Supplemental Information

Section S1: Literature Review

A search of PubMed was conducted on 30/01/22 with the aim of identifying models for inference of TCR epitope specificity, supplemented by a manual search of the literature making use of Google Scholar and BioRxiv.

Search terms:
(T cell receptor OR TCR) AND (antigen OR peptide) AND
(algorithm OR (cluster* OR predict*)) NOT
(chimeric antigen receptor OR CAR OR CAR-T)

Inclusion criteria:

- Primary papers released from 2017 onwards
- Methods for clustering T cell receptors according to antigen specificity

Exclusion criteria:

- BCR clustering / antigen predictions
- Papers published before 01 Jan 2017
- Review papers
- Supervised predictive models and deep neural networks

Models identified through systematic review:

- ClusTCR [29]
- GIANA [30]
- GLIPH2 [17]
- iSMART [20]
- tcrdist3 [31]

Section S2: 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. Hamming distance, GLIPH2, tcrdist3, and iSMART implementations were adapted from the ClusTCR python package [29]. Unless otherwise specified, models were applied using default settings.

ClusTCR [29] 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. The ClusTCR python package (v1.0.2) was imported with Conda and implemented using default settings. Benchmarking of ClusTCR was conducted with the CPU version for fair comparison with non-parallelisable models, and V gene inputs enabled.

GIANA [30] 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. 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.

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. 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.

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. iSMART was implemented as in [29] except that V gene usage was included by default.

tcrdist3 [31] 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. tcrdist3 (v0.2.2) was installed with PyPI and called with a Python script making use of sparse distance matrices and chunking for large datasets. As tcrdist3 generates a distance measure but does not explicitly cluster instances, a scikit-learn implementation of DBSCAN [34] was used to group distance matrices produced with tcrdist3, consistent with [29] and following comparison of model performance with different model implementations and clustering approaches (**Fig. S1**). Amino acid distance matrices were generated with the default meta-clonotype radius of

50. A faster C++ implementation of tcrdist is available as part of the CoNGA package [28], 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 [29], was excluded from the analysis due to prohibitively slow runtimes.

Baseline models A Hamming distance model was adapted from a version published in the ClusTCR repository [29] that makes use of sequence hashing for efficient CDR3 comparison, first grouping CDR3 sequences by length and then sorting these superclusters into subclusters differing by only one amino acid. Length, V-gene, and random baseline models were added that assign TCRs to clusters based on CDR3 length, V-gene codes, or random shuffling, respectively.

