaded from GitHub with an IMGT TRBV reference and implemented in CPU mode using default settings following the framework developed for iSMART. tcrdist3 (v0.2.2) was installed with PyPI and called with a Python script making use of sparse distance matrices for large datasets. tcrdist3 amino acid distance matrices were generated with the default meta-clonotype radius of 50 and clustered with DBScan (eps=0.5) after an initial parameter search (**Fig. S1**). A faster C++ implementation of tcrdist is available as part of the CoNGA package [34], however a steep drop-off in epitope-specific performance was observed when combining this model with DBSCAN (**Fig. S1**). Greedy clustering, used in the original tcrdist publication [14] and evaluated in [25], was excluded from the analysis due to prohibitively slow runtimes. Finally, length and random baseline models were added which assigned TCRs to clusters based on CDR3 amino acid sequence length and random shuffling, respectively.

Metrics

Performance of each model was analysed using the cluster metrics described previously in [25], [39]: purity, consistency and retention, with the addition of the scikit learn implmentation of adjusted mutual information (AMI)[28] to account for cluster entropy. Balanced accuracy, weighted precision, weighted recall, and weighted F1-score were computed as a mean over all clusters and for a given epitope, using the Scikit learn library [28].

Statistics

All statistical comparisons were performed using an R implementation of one-way analysis of variance (ANOVA) and Tukey's HSD for post-hoc significance testing, which analyses are accessible in the accompanying GitHub repository. Comparative boxplots and probability heat maps were produced using ggplot2[40].

CDR3 amino acid motifs

Sequence logos were produced from β chain selections of dataset V500 by retaining TCRs having the modal length from the largest cluster for each of five epitopes of interest. Sequences were aligned with MUSCLEv5.1 [30], and logos produced from the resulting multiple sequence alignments with WebLogo v3.7.12 [29].

Supplemental Information

Section S1: Model methodologies

Here we provide a brief overview of the principal methods underlying each of the models tested, referring the interested reader to the original citations for further details.

ClusTCR [25] makes use of a two-step approach to clustering, in which an N x M matrix of CDR3 amino acid sequence and physicochemical properties is sorted into superclusters using the Faiss library, and the resulting embeddings are sorted with KMeans. A graph network of distances is then produced from these superclusters based on Hamming distances between length sorted CDR3 sequences. Final cluster assignments are made by applying Markov Clustering (MCL) to the network graph.

GIANA [26] applies multidimensional scaling (MDS) to produce matrix representations of TCR CDR3 sequences that approximate BLOSUM62 physicochemical properties, such that the Euclidean distance between two sequences represented with MDS is equivalent to the Smith-Waterman alignment between the BLOSUM representations of those sequences. MDS vectors are pre-sorted on length, and the resulting superclusters are then sorted into subclusters using the Faiss library before clustering on Smith-Waterman distances between kmers.

GLIPH2 [17] is an update to GLIPH [15] that combines global and local cluster analyses. Global distance is defined as sequence mismatches in CDR3 sequences differing at a given position according to a BLOSUM62 subsititution matrix, having shared TRBV gene usage and identical length. Local distance is computed as a statistically significant kmer frequency enrichment in residues predicted to contact peptide-MHC, compared to a sample population.

iSMART [20] incorporates CDR3 and (optionally) V gene usage information, pre-sorting CDR3 sequences according to length and imposing a gap penalty for length mismatched CDR3s related by a single insertion. Alignment scores are computed for a subset of the CDR3 sequences using a BLOSUM62 substitution matrix, and output clusters are assigned based on a threshold alignment score.

tcrdist3 [27] is the latest iteration of tcrdist [14], which makes use of a BLOSUM62 mismatch distance between CDR1, CDR2, CDR2.5 (an MHC-facing loop), and CDR3 sequences. Non CDR3 sequences are inferred from a reference database, a gap penalty is applied to account for sequence insertions/deletions, and a combined similarity score is computed that assigns greater weighting to CDR3 sequences. The resulting distance matrix may then be clustered, for example using a greedy hierarchical search (see **Methods** and **Fig. S1**).

Supplemental Tables

Dataset	**Minimum TCRs per epitope**	**# Unique epitopes**	**N total**
VDJdb			
V10	10	76	760
V50	50	25	1250
V100	100	16	1600
V500	500	5	2500
V1000	1000	4	4000
McPas-TCR	N/A	24	509

Table S1. Dataset size.

