Autoencoder based local T cell repertoire density can be used to classify samples and T cell receptors

T cell repertoire density estimate

Shirit Dvorkin[1], Reut Levi[1], Yoram Louzoun[1, +]

1. Department of Mathematics, Bar Ilan University, Ramat Gan, Israel.

+ To whom correspondence and requests should be sent: louzouy@math.biu.ac.il

Abstract

Recent advances in T cell repertoire (TCR) sequencing allow for characterization of repertoire properties, as well as the frequency and sharing of specific TCR. However, there is no efficient measure for the local density of a given TCR. TCRs are often described either through their Complementary Determining region 3 (CDR3) sequences, or theirV/J usage or their clone size. We here show that the local repertoire density can be estimated using a combined representation of these components through distance conserving autoencoders and Kernel Density Estimates (KDE).

We present ELATE – an Encoder based LocAl Tcr dEnsity and show that the resulting density of a sample can be used as a novel measure to study repertoire properties. The cross-density between two samples can be used as a similarity matrix to fully characterize samples from the same host. Finally, the same projection in combination with machine learning algorithms can be used to predict TCR-peptide binding through the local density of known TCRs binding a specific target.

Code availability- https://github.com/louzounlab/Autoencoder

Keywords: T cell receptor, Local density, Autoencoder, Distance conserving, Machine learning

Introduction

T-cell receptor (TCR) repertoire formation results from two major procedures: Rearrangements of the TCR α and β chains gene segments (V (D) and J) and formation of the junctions between them [1]. During segments joining, nucleotides are inserted and deleted at the junctions between pairs of rearranging genes, forming the highly variable complementarity determining region 3 (CDR3) that is directly implicated in antigen recognition, and further augmenting diversity [1, 2].

Repertoire next-generation immune-sequencing methods (often denoted rep-seq)[3] supply detailed measures of T cell clone frequencies in samples[4-8]. At the broad level, Rep-Seq methods have three main usages: A) Descriptive - general statistics of the repertoire, including clone size distribution, V, D and J usages, CDR3 length, descriptions of alleles and haplotypes, and in B cells also properties of somatic hypermutations and clonal trees[3, 9]. B) Comparative - comparison between different hosts or different compartments within the same host and the sharing of clones between such compartments, comparison between repertoires in different conditions (e.g. patients with specific diseases), and even comparisons between B and T cells[10, 11]. C) Predictions and Machine learning, including prediction of receptor targets, and prediction of compartments using CDR3 composition or V gene usage [12].

The full sequence of T cell receptors is typically not studied. Instead, sequences are characterized by their V and J genes, their observed clone size, and CDR3 sequence[13]. The CDR3 sequence, the V-beta usage, and the clone size distribution are usually analyzed as distinct features of the repertoire[2, 14, 15]. The separation between these components is mainly the result of the different representations (categorical for V(D)J usage, CDR3 sequence, and counts for clone size). There is a need for an efficient framework to unify them into a single coherent representation. Moreover, while there are standard methods to compare V(D)J usage (e.g. Kullback Leibler divergence on the distributions [16]) or the clone size distribution (e.g. entropy or Simpson index [14]), an efficient measure to define TCR local density based on the

CDR3 sequence is still needed. The local density of a TCR can be defined through the distance of a TCR to other nearby TCRs. Such a density, requires a definition of a distance between TCRs - for example the edit distance between CDR3 sequences. However, computing the edit distance between each pair of CDR3 sequences is prohibitively expensive. for large samples.

Estimating the TCR local density in the receptor space has important applications, including the comparison of densities of different TCR groups, estimating the cross density of different repertoires (i.e. the density of one TCR from one repertoire around TCRs from a different repertoire), and characterization of the TCR distribution properties. For example, in order to follow the developmental stages of T cells, one may want to detect regions in receptor space that are evolving to become denser or less dense. Similarly, one can ask whether some regions are “holes” in the repertoire, with a very low receptor density. We here propose that through autoencoders and kernel density estimators, CDR3 sequence, V/J gene and clone size can be unified to provide a computationally efficient description of the repertoire, and the local density in receptor space around each TCR can be properly defined.

Autoencoders are typically used for dimensionality reduction[17]. An autoencoder is an encoder-decoder combination where the output of the decoder is simply the input to the encoder, and the goal is to minimize the difference between the two. Autoencoders are used to design efficient data representation. Encoders project an input vector to a lower dimension. The decoder projects the low-dimensional vector back to the original dimension. In neural nets based autoencoders, during the training process, the autoencoder minimizes the difference between the input of the encoder and the output of the decoder. The low-dimensional projection is then used as an efficient representation of the input vector.

We propose the following outline for repertoire characterization by developing ELATE (Encoder based LocAl Tcr dEnsity), with each step further explained in the Results and Methods sections:

A) Use an autoencoder (either distance maintaining or not) to project each CDR3 sequence into a Z dimensional real space.

B) Project repertoires using a constant number of clones from each repertoire.

C) Define cross/self repertoire local density in this Z-dimensional space using KDE methods[18].

To test the possibility of estimating the local density of T cell receptor sequences as defined by their CDR3 sequence, their V gene, and their clone size, we developed the ELATE encoder. This analysis is mostly focused on the beta chain since it has been assumed the repertoire diversity and its binding to target MHC-peptide are determined more by the beta chain than the alpha chain[19], and there are more beta than alpha-beta published datasets However, ELATE can be equally applied to the alpha chain.

Results

ELATE can produce accurate reconstruction and also accurate TCR distances.

ELATE projects the CDR3 sequence and optionally the V gene using a Multi Layer Perceptron (MLP) based AE. (see Methods and Fig. 1 for encoder structure and for the computational flow for a single receptor). An Long-Short-Term-Memory (LSTM) recurrent neural network [20] based approach to handle different lengths was also tested but had a lower decoder accuracy. Similarly, when comparing encoder and LSTM for predicting TCR-peptide binding, encoders provided better results[21]. The decoder predicts the entire input vector, including the stop signal and what comes after it. All values after the predicted stop signal are then zeroed. To evaluate the performance of the AE, we used the percentage of sequences successfully reconstructed with up to *K* errors, where a mismatch in length was defined as an error for all *K*s. Note that the current definition does not take into account the similarity between amino acids as measured through either their evolutionary distance or through their biochemical

distances. Such distances can be easily integrated in both the training and evaluation stages of ELATE.