Supplemental Figures

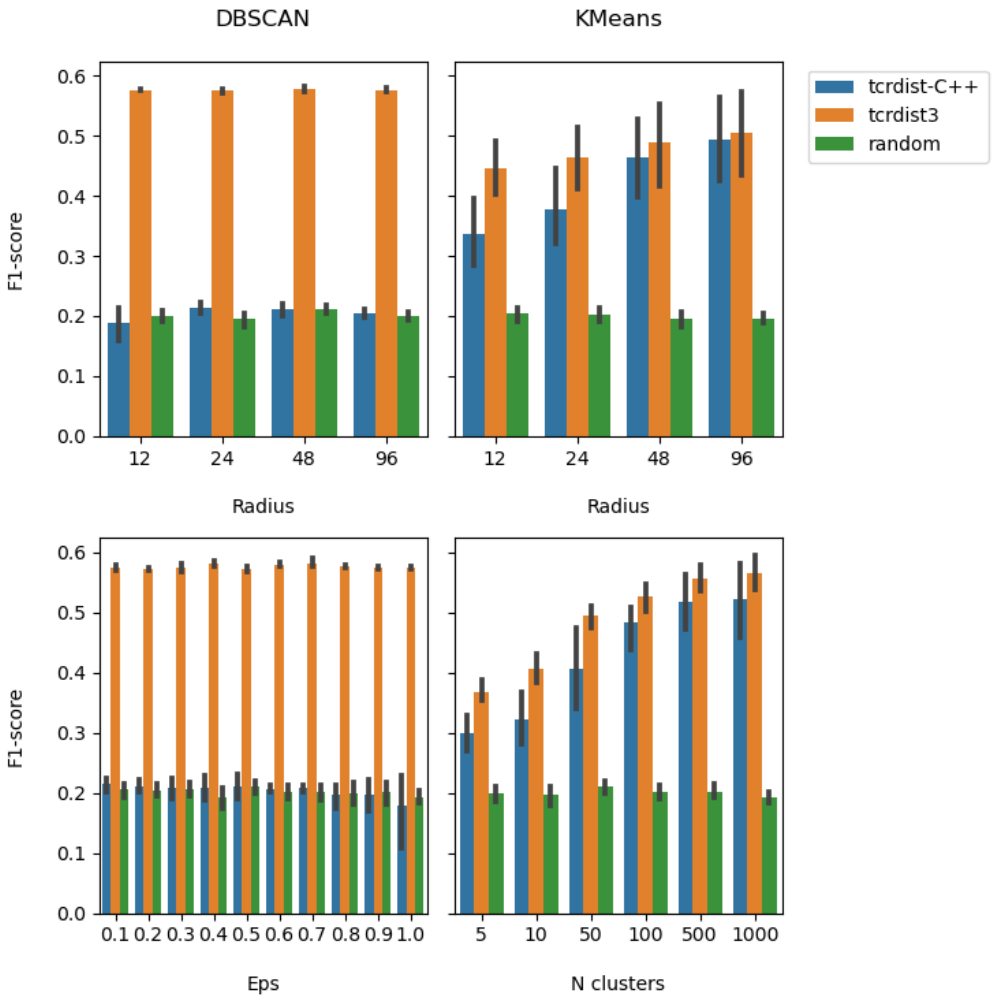

Figure S1. Selecting tcrdist hyperparameters for C++ (tcrdist) or python (tcrdist3) implementations of tcrdist (Schattgen et al., 2022, Mayer-Blackwell et al., 2021), using KMeans or DBSCAN applied to dataset V1000, β chain selections. *Top row*: Performance as a function of tcrdist radius. *Bottom row*: performance as a function of clustering algorithm hyperparameters.