Epitope	**# Instances**
KLGGALQAK	13,552
GILGFVFTL	1,830
AVFDRKSDAK	1,143
RAKFKQLL	1,120
IVTDFSVIK	572

Table S2. Frequency of TCR representatives per epitope in preprocessed VDJdb input data prior to down sampling.

	All chains		α **only**		β **only**	
Model	*F1-score*	*Retention*	*F1-score*	*Retention*	*F1-score*	*Retention*
tcrdist3	0.90 ± 0.02	0.20 ± 0.04	0.88 ± 0.01	0.20 ± 0.04	0.93 ± 0.02	0.19 ± 0.04
ismart	0.85 ± 0.03	0.27 ± 0.04	0.80 ± 0.02	0.29 ± 0.05	0.91 ± 0.01	0.24 ± 0.04
clustcr	0.82 ± 0.03	0.20 ± 0.04	0.75 ± 0.02	0.25 ± 0.03	0.88 ± 0.02	0.15 ± 0.03
GIANA	0.81 ± 0.05	0.31 ± 0.05	0.75 ± 0.02	0.25 ± 0.03	0.92 ± 0.01	0.24 ± 0.04
hamming	0.78 ± 0.02	0.32 ± 0.05	0.74 ± 0.02	0.34 ± 0.05	0.82 ± 0.02	0.30 ± 0.05
gliph2	0.74 ± 0.06	0.28 ± 0.06	0.63 ± 0.03	0.41 ± 0.04	0.86± 0.02	0.16 ± 0.03
length	0.17 ± 0.06	1.00 ± 0.00	0.17 ± 0.06	1.00 ± 0.00	0.17 ± 0.06	1.00 ± 0.00
random	0.14 ± 0.03	1.00 ± 0.00	0.13 ± 0.03	1.00 ± 0.00	0.14 ± 0.03	1.00 ± 0.00

Table S3. Global UCM performance, showing mean values ± 95% confidence (datasets V10, V50, V100, V500 and V1000 combined, α and β chain selections, 25 repeats.)

	AVFDRKSDAK		**GILGFVFTL**		**IVTDFSVIK**	
Model	*F1-score*	*Retention*	*F1-score*	*Retention*	*F1-score*	*Retention*
clustcr	0.66 ± 0.02	0.12 ± 0.03	0.93 ± 0.02	0.56 ± 0.01	0.45 ± 0.06	0.11 ± 0.03
GIANA	0.65 ± 0.01	0.20 ± 0.04	0.90 ± 0.02	0.65 ± 0.02	0.80 ± 0.03	0.46 ± 0.03
gliph2	0.64 ± 0.02	0.22 ± 0.06	0.90 ± 0.03	0.62 ± 0.02	0.47 ± 0.03	0.20 ± 0.06
hamming	0.65 ± 0.01	0.23 ± 0.02	0.89 ± 0.01	0.65 ± 0.01	0.79 ± 0.02	0.47 ± 0.02
ismart	0.66 ± 0.01	0.15 ± 0.02	0.91 ± 0.01	0.62 ± 0.01	0.82 ± 0.02	0.42 ± 0.01
length	0.23 ± 0.03	1.00 ± 0.00	0.43 ± 0.06	1.00 ± 0.00	0.38 ± 0.01	1.00 ± 0.00
random	0.19 ± 0.04	1.00 ± 0.00	0.19 ± 0.04	1.00 ± 0.00	0.20 ± 0.04	1.00 ± 0.00
tcrdist3	0.66 ± 0.02	0.10 ± 0.01	0.91 ± 0.01	0.44 ± 0.01	0.85 ± 0.02	0.38 ± 0.01

	KLGGALQAK		**RAKFKQLL**	
Model	*F1-score*	*Retention*	*F1-score*	*F1-score*
clustcr	0.47 ± 0.06	0.10 ± 0.03	0.84 ± 0.02	0.31 ± 0.06
GIANA	0.54 ± 0.03	0.17 ± 0.04	0.84 ± 0.02	0.60 ± 0.03
gliph2	0.49 ± 0.04	0.19 ± 0.06	0.8 ± 0.02	0.39 ± 0.09
hamming	0.55 ± 0.02	0.19 ± 0.02	0.83 ± 0.01	0.61 ± 0.01
ismart	0.56 ± 0.03	0.11 ± 0.02	0.86 ± 0.01	0.57 ± 0.01
length	0.20 ± 0.03	1.00 ± 0.00	0.33 ± 0.04	1.00 ± 0.00
random	0.18 ± 0.05	1.00 ± 0.00	0.17 ± 0.05	1.00 ± 0.00
tcrdist3	0.56 ± 0.04	0.06 ± 0.01	0.87 ± 0.01	0.53 ± 0.01

Table S4. UCM performance and retention by epitope, showing mean values ± 95% confidence (dataset V500, α and β chain selections, 25 repeats.)