In all the observed datasets, very few pairs of CDR3 sequences differ by less than 2 AA one from each other (more precisely have an edit distance of less than 2). Indeed, when the distribution of edit distance between every pair of sequences from pairs of random samples is calculated, the distributions are consistently normal, with practically no values below 2 (see a representing example in Fig. 2A). Thus, even an AE producing 1 or 2 errors would lead to a representation closer to the original CDR3 than the vast majority of other sequence in a different sample. Note that the absolute number of non-shared CDR3 sequences within 2AA of a CDR3 sequence from another sample depends on the sampling depth. We thus report the fraction of sequences successfully reconstructed with 0, 1, or 2 errors (Fig. 2).

To test the applicability of the AE, several previously published datasets with different categories were used in this analysis. The first set studied was the GLIPH dataset which is composed of seven TCR samples, where all TCRs in a sample share specificity to the same peptide[22]. Two more datasets of tumor-bearing mice from ImmunoMap and TILs[23] were used. The first had two categories: CD8 T cells responding to TRP2, a shared self-peptide tumor antigen and SIY, a model foreign-antigen. The second is composed of samples taken from mice treated with RT and 9H10 immunotherapies and a combination of both. Also, a set of sorted naïve and memory cells (Benny Chain dataset), CD4+ and CD8+ from alpha and beta chains of three donors was collected[24]. To explore variations over time, we used the response to the yellow fever vaccine from different time-points in three pairs of identical twins[25]. Finally, a collection of TCRs with one sample from each host with any type of cancer and samples from healthy donors were collected from the Immune Epitope Database (IEDB). The cancer analysis was performed to show the applicability of ELATE to TCR samples not directly taken from deep sequencing experiments.The GLIPH and Benny Chain datasets do not contain V usage in their repertoire and the GLIPH dataset has no frequencies of clones.

While ELATE can quite precisely encode the CDR3 distribution (Fig. 2C, D), it fails to reproduce the distance between TCRs (Fig. 2E). To handle that, we produced a novel distance maintaining encoder (EM). In this framework, besides minimizing the reconstruction error. An embedding loss ($loos_{DIS}$) of the sum over pairs of TCRs of the difference between the Euclidean distance between the embeddings and the original Euclidean distance between the CDR3 one-hot vectors is added at the hidden layer. When the V gene of each TCR was provided, a V gene-specific E was also concatenated to the CDR3 encoder. The V based encoder can be used with or without the embedding loss.

The accuracies with zero, one, and two mismatches for all four models (E, VE, EM, VEM) are shown for all the test set of all datasets (except for GLIPH and Benny Chain when analyzing V) in Figures 2C-D, G-H. None of the receptors in the test set were used for either training or hyper-parameter tuning. The accuracy of E is consistently much higher than the EM, with or without V, with high accuracy of >90% in multiple data sets for 2 mismatches accuracy. The models with the additional embedding constraints have lower accuracy (Fig. 2G, H), as expected given the need to minimize in parallel two loss functions. Similarly, encoding with V gives a lower accuracy than without it, again as expected from the larger number of degrees of freedom. However, as will be further shown, the encoding with the embedding is better for multiple applications. To test whether the addition of the distances loss does maintain the distance distribution, the pairwise edit distances of the input one-hot vectors of the CDR3s in the original set and the all distances in the new representations set were compared (Fig. 2E, F). The correlation is higher in the constrained model (Fig. 2F) with a Spearman correlation of 0.9 ($p<1.e-100$), compared with the regular AE (correlation of 0.5, Fig. 2E).

ELATE can be used to detect features of single clones.

To highlight the importance of the encoding, we first tested whether the encoder reveals the reported selection against specific AA in CDR3 sequences[26]. Samples composed of 100 TCR per sample from the Benny Chain dataset were projected using the AE. To properly visualize the projection, TSNE was used to lower the dimensions to 2 (Fig. 3A). Although we used TCR

from different samples, consistently, TCRs with Cysteines (red color) were projected as clusters (Fig. 3A). Moreover, CDR3 sequences containing a Cysteine or Glycine have lower densities than other CDR3 sequences (Fig. 3G). Cysteines form disulfide bonds, an important feature that contributes to the stability of proteins, yet in CDR3, such bonds may deform the CDR3 surface. Therefore, most CDR3 sequences do not contain Cysteines. For Proline, we observe an interesting difference, where TCRs with a single proline are denser that either without or with 2. We currently have no clear explanation for that.

Next, we tested whether the AE captures the CDR3 length distribution of the repertoire. Indeed, if distances are maintained, similar length CDR sequences should have close projections. Moreover, CDR3 with much longer or shorter than average sequences should be again projected to the periphery, as they are also less frequent and selected against[26, 27]. Again, the AE projects the mid-length CDR3 sequences to the center of the distribution and the long and short CDR3 sequences to the periphery (Fig. 3B, where each color is a different length).

Public TCRs have been described in a variety of immune responses[28]. A public clone is a clone shared by different donors. Again, if the encoded projections represent the functionality of the repertoire, one would expect public clones to be more central. We calculated the radius from the center of mass of all projected TCRs in Benny Chain dataset. Note that such an analysis is only possible following the translation to real-valued representations. We computed the radius of TCRs vs their publicity, where the publicity of each sequence is defined as the number of samples containing it, out of 56 samples (Fig. 3C). Indeed, public clones are more central (spearman correlation, $p<0.001$). We further tested if public clones are in denser regions of the projection, where the self-density was defined as the density of other TCR projections from the sample around the clone. Density was estimated using a Kernel Density Estimation (KDE). Again public clones reside in denser regions than private clones (spearman correlation, $p<0.001$) (Fig. 3C, D).

Finally, we checked the overall characterization of the TCR repertoire as a possible biomarker for different diseases[29], for example, a biomarker for cervical cancer[11]. We expect TCR samples (those are not actual rep-seq, but rather samples of reported TCRs) in such conditions to show distinct projections compared with random repertoires. To evaluate the scattering of different categories in the embedded space, we used the AE projections in two disease state datasets. Again, the vectors were projected to 2 dimensions using TSNE to visualize their positions. When analyzing the GLIPH data set, all of the clones in the TPRVTGGGAM peptide are significantly clustered together (Fig. 3E). In the cancer data set, when projecting the Cancer vectors to 2 dimensions, a separation between most TIL and naïve TCRs emerged (Fig. 3F, and t-test with the y-axis position $p<0.001$). In contrast with the previous tests, the cancer samples are from the IEDB and are enriched for disease specific TCRs. One can now use this density to estimate whether peptide-specific TCRs are in denser regions than random peptides from the same sample. For almost all targets, TCRs specific for this target have a higher density than random TCRs, not defined to bind a known target. This could be a sampling bias or a real difference, but the current method is the first that allows for such a measurement (Fig. 3G).