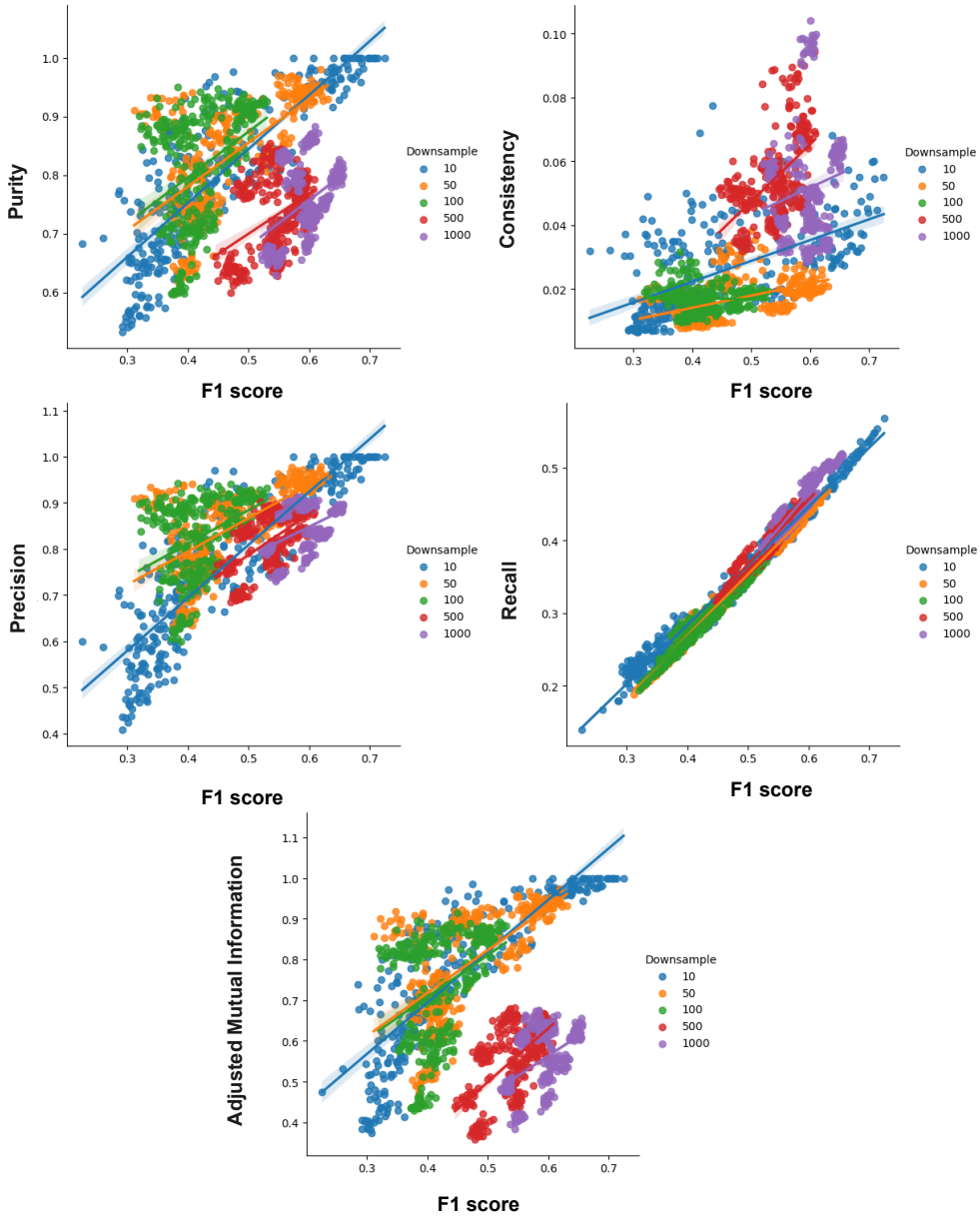

Figure S2. Correlation of UCM metrics, datasets V10, V50, V100, V500 and V1000, α and β chain selections combined (25 repeats). Results combined for all models except length, V-gene and random baselines.