Model	**F1-score**	**Retention**
tcrdist3	0.98 ± 0.02	0.15 ± 0.12
GIANA	0.97 ± 0.01	0.25 ± 0.11
ismart	0.97 ± 0.00	0.23 ± 0.06
clustcr	0.96 ± 0.01	0.22 ± 0.08
hamming	0.95 ± 0.04	0.28 ± 0.07
gliph2	0.93 ± 0.03	0.28 ± 0.14
length	0.33 ± 0.09	1.00 ± 0.00
random	0.25 ± 0.02	1.00 ± 0.00

Table S5. UCM performance on instances from McPas-TCR (5 repeats).

	No Synthetic TCRs		+10K Synthetic TCRs	
Model	*F1-score*	*Retention*	*F1-score*	*Retention*
		AVFDRKSDAK		
clustcr	0.68 ± 0.06	0.06 ± 0.02	0.67 ± 0.03	0.09 ± 0.01
GIANA	0.65 ± 0.02	0.11 ± 0.03	0.66 ± 0.02	0.12 ± 0.02
gliph2	0.67 ± 0.03	0.07 ± 0.02	0.66 ± 0.01	0.12 ± 0.01
hamming	0.64 ± 0.02	0.17 ± 0.03	0.57 ± 0.02	0.27 ± 0.02
ismart	0.65 ± 0.03	0.11 ± 0.03	0.67 ± 0.02	0.12 ± 0.02
length	0.20 ± 0.05	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
random	0.17 ± 0.14	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
tcrdist3	0.63 ± 0.05	0.09 ± 0.03	0.68 ± 0.03	0.07 ± 0.01
		GILGFVFTL		
clustcr	0.96 ± 0.01	0.57 ± 0.02	0.94 ± 0.00	0.59 ± 0.02
GIANA	0.93 ± 0.02	0.62 ± 0.03	0.92 ± 0.01	0.64 ± 0.02
gliph2	0.96 ± 0.01	0.57 ± 0.02	0.94 ± 0.01	0.60 ± 0.01
hamming	0.91 ± 0.02	0.68 ± 0.02	0.85 ± 0.01	0.73 ± 0.02
ismart	0.93 ± 0.02	0.62 ± 0.03	0.92 ± 0.01	0.64 ± 0.01
length	0.56 ± 0.01	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
random	0.17 ± 0.15	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
tcrdist3	0.92 ± 0.02	0.47 ± 0.03	0.93 ± 0.01	0.49 ± 0.03
		IVTDFSVIK		
clustcr	0.32 ± 0.04	0.04 ± 0.01	0.54 ± 0.04	0.07 ± 0.01
GIANA	0.87 ± 0.01	0.40 ± 0.02	0.87 ± 0.01	0.41 ± 0.01
gliph2	0.49 ± 0.02	0.05 ± 0.01	0.57 ± 0.04	0.08 ± 0.01
hamming	0.83 ± 0.01	0.44 ± 0.02	0.78 ± 0.02	0.52 ± 0.02
ismart	0.86 ± 0.01	0.39 ± 0.02	0.86 ± 0.01	0.41 ± 0.01
length	0.40 ± 0.01	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
random	0.20 ± 0.13	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
tcrdist3	0.88 ± 0.01	0.38 ± 0.02	0.89 ± 0.02	0.38 ± 0.01
		KLGGALQAK		
clustcr	0.43 ± 0.14	0.03 ± 0.01	0.61 ± 0.03	0.07 ± 0.01
GIANA	0.52 ± 0.08	0.06 ± 0.01	0.61 ± 0.09	0.08 ± 0.01
gliph2	0.46 ± 0.13	0.05 ± 0.01	0.60 ± 0.04	0.11 ± 0.02
hamming	0.55 ± 0.06	0.15 ± 0.02	0.53 ± 0.03	0.28 ± 0.02
ismart	0.52 ± 0.09	0.06 ± 0.01	0.61 ± 0.07	0.09 ± 0.01
length	0.18 ± 0.09	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
random	0.18 ± 0.14	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
tcrdist3	0.55 ± 0.13	0.04 ± 0.01	0.59 ± 0.08	0.04 ± 0.01
		RAKFKQLL		
clustcr	0.80 ± 0.12	0.15 ± 0.10	0.83 ± 0.02	0.20 ± 0.04
GIANA	0.88 ± 0.02	0.53 ± 0.03	0.88 ± 0.02	0.53 ± 0.03
gliph2	0.78 ± 0.13	0.13 ± 0.10	0.80 ± 0.03	0.21 ± 0.04
hamming	0.84 ± 0.02	0.58 ± 0.02	0.79 ± 0.01	0.64 ± 0.02
ismart	0.88 ± 0.02	0.53 ± 0.03	0.88 ± 0.01	0.54 ± 0.03
length	0.27 ± 0.04	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
random	0.21 ± 0.10	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
tcrdist3	0.89 ± 0.01	0.51 ± 0.02	0.89 ± 0.02	0.51 ± 0.03