ELATE with KDE can be used to cluster samples.

The autoencoder can be used to compare samples and not only to compare TCRs within a sample. Again, we used a constant sample size (100 here) to avoid sample size bias. We then defined a similarity between samples and tested whether this similarity can be used to detect which sample comes from the same host. The similarity was defined using the AE, the Kullback- Leibler (KL) divergence of the V gene usage on the sample, or the edit distance (ED) between CDR3 of different samples. The ED distance between each sequence from one sample and each sequence from the other was computed. The distance between the two samples was defined as the average of all minimum ED values (see Methods). For the AE based distance, the similarity between samples A and B is defined as the average KDE based density of sample A projected sequences based on the projected sequences of sample B. The distance was computed for different variants of the AE, including the basic autoencoder (E), the autoencoder

combined with V (VE) or with CDR3 distances (EM) and combined with both (VEM). The KDE was performed twice, one time with the original set of vectors, and again when the clones frequency was incorporated (see Methods). We analyzed the vaccine dataset. This dataset contained samples from three pairs of identical tweens. The samples were taken at six different time points before and after a yellow fever vaccination. Hierarchical clustering was performed using the distance/similarity definitions above. The clustering that best reproduces the division to hosts is VE (Fig. 4A). Moreover, twins are closer to each other in the hierarchy than unrelated donors. When computing the distances using only the V usage (KL distance), the clusters are shuffled, with some distinction of twins but no clear separation of hosts (Fig. 4B). To quantify the distance method that best characterizes hosts, we tested the hierarchical clusters host ID frequency and time-points entropies for each method (Fig. 4C, D). All AEs produce a better detection of hosts (lower entropy) than ED and KL methods, and the basic autoencoder combined with V (VE) without the frequencies is the best method among all models (Fig. 4C). However, none of the methods separates samples by time-points (Fig. 4D).

ELATE with KDE can be used for supervised sequence classification.

Another important application of AE is sequence classification. A classifier was built on top of the AE. The classifier is a combination of the encoder with three fully connected layers, which replace the decoder (Fig. 1). After the autoencoder was trained to predict the vector representation of the CDR3 set, the classifier layers are trained using a binary cross-entropy loss on the class detection problem. Such a classifier was built and trained for each of our four autoencoders. Each classifier was trained as one vs all classifier (i.e. for each class, this class vs all others), for each of the categories (Table 1). All test set AUC values are shown in Table 2.

We compared the first two datasets with DeepTCR results for a single sequence classifier. When analyzing GLIPH and Benny Chain datasets, no Vs are provided, so there are no models with the combination of V. As for the last three datasets (Benny Chain, vaccine and cancer), we have no comparison since there are no previous classifications for these repertoires. Deep-

TCR's based classifiers typically outperform ELATE on GLIPH. However, ELATE based classifier outperforms the Deep-TCR's sequence classifier when classifying TRP2 and SIY samples (ImmunoMap) with an average AUC of 0.81 vs 0.71. ELATE also performs best when handling TCR samples taken from healthy hosts and hosts with cancer, with an average AUC of 0.925. Here, different combinations of the autoencoder show different results in different datasets and there is no clear pattern or preference for any model.

Self-density estimate using ELATE

Finally, we checked whether the AE can be used to determine how dense is a given sample. This can be defined by the average similarity between TCR in a dataset, or alternatively the frequency of other TCRs from the same dataset around each TCR. We analyzed again the Benny Chain data set of TCR repertoires, and compared alpha to beta chains, naïve to memory and CD4 to CD8 cells self-densities. The self-density of a given sample S is formally defined as the KDE based density between S and itself, where each TCR is ignored for the computation of the density at its own projection. The average density of all groups was normalized to 0 over all projections for the ease of presentation (Figure 5).

We computed the log of the self KDE values among all samples for the two autoencoders (E, EM), with or without the frequencies, alpha chains have a lower self-density than beta chains. This is surprising given the theoretically much larger beta repertoire. When the embedding is used, a clear hierarchy can be observed alpha is less dense than beta, CD4 is on average less dense than CD8 and naïve are slightly less dense than memory. In the encoder or when the clone size is used, the same order is kept, but with a much noisier separation. Thus, again the embedding seems to be the method to represent receptors as real-valued vectors.

Discussion

Characterization of the repertoire can serve as an indicator for immune monitoring and prognosis assessment for different diseases[11]. Next-generation sequencing (NGS) allows

massive analysis of the TCR repertoire with high-throughput and provides descriptive and comparative analysis for million TCRs. The repertoire can analyzed through different features such as the CDR3 set or V gene usage and more. However, current methods of analyzing the repertoire are focused on one measure or another. We have here proposed ELATE a method to integrate them. ELATE, is a projection of TCRs that combines different aspects of the repertoire into a coherent real-valued representation of each clone. ELATE conserves important features of the repertoire. It can be used with unsupervised and supervised methods to distinguish different groups and assign a sequence to a group. ELATE is highly efficient and can be applied to repertoires with millions of clones using standard hardware.

Representing clones as vectors in real space has many advantages, including the possibility of measuring the repertoire density to detect sparse vs dense regions, computing the similarity between repertoires, using clones for machine learning applications, which are often based on real-valued input and visualizations. We have here shown that ELATE can indeed perform these different tasks, and provide important insight on the repertoire, including among many others:

A) The effect of amino acids in the CDR3 region on the repertoire density (with a decreasing density as a function of the number of Glycines or Cysteins), or the increased density around TCR recognizing cognate antigens.
B) The possibility of classifying single clones to specific cognate antigens or specific compartments.
C) The estimate of the repertoire density and the difference between the density of the repertoire in naïve and memory compartments.
D) The similarity between repertoire and showing that samples from different compartments and time points of the same donor are much more similar to other samples from the same donor than to any other sample. Samples of twins are much more similar than samples of non-twins.