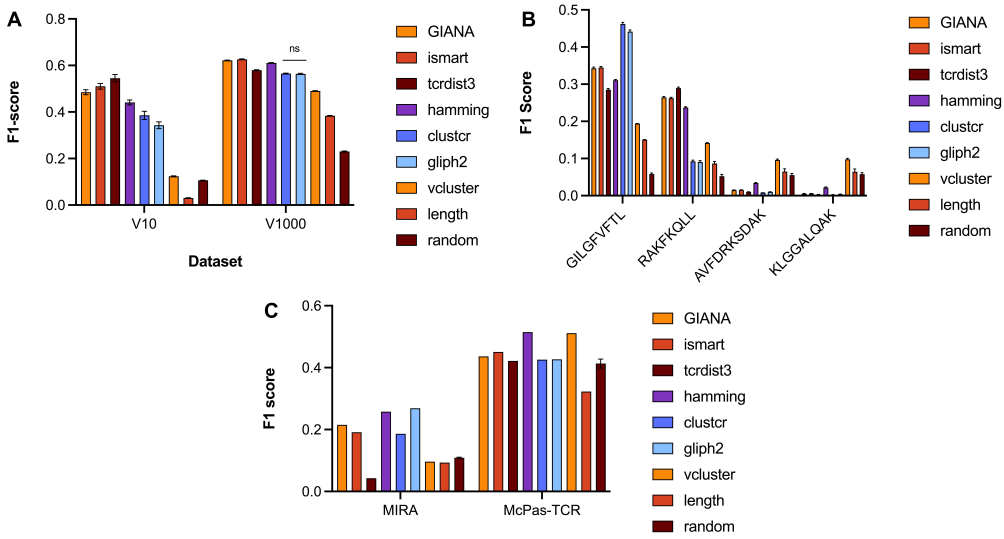

Figure S3. Performance differences A) by dataset, α and β chains combined, all comparisons reaching statistical significance ($p<0.05$) except for those between ClusTCR and GLIPH2 for V1000; B) V1000, β chain selection, by epitope; C) MIRA [26] and McPas [25], β chain selections.

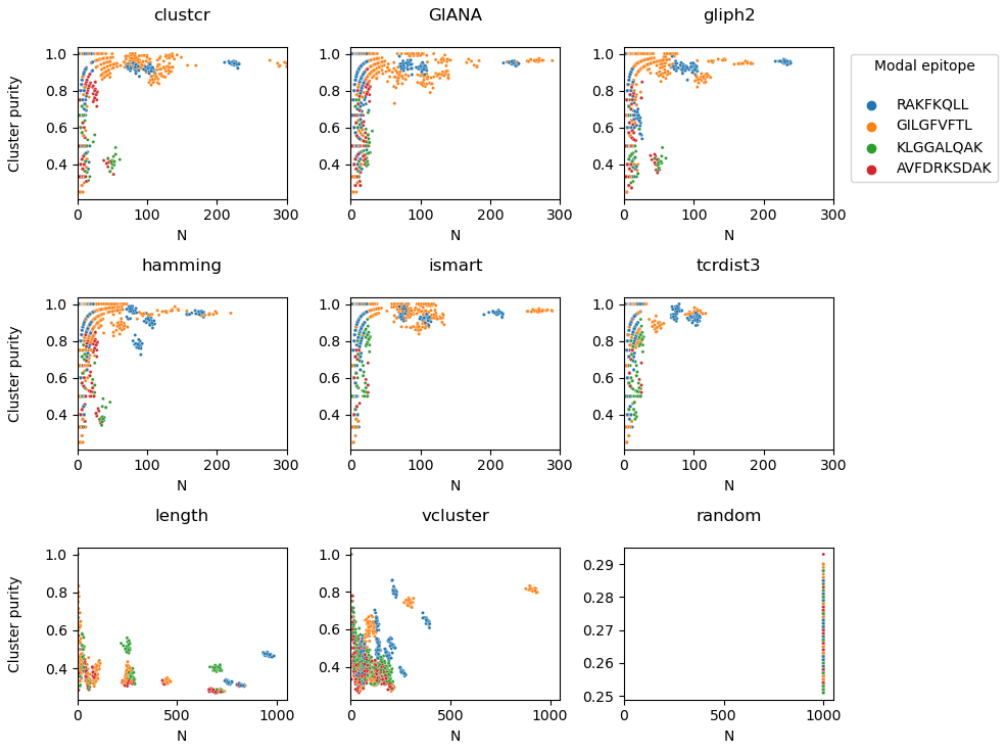

Figure S4. Cluster size and epitope purity, V1000, α and β chain selections combined.