Table S6. UCM performance in the presence of synthetic TCR sequences produced with OLGA, V500, β chain selections (5 repeats).

Supplemental Figures.

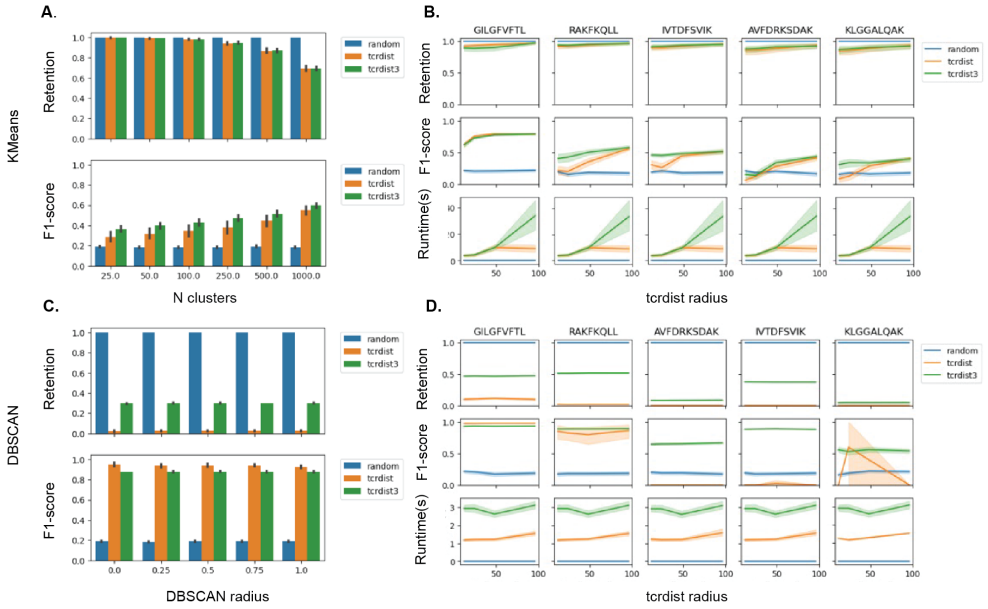

Figure S1. Selecting tcrdist hyperparameters for C++ (tcrdist) or python (tcrdist3) implementations of tcrdist (Schattgen et al., 2022, Mayer-Blackwell et al., 2021), using KMeans (A-B) or DBSCAN (C-D) applied to dataset V500, β chain selections. A), C): Performance as a function of the number of clustering algorithm hyperparameters. B, D): performance per epitope as a function of tcrdist radius.