While we have shown some typical applications of ELATE, the range of possible applications is much greater. Each flavor of ELATE has advantages and limitations. Distance maintaining versions are better for density estimates but produce worse sequence reproduction. Adding the V gene to the encoder improves the quality of clone classification but further reduces the reproduction accuracy. Incorporating the clone frequency helps to distinguish between the average density of some compartments, but often ruins the distance definition between compartments. Thus, for each application of ELATE, one must consider which element should be introduced.

We have used standard parameters for ELATE to ensure optimal reconstruction in the vanilla flavor of ELATE (AE). One may further improve the accuracy of different tasks by tuning the projection parameters for this specific task. However, we suggest that using a uniform projection within and between experiments is important to allow for a consistent method to compare repertoires among different settings.

A previous variational autoencoder (VAE) has been proposed in the literature - DeepTCR[12]. The autoencoder used here is different than the VAE used in DeepTCR. The main differences between ELATE and DeepTCR are that in ELATE all CDR3 sequences are converted to one-hot vectors as opposed to DeepTCR, where the sequences are converted to vectors using a trainable embedding layer. Given the wide CDR3 length distribution, ELATE uses right zero-padding to combine different lengths, and a novel loss function to optimize the projection. Finally, and most crucially, ELATE conserves distances, in contrast with all existing methods, allowing for the usage of KDE methods on the projection. At the technical level, ELATE uses only fully connected layers, while DeepTCR uses convolutional layers.

Methods

Studied Samples

Six previously published datasets with different categories were collected to test ELATE (Table 3):

- GLIPH (Glanville) [22]. TCR sequences were taken from T-cell binding to 7 different target peptides. Each peptide has one sample of TCRs with a high probability of sharing specificity to it. The TCRs from different individuals were clustered using the GLIPH algorithm.
- ImmunoMap (Sidhom) [23]**.** CD8 T cells responding to Kb-TRP2, a shared self-peptide tumor antigen, and Kb-SIY, a model foreign-antigen, in naïve and tumor-bearing B6 mice.
- TILs (Tumor-infiltrating lymphocytes) Immunotherapies (Rudqvist) [30]**.** Samples were taken from cohorts of tumor-bearing mice treated with radiotherapy (RT), anti–CTLA-4 (9H10) and a combination of them.
- Benny Chain[24]. Peripheral blood mononuclear cells (PBMC) of three donors were sorted into memory (central memory, CM; effector memory, EM; effector memory RA-expressing revertants, EMRA) and naïve CD4+ and CD8+ populations, alpha and beta chains.
- RF Vaccine[25]. Yellow fever vaccine (YFV) responding TCRs in three pairs of identical twin donors (P1, P2, Q1, Q2, S1, S2) at different time-points.
- Cancer[31]. A collection of TCRs from human hosts with any cancer and healthy hosts downloaded from the Immune Epitope Database (IEDB) (See Github for fasta file of healthy and cancer TCRs)**.**
- Not all datasets contain resolved Vs or clone frequencies. All categories, number of samples, and whether V and frequencies are available are summarized in Table 3 All datasets are available at https://github.com/louzounlab/Autoencoder.

Data Curation

TCR sequencing files were collected from the sources above. The samples were parsed to compositions of CDR3 Amino Acid sequences and when available the V usage and the frequency per clone. Sequences with non-IUPAC letters (X, *, _, #) were ignored. For each dataset, we used the pre-processing performed in the original manuscript. The J usage was ignored here for the sake of clarity. It can be incorporated the same way as V.

Glossary

Table 4 is a detailed glossary of all technical terms used in the analysis.

Data Preprocessing

The full flow of analysis is described in figure 1. Since TCRs have varying lengths of CDR3 sequences, right zero-padding was added to all TCRs to generate a constant length representation. A stop signal (!) was first added at the right end of each sequence of length *l*. The sequence was then converted to a one-hot vector. Each character had a corresponding 21 dimensions one-hot vector (for 20 possible amino acids (AA) and an additional position for the stop signal). The one hot was right zero-padded to produce a constant length representation (Fig. 1). Note that other alignment options could be considered, such as performing a multiple sequence alignment on the CDR3 sequences before the encoding. However, the proposed methods provided the highest accuracy on average on the tested samples.

When combining V usage with the CDR3 sequences, the V genes were represented as n_V dimensional one-hot vectors, where n_V is the number of all possible Vs in a given dataset (each dataset has a different number of different recognized Vs based on the algorithm used for the clone analysis and the sequencing length). The CDR3 and V one-hot vectors were then concatenated. The final representation is the concatenated one-hot vectors in this order: AA_1, AA_2, …., AA_l, stop codon, zero-padding, and (optionally) V. The clone size was never used for the training of the autoencoder.

Autoencoder (AE) training

The AE of ELATE has three internal fully connected symmetric encoder and decoder layers of 300, 100, and 30 nodes. Each layer has a dropout rate of 0.1 and an 'Elu' activation function. Beyond that, there are three additional layers in the decoder (Fig. 1A):

A) A reshape layer (to n/21 x 21) that only changes the output vector to a matrix, where each row represents a position in the CDR3, or the V gene (Technically, the V is represented as different vectors, since it has a different dimension than the AA).
B) A fully connected layer with 'softmax' activation function on each row separately
C) Another reshape layer (back to nx1 dimensions).

The AE was trained with an 'Adam' optimizer with a learning rate of 0.0001 and a Mean Square Error (MSE) loss (Eq. 1). The number of epochs for the training varied for different datasets, where N is the number of samples in the training set.

$$1.loss_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i)^2$$

Training AE to maintain the distance

To combine the original CDR3s distances in the training process, we first computed the NxN size distance matrix D_N of all Euclidean distances among all pairs of one-hot representation of TCRs from all samples in the training set. Then, batches of *b* samples were generated from the training set along with their corresponding *b* size distances matrices D_b produced from D_N.

Then, to conserve the distances in the embedded space, the Euclidean distances between pairs of projections and the original Euclidean distance between the original pairs in the input set (Eq. 2). was added to the loss function ($loss_{DIS}$).

$$2.loss_{DIS} = \sum_{i\in b}\left(\sqrt{\sum_{j>i\in b}(z_i - z_j)^2} - D_{ij}\right)^2$$

where z_i and z_j are the embedded vectors of x_i and x_j and D_{ij} is the Euclidean distance between x_i and x_j.