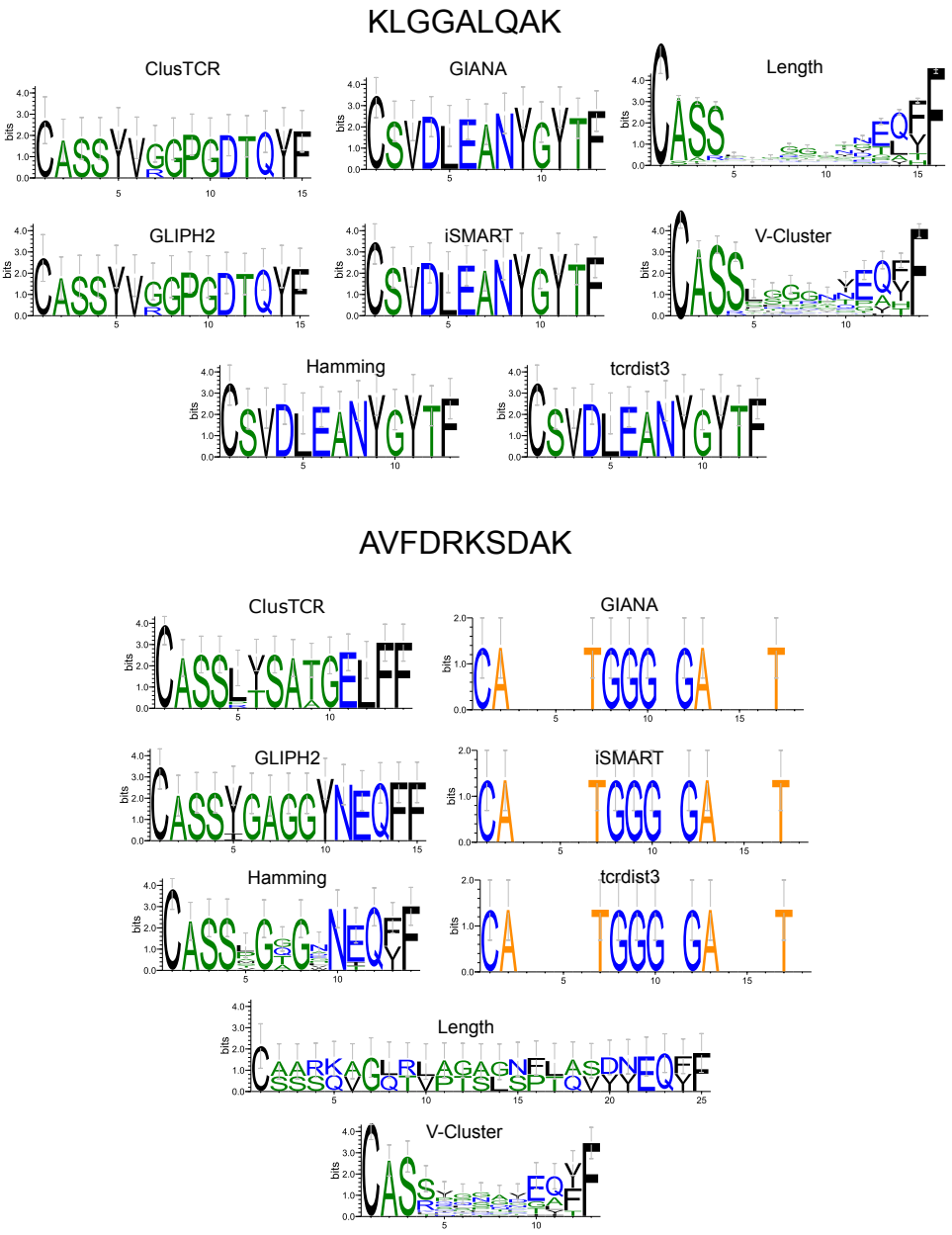

Figure S5. CDR3 sequence logos for the largest clusters produced per epitope, dataset V1000 (β chain selection). Logos were produced with WebLogo [41] for TCRs in the largest cluster produced for a given epitope per model following sequence alignment with MUSCLE [40].