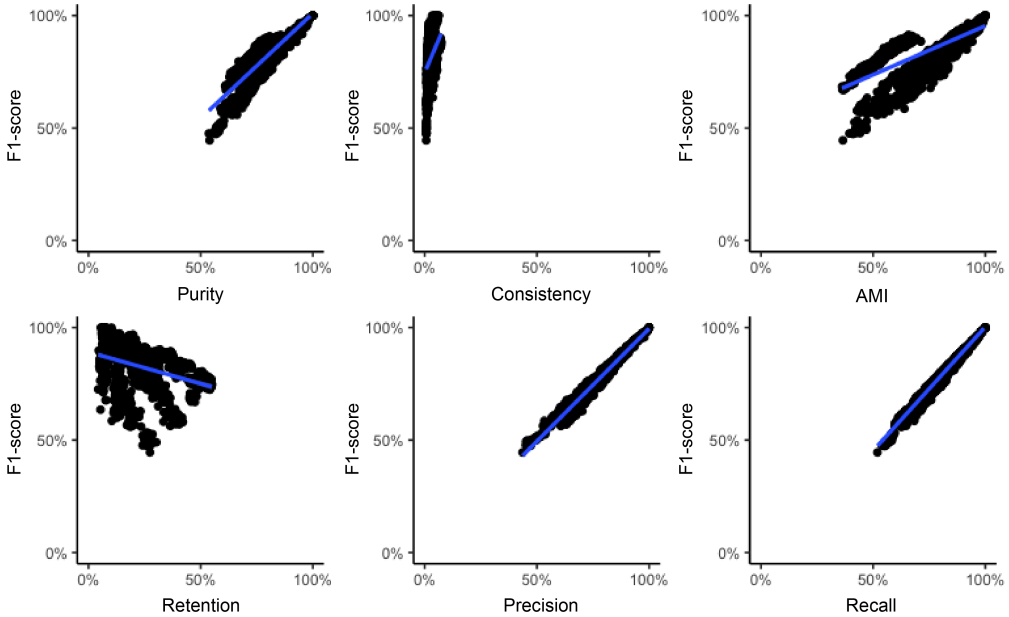

Figure S2. Correlation of UCM metrics, datasets V10, V50, V100, V500 and V1000, α and β chain selections combined (25 repeats).