Autoencoder accuracy

To test the AE prediction, we used the fraction of sequences properly reconstructed, instead of the loss function used for the learning process (See figure 1B). The reconstructed sequence was produced by replacing the softmax by a rigid max in layer B above, and a translation of the one-hot back to a sequence, stopping at the leftmost stop signal or at the end of the CDR3. All positions beyond the leftmost stop signal were ignored. If the length of the reconstructed vector was different than the length of the original vector, the entire reconstruction was defined as an error. Otherwise, the error level was defined as the number of positions differing between the original and reconstructed sequence. Thus an error rate of 15 % for one mismatch implies that 15 % of all sequences either had a wrong length or the proper length and one mismatch.

Training Classifier

The classifier is composed of the previously described encoder, combined with four fully connected layers with 30, 15, 10 and 1 dimensions, with a dropout of 0.1 after the second and third layers, and a *Tanh* activation-function except for the last layer, where we use a sigmoid. The classifier is trained with similar parameters as the encoder and a cross-entropy loss (Eq. 3) (Fig. 1A). When the classifier was trained, the encoder weights were fixed. For multi-class, the encoder was trained as one vs all.

$$3.loss_{BinaryCrossentropy} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i))$$

The classifier was trained as one vs all classifier, for each of the categories in the various datasets.

The TCRs were divided into a training fraction composed of 500 TCR CDR3 sequences from each sample and all the remaining CDR3 sequences (the vast majority) as a test fraction. Hyperparameter tuning was performed using a single dataset and the E encoder.

KL Similarity

When analyzing only V gene usage, the V-gene distributions were computed for a random set of *R* crsprinnlones from each sample. Then, the distance between the distribution of a pair of samples p and q was defined to be their KL divergence [32] (Eq. 4).

$$4. D(p \,||\, q) = \sum_{v_i \in V} p(v_i) \log(\frac{p(v_i)}{q(v_i)})$$

ED Similarity

A two-sample edit distance was determined as the average of all minimum values per row in *E*, where *E* is a *RXR* distance matrix of the edit distances (Levenshtein distance) of R random clones in both samples. Levenshtein distance represents the number of insertions, deletions, and substitutions required to change one word to another[29]. When computing the self-distance, the diagonal values were ignored. Note that the edit distance is different than the one-hot distance.

KDE Similarity

Given two of projections of samples, S1={x_1, x_2, …, x_R} and S2={y_1, y_2, …, y_R}sprins[sprins (or a single projection if S1 and S2 are the same), we estimated the local density using a Kernel Density Estimator KDE[18]. However, in contrast with standard KDE, we here found the density of each projection in S1 with respect to the projections in S2 (Eq. 5).

$$5. f_h(x) = \frac{1}{Rh} \sum_{i=1}^{R} K(\frac{x - y_i}{h})$$

where K is the kernel, and $h > 0$ is a smoothing parameter called the *bandwidth*. x is the projection of a TCR in sample S1, while y are all projections in sample S2

We can define the average density of the projections S1 in S2 as the average of densities for all vectors in S1. When measuring the density of samples in S1 vs S2 in combination with the frequencies, each TCR clone was multiplied by its clone size (Eq. 6).

$$6. f_h(x) = \frac{1}{Rh} \sum_{i=1}^{R} freq_{x_i} K(\frac{x - x_i}{h})$$

Also, the new representation of the data allowed us to define for the first time the average self-density of a given repertoire, as the density when S1 and S2 are the same, and ignoring for each projected sequence its own contribution to its density. Here, we used Gaussian kernel and h = 1.06*STD(S)*len(S)**(-1/5) [33].

Statistical analysis

The difference between the various methods in our analysis was tested using two-way ANOVA[34],and Tukey's test. An ANOVA test was performed with all autoencoder models. Then, Tukey's test was performed between all methods and categories pairs in order to find which are significantly different. The clustering of samples was done using Hierarchical cluster analysis (HCA) [35] using average linkage on the KL, ED and KDE distances matrices.

Visualization with TSNE

T-distributed Stochastic Neighbor Embedding (TSNE)[36] is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability. We here use TSNE to visualize the embedded vectors of 30 dimensions in two dimensions.

Acknowledgements

This manuscript has been released as a pre-print at bioRxiv, :
https://www.biorxiv.org/content/10.1101/2020.06.12.148502v3[37]

References

1. Bassing CH, Swat W, Alt FW. The Mechanism and Regulation of Chromosomal V (D) J Recombination. Cell. 2002;109(D):45-55.

2. Benichou JIC, van Heijst JWJ, Glanville J, Louzoun Y. Converging evolution leads to near maximal junction diversity through parallel mechanisms in B and T cell receptors. Physical Biology. 2017;14(4):045003-. doi: 10.1088/1478-3975/aa7366.

3. Benichou J, Ben-Hamo R, Louzoun Y, Efroni S. Rep-Seq: uncovering the immunological repertoire through next-generation sequencing. Immunology. 2012;135(3):183-91. doi: 10.1111/j.1365-2567.2011.03527.x.

4. Heather JM, Ismail M, Oakes T, Chain B. High-throughput sequencing of the T-cell receptor repertoire: pitfalls and opportunities. Briefings in bioinformatics. 2018;19(4):554-65. doi: 10.1093/bib/bbw138.

5. Muraro PA, Robins H, Malhotra S, Howell M, Phippard D, Desmarais C, et al. T cell repertoire following autologous stem cell transplantation for multiple sclerosis. The Journal of clinical investigation. 2014;124(3):1168-72. doi: 10.1172/JCI71691.

6. Kedzierska K, Izraelson M, Tatiana O, Egorov ES, Kasatskaya SA, Ekaterina V, et al. Origin and evolution of the T cell repertoire after posttransplantation cyclophosphamide. 2016;9(June):133-44. doi: 10.3389/fimmu.2018.01453.

7. Pogorelyy MV, Fedorova AD, McLaren JE, Ladell K, Bagaev DV, Eliseev AV, et al. Exploring the pre-immune landscape of antigen-specific T cells. Genome Medicine. 2018;10(1):68-82.

8. Gerlinger M, Quezada SA, Peggs KS, Furness AJS, Fisher R, Marafioti T, et al. Ultra-deep T cell receptor sequencing reveals the complexity and intratumour heterogeneity of T cell clones in renal cell carcinomas. The Journal of pathology. 2013;231(4):424-32. doi: 10.1002/path.4284.

9. Rosati E, Dowds CM, Liaskou E, Henriksen EKK, Karlsen TH, Franke A. Overview of methodologies for T-cell receptor repertoire analysis. BMC Biotechnology. 2017;17(1). doi: 10.1186/s12896-017-0379-9.