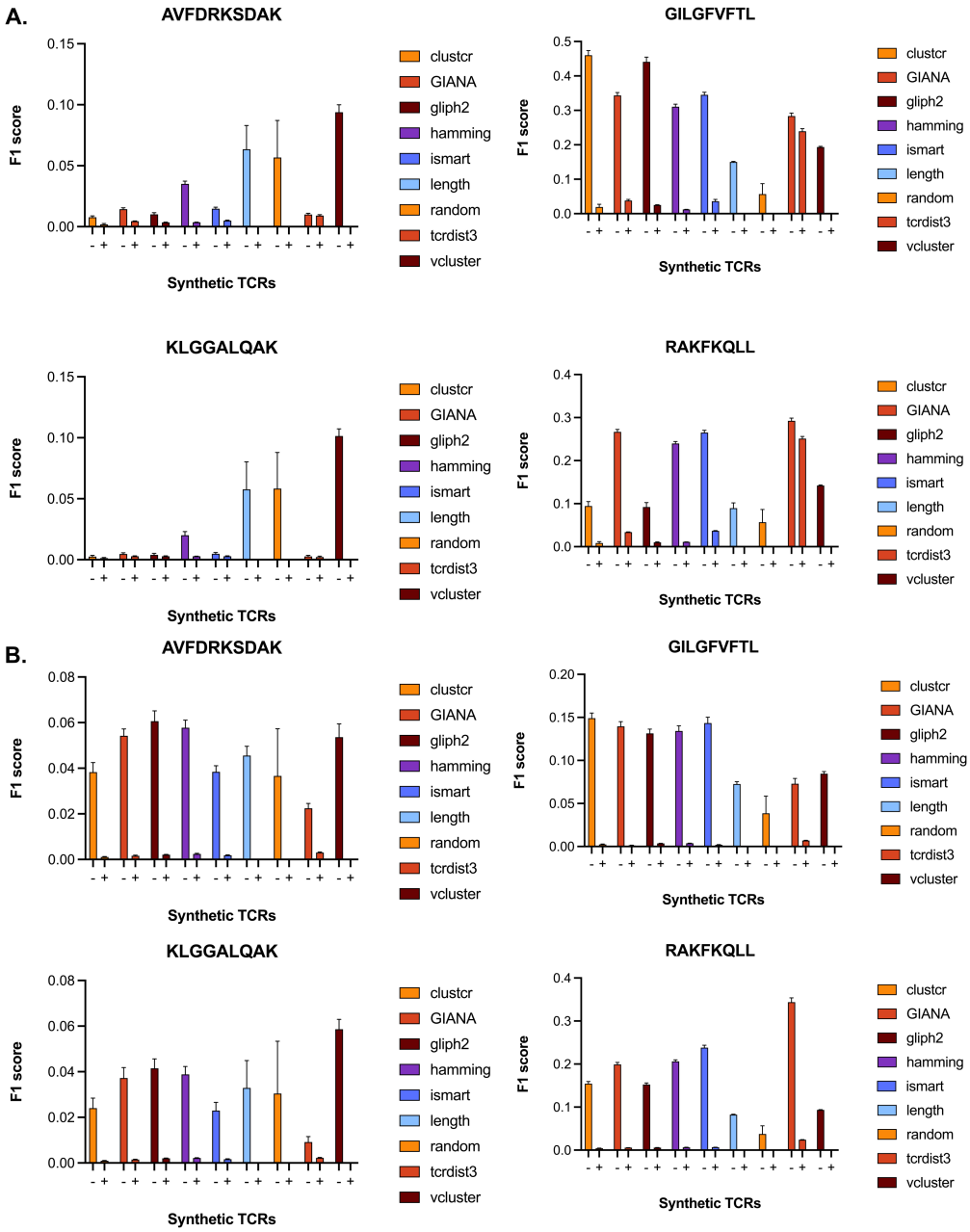

Figure S6. F1-scores per epitope in the presence or absence of 100,000 synthetic TCR sequences, produced with OLGA [35] All experiments conducted on dataset V1000 for A) β chains and B) α chains.

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
MIRA	N/A	796	593888

Table S1. Dataset size.

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

Table S2. Frequency of TCR representatives per epitope in V1000 prior to down-sampling.