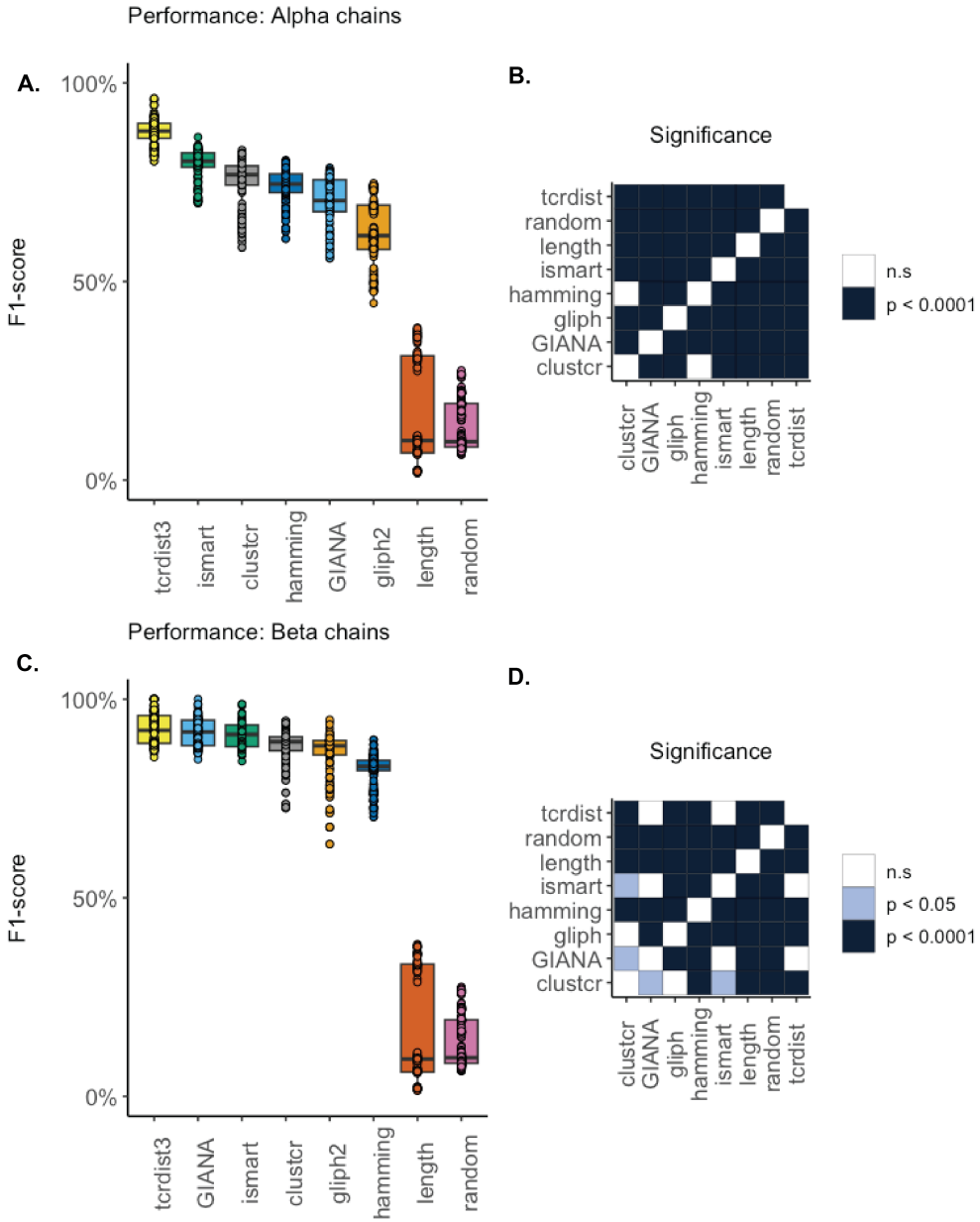

Figure S3. Performance differences (A, C) and statistical significance (B, D) by chain selection, datasets V10, V50, V100, V500 and V1000 combined (25 repeats). A-B): α chain selections; C-D) β chain selections.

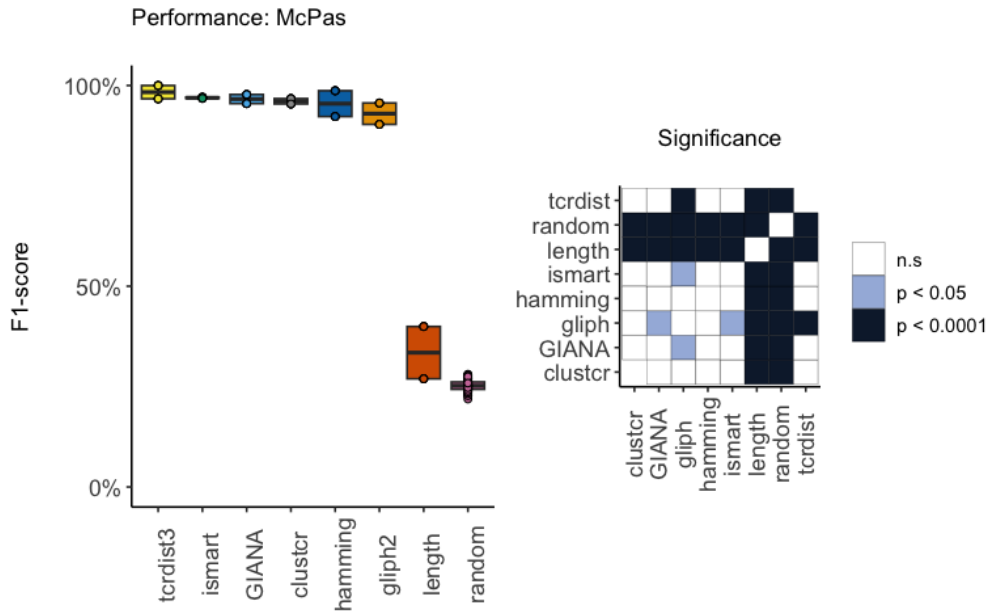

Figure S4. Performance differences (*left*) and significance (*right*) for McPas-TCR, α and β chain selections (25 repeats).