10. Alves Sousa AdP, Johnson KR, Ohayon J, Zhu J, Muraro PA, Jacobson S. Comprehensive Analysis of TCR-β Repertoire in Patients with Neurological Immune-mediated Disorders. Scientific Reports. 2019;9(1). doi: 10.1038/s41598-018-36274-7.

11. Cui JH, Lin KR, Yuan SH, Jin YB, Chen XP, Su XK, et al. TCR repertoire as a novel indicator for immune monitoring and prognosis assessment of patients with cervical cancer. Frontiers in Immunology. 2018;9(NOV). doi: 10.3389/fimmu.2018.02729.

12. Sidhom J-W, Larman HB, Pardoll DM, Baras AS. DeepTCR: a deep learning framework for revealing structural concepts within TCR Repertoire. bioRxiv. 2018:464107-. doi: 10.1101/464107.

13. Farmanbar A, Kneller R, Firouzi S. RNA sequencing identifies clonal structure of T-cell repertoires in patients with adult T-cell leukemia/lymphoma. npj Genomic Medicine. 2019;4(1). doi: 10.1038/s41525-019-0084-9.

14. Kaplinsky J, Arnaout R. Robust estimates of overall immune-repertoire diversity from high-throughput measurements on samples. Nature Communications. 2016;7. doi: 10.1038/ncomms11881.

15. Sethna Z, Elhanati Y, Callan CG, Walczak AM, Mora T. OLGA: fast computation of generation probabilities of B- and T-cell receptor amino acid sequences and motifs. Bioinformatics. 2019. doi: 10.1093/bioinformatics/btz035.

16. Tickotsky-Moskovitz N, Dvorkin S, Rotkopf A, Kuperman AA, Efroni S, Louzoun Y. CDR3 and V- genes show distinct reconstitution patterns in T-cell repertoire post allogenic bone marrow transplantation. bioRxiv. 2020:2020.03.05.977900-2020.03.05. doi: 10.1101/2020.03.05.977900.

17. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504-7. doi: 10.1126/science.1127647.
18. Parzen E. On Estimation of a Probability Density Function and Mode. The Annals of Mathematical Statistics. 1962;33(3):1065-76. doi: 10.1214/aoms/1177704472.
19. Schrama D, Ritter C, Becker JC. T cell receptor repertoire usage in cancer as a surrogate marker for immune responses. Seminars in immunopathology. 2017;39(3):255-68. doi: 10.1007/s00281-016-0614-9.
20. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997;9(8):1735-80. doi: 10.1162/neco.1997.9.8.1735.
21. Springer I, Besser H, Tickotsky-Moskovitz N, Dvorkin S, Louzoun Y. Prediction of specific TCR-peptide binding from large dictionaries of TCR-peptide pairs. Frontiers in immunology. 2020;11:1803.
22. Glanville J, Huang H, Nau A, Hatton O, Wagar LE, Rubelt F, et al. Identifying specificity groups in the T cell receptor repertoire. Nature. 2017;547(7661):94-8. doi: 10.1038/nature22976.
23. Sidhom J-W, Bessell CA, Havel JJ, Kosmides A, Chan TA, Schneck JP. ImmunoMap: A Bioinformatics Tool for T-cell Repertoire Analysis. Cancer immunology research. 2018;6(2):151-62. doi: 10.1158/2326-6066.CIR-17-0114.
24. Oakes T, Heather JM, Best K, Byng-Maddick R, Husovsky C, Ismail M, et al. Quantitative Characterization of the T Cell Receptor Repertoire of Naïve and Memory Subsets Using an Integrated Experimental and Computational Pipeline Which Is Robust, Economical, and Versatile. Frontiers in immunology. 2017;8:1267-. doi: 10.3389/fimmu.2017.01267.
25. Pogorelyy MV, Minervina AA, Touzel MP, Sycheva AL, Komech EA, Kovalenko EI, et al. Precise tracking of vaccine-responding T cell clones reveals convergent and personalized response in identical twins. Proceedings of the National Academy of Sciences of the United States of America. 2018;115(50):12704-9. doi: 10.1073/pnas.1809642115.
26. Toledano A, Elhanati Y, Benichou JIC, Walczak AM, Mora T, Louzoun Y. Evidence for Shaping of Light Chain Repertoire by Structural Selection. Frontiers in Immunology. 2018;9:1307-. doi: 10.3389/fimmu.2018.01307.
27. Pickman Y, Dunn-Walters D, Mehr R. BCR CDR3 length distributions differ between blood and spleen and between old and young patients, and TCR distributions can be used to detect myelodysplastic syndrome. Physical biology. 2013;10(5):056001-. doi: 10.1088/1478-3975/10/5/056001.
28. Elhanati Y, Sethna Z, Callan CG, Mora T, Walczak AM. Predicting the spectrum of TCR repertoire sharing with a data-driven model of recombination. Immunological reviews. 2018;284(1):167-79. doi: 10.1111/imr.12665.
29. Postow MA, Manuel M, Wong P, Yuan J, Dong Z, Liu C, et al. Peripheral T cell receptor diversity is associated with clinical outcomes following ipilimumab treatment in metastatic melanoma. Journal for ImmunoTherapy of Cancer. 2015;3(1). doi: 10.1186/s40425-015-0070-4.
30. Rudqvist N-P, Pilones KA, Lhuillier C, Wennerberg E, Sidhom J-W, Emerson RO, et al. Radiotherapy and CTLA-4 Blockade Shape the TCR Repertoire of Tumor-Infiltrating T Cells. Cancer immunology research. 2018;6(2):139-50. doi: 10.1158/2326-6066.CIR-17-0134.
31. Dvorkin S. Autoencoder/cancer data IEDB at master · shiritdvir/Autoencoder · GitHub. 2020.
32. Kullback S, Leibler RA. On Information and Sufficiency. The Annals of Mathematical Statistics. 1951;22(1):79-86. doi: 10.1214/aoms/1177729694.
33. Silverman BW. Density estimation for statistics and data analysis: CRC press; 1986.
34. Sawyer SF. Analysis of Variance: The Fundamental Concepts. Journal of Manual & Manipulative Therapy. 2009;17(2):27E-38E. doi: 10.1179/jmt.2009.17.2.27e.