Model	F1-score (%)
GIANA	52.9 ± 1.1
ismart	53.1 ± 1.1
hamming	50.8 ± 1.2
tcrdist3	50.1 ± 1.2
clustcr	45.8 ± 1.4
gliph2	44.7 ± 1.5
vcluster	32.0 ± 2.4
length	19.6 ± 2.5
random	13.9 ± 0.1

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

Model	α	β
GIANA	46.5 ± 1.7	59.2 ± 0.9
ismart	47.4 ± 1.6	58.8 ± 1.0
tcrdist3	44.0 ± 1.4	56.2 ± 1.3
hamming	46.9 ± 1.5	54.6 ± 1.0
clustcr	42.7 ± 1.2	48.9 ± 1.6
gliph2	42.8 ± 1.5	46.5 ± 1.6
vcluster	30.7 ± 2.1	33.4 ± 2.7
length	19.2 ± 2.4	20.0 ± 2.6
random	14.0 ± 0.2	13.9 ± 0.1

Table S4. UCM performance by chain selection, showing mean values ± 95% confidence for datasets V10, V50, V100, V500 and V1000 combined, after 25 repeats.

Model	V10	V1000
GIANA	54.5 ± 1.7	58.0 ± 0.9
ismart	51.1 ± 1.6	62.7 ± 1.0
tcrdist3	48.5 ± 1.4	62.1 ± 1.3
hamming	44.1 ± 1.5	61.1 ± 1.0
clustcr	38.5 ± 1.2	56.5 ± 1.6
gliph2	34.3 ± 1.5	56.4 ± 1.6
vcluster	12.3 ± 2.1	49.0 ± 2.7
length	3.0 ± 2.4	38.4 ± 2.6
random	10.6 ± 0.2	23.0 ± 0.1

Table S5. UCM performance by dataset, showing mean values ± 95% confidence for V10 and V1000, α and β chain selections combined, after 25 repeats.

Model	AVFDRKSDAK	GILGFVFTL	KLGGALQAK	RAKFKQLL
GIANA	0.8 ± 0.1	46.2 ± 0.5	0.3 ± 0.1	9.2 ± 0.3
ismart	1.5 ± 0.1	34.3 ± 0.3	0.5 ± 0.1	26.4 ± 0.2
tcrdist3	1 ± 0.1	44.2 ± 0.5	0.4 ± 0.1	9.1 ± 0.4
hamming	3.4 ± 0.1	31.1 ± 0.3	2.2 ± 0.2	23.7 ± 0.2
clustcr	1.5 ± 0.1	34.5 ± 0.3	0.5 ± 0.1	26.2 ± 0.2
gliph2	6.5 ± 0.8	15 ± 0	6.4 ± 0.7	8.7 ± 0.6
vcluster	5.5 ± 0.4	5.8 ± 0.3	5.8 ± 0.4	5.2 ± 0.5
length	1 ± 0.1	28.5 ± 0.3	0.3 ± 0.1	28.9 ± 0.2
random	9.6 ± 0.3	19.4 ± 0.1	9.8 ± 0.3	14.2 ± 0.1

Table S6. UCM performance by epitope, showing mean values ± 95% confidence for V1000, β chain selections, after 25 repeats.

Model	MIRA	McPas-TCR
GIANA	21.5 ± 1.7	43.6 ± 0.9
ismart	19.1 ± 1.6	45.1 ± 1
tcrdist3	4.3 ± 1.4	42.2 ± 1.3
hamming	25.8 ± 1.5	51.5 ± 1
clustcr	18.6 ± 1.2	42.6 ± 1.6
gliph2	26.9 ± 1.5	42.7 ± 1.6
vcluster	9.6 ± 2.1	51.1 ± 2.7
length	9.3 ± 2.4	32.3 ± 2.6
random	10.9 ± 0.2	41.3 ± 0.1

Table S7. UCM performance for test datasets, showing mean values ± 95% confidence, β chain selections and no down-sampling, after 5 repeats.