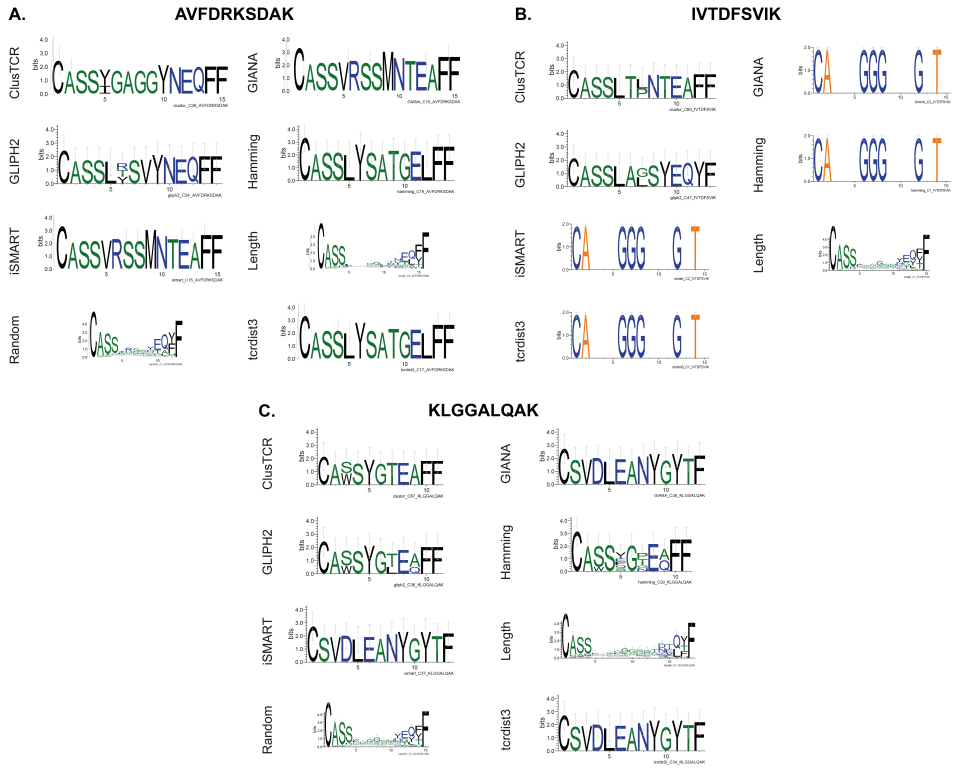

Figure S5. Motifs for the largest cluster formed per epitope, dataset V500 (β chain selection). A): AVFDRKSDAK; B): IVTDFSVIK; C): KLGGALQAK

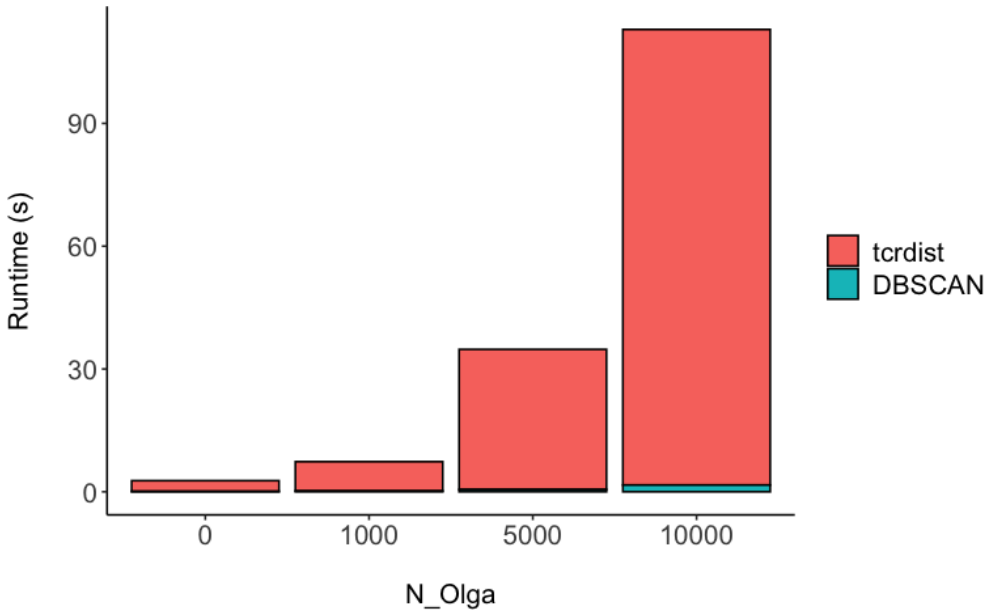

Figure S6. Relative contribution to runtimes of tcrdist3 matrix calculation and clustering with DBSCAN in the presence of increasing synthetic TCR sequences produced with OLGA[31], dataset V500, β chain selection [31].