35. Johnson SC. Hierarchical clustering schemes. Psychometrika. 1967;32(3):241-54. doi: 10.1007/BF02289588.
36. Maaten Lvd, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(Nov):2579-605.
37. Dvorkin S, Levi R, Louzoun Y. Autoencoder based local T cell repertoire density can be used to classify samples and T cell receptors. bioRxiv. 2020:2020.06.12.148502. doi: 10.1101/2020.06.12.148502.

Figure captions

Fig. 1. A. *Autoencoder description. The autoencoder is composed of two models: encoder and decoder. Encoder's input: a vector of n dimensions of concatenated one-hot (oh) vectors, representing a CDR3 sequence. Encoder's output (decoder's input): the embedded representation with 30 dimensions. Decoder's output: the reconstructed vector. The autoencoder is trained by minimizing the reconstruction MSE (Mean Squared Error). The classifier is a combination of the encoder (with fixed weights) and 4 more fully connected layers. The classifier is trained by minimizing the BCE loss (Binary Cross Entropy).* ***B.*** *Data preprocessing and distance maintaining. Encoder's input: Two 1× n one-hot representations of TCR sequences (x_1, x_2). Each amino acid is represented by a 21 long one hot representation. The last value is for the stop signal. Each sequence ends with a stop signal. The encoder's output and decoder's input is a 1× 30 embedded vector with lower dimension z for each input sequence (z_1, z_2). Decoder's output: a 1× n reconstructed one-hot vector for each input (x'_1, x'_2). In the EMB model, an additional loss term is added to reduce the difference between the original distances (Euclidean distances between every x_i and x_j) and embedded vectors distances (euclidean distances between every z_i and z_j). The distance is computed on the output vector. However, when reporting the results, the definition of the accuracy is based on a vector after the max argument on each 21 positions is performed.*

Fig 2. A. *Edit distances distribution. Histogram of edit distances between all pairs of CDR3 sequences from two samples. Few sequences have 2 or fewer mismatches.* ***B.*** *Models glossary. The table describes the model and model acronym used in all the following figures and in the text.* ***C.*** *E accuracy. All accuracies of E for the different datasets with no mismatches (fully matched, red), 1 mismatch (orange), and 2 mismatches (blue).* ***D.*** *VE accuracy. All accuracies of VE for the different datasets (except for datasets without Vs).* ***E.*** *Encoder and CDR3 distances correlation. Pairwise Euclidean distances in the set of CDR3 one-hot vectors (Xs) in the x axis and pairwise Euclidean distances in the embedded set (Zs) predicted by E in the y axis. The legend contains the Spearman correlation between the axes.* ***F.*** *Encoder's distances correlation when combined with distances. Pairwise Euclidean distances in the set of CDR3 one-hot vectors (Xs) in the x axis and pairwise Euclidean distances in the embedded set (Zs) predicted by EM in the y axis.* ***G.*** *EM accuracy. All accuracies of EM for the different datasets.* ***H.*** *VEM accuracy. All accuracies of VEM for the different datasets.*

Fig. 3. A. *Encoder's projection TSNE (Benny Chain dataset). TCRs with cysteine are colored in red.* ***B.*** *Encoder's projection TSNE (Benny Chain dataset). Colored by CDR3 length.* ***C.*** *Radius vs Publicity (Benny Chain dataset). The x axis is the number of samples containing this TCR clone and the y axis is the distance from the center of all projections. The Spearman correlation is in the legend.* ***D.*** *Self-density vs Publicity (Benny Chain dataset). Axes: y- The KDE between a TCR clone and its sample, x-number of samples that contain this TCR clone.* ***E.*** *Encoder's projection TSNE (GLIPH). TCRs with specificity to TRPVTGGGAM peptide are colored in red. The other groups are colored in grey.* ***F.*** *Encoder's projection TSNE (Cancer dataset). TCRs from hosts with cancer are colored in red and TCRs from samples collected from*

healthy hosts are colored in grey. ***G.*** *Average self-density per feature. The average self-density of all 7 peptides in* GLIPH *data set and for TCRs with a specific number of G (Glycine), C (Cysteine) and P (Proline). The p-value of t-test is shown as * (<0.05), ** (<0.01), *** (< 0.001).*

Fig. 4 A. *KDE dendogram. Hierarchical clustering distances computed on the KDE distance matrix (without the clone frequencies) of the vectors samples predicted by the autoencoder combined with Vs (VE). Samples are colored by hosts and twins are colored is similar colors (P1 and P2: blue and light blue, S1 and S2: red and salmon, Q1 and Q2: spring green and lawn green).* ***B.*** *KL dendogram. Hierarchical clustering distances computed on the KL distance matrix.* ***C.*** *Clusters entropies by hosts. All hosts (P1, P2, S1, S2, Q1 and Q2) entropies through the different methods (labeled with a combination of: E= AE, EM=AE+DIS, V= with v genes, F= with frequencies).* ***D.*** *Clusters entropies by time-points. All time-points (7 D pre, 0 D post, 7 D post, 15 D post, 45 D post and 2 Y post-vaccination) entropies through the different methods.*

Fig. 5. *Encoders' self-density (Benny Chain dataset).* ***A.*** *Average KDE density within the set of vectors predicted by the basic autoencoder (E).* ***B.*** *Average KDE density within the set of vectors predicted by the basic autoencoder combined with the clone frequencies (EF).* ***C.*** *Average KDE density within the set of vectors predicted by the autoencoder combined with the CDR3 distances (EM).* ***D.*** *Average KDE density within the set of vectors predicted by the autoencoder combined with the CDR3 distances and the clone frequencies (EMF).*

Tables

Table 1. *Description of each category.*

DATA SET	TAGS	Description
GLIPH	CTELKLSDY	TCRs with shared specificity to CTELKLSDY peptide
	VTEHDTLLY	TCRs with shared specificity to VTEHDTLLY peptide
	GLCTLVAML	TCRs with shared specificity to GLCTLVAML peptide
	GILGFVFTL	TCRs with shared specificity to GILGFVFTL peptide
	TPRVTGGGAM	TCRs with shared specificity to TPRVTGGGAM peptide
	NLVPMVATV	TCRs with shared specificity to NLVPMVATV peptide
	LPRRSGAAGA	TCRs with shared specificity to LPRRSGAAGA peptide
ImmunoMap	TRP2	TCRs responding to TRP2 immunotherapy in tumor-bearing mice
	SIY	TCRs responding to SIY immunotherapy in tumor-bearing mice
Benny Chain	Memory	TCR samples from memory cells
	Naive	TCR samples from naïve cells
Vaccine	7 D Pre	TCR samples collected 7 days before immunization
	0 D Post	TCR samples collected in the day of immunization
	7 D Post	TCR samples collected 7 days after immunization
	15 D Post	TCR samples collected 15 days after immunization
	45 D Post	TCR samples collected 45 days after immunization
	2 Y Post	TCR samples collected 2 years after immunization
Cancer	Healthy	TCRs collected from healthy hosts
	Cancer	TCRs collected from hosts with different types of cancers

Table 2. *AUC values for each category for our four models (E, VE, EM, VEM) and DeepTCR.*

DATASET	CATEGORY	E AUC	VE AUC	EM AUC	VEM AUC	Deep-TCR
GLIPH	CTELKLSDY	0.4	x	0.47	x	**0.5**
	VTEHDTLLY	0.58		0.6		**0.7**
	GLCTLVAML	**0.64**		**0.64**		0.6
	GILGFVFTL	0.61		**0.67**		0.65
	TPRVTGGGAM	0.81		0.54		**0.83**
	NLVPMVATV	0.65		0.56		**0.73**
	LPRRSGAAGA	**0.58**		0.47		**0.58**
	Average	0.61		0.56		**0.65**
ImmunoMap	TRP2	0.78	0.74	0.74	**0.8**	0.71

	SIY	0.68	0.79	0.78	**0.82**	0.71
	Average	0.73	0.765	0.76	**0.81**	0.71
Benny Chain	Memory	**0.58**	x	0.57	x	x
	Naive	**0.57**		0.57		
	Average	**0.575**		0.57		
Vaccine	15 D Post	**0.54**	0.53	0.52	0.52	x
	7 D Post	**0.5**	0.49	0.48	0.48	
	2 Y Post	0.6	0.63	**0.64**	0.62	
	0 D Post	**0.49**	**0.49**	**0.49**	0.48	
	7 D Pre	**0.51**	**0.51**	0.49	0.5	
	45 D Post	0.49	**0.51**	0.5	0.48	
	Average	**0.52**	**0.52**	**0.52**	0.51	
Cancer	Healthy	0.87	0.91	**0.93**	0.91	x
	Cancer	**0.94**	**0.94**	0.86	0.91	
	Average	0.9	**0.925**	0.895	0.91	

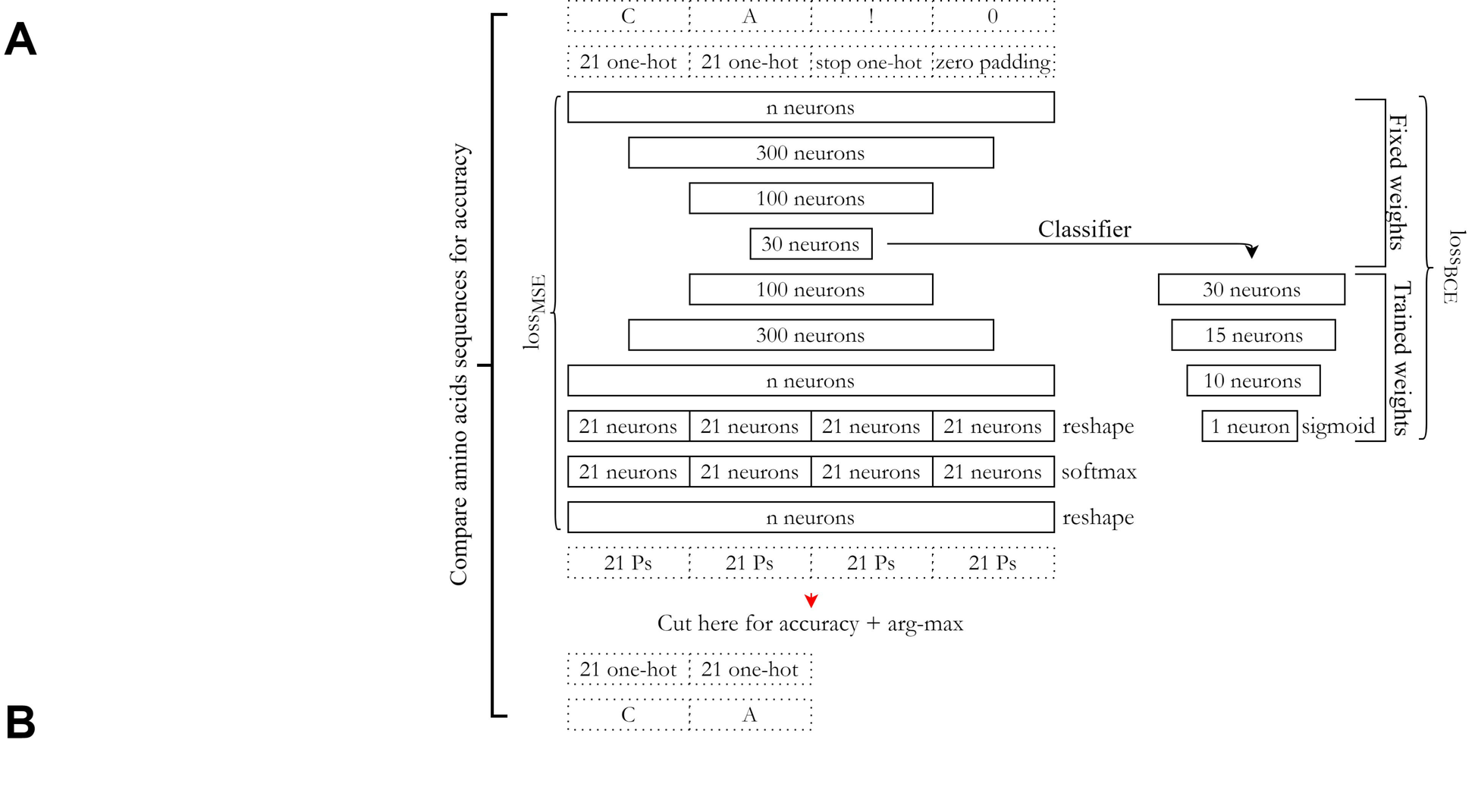
Table 3. *The table contains the dataset name, the categories in each dataset, the number of sequenced libraries for each category, the number of clones in each category, whether the dataset contains V gene usage, and whether the dataset details the clone frequencies.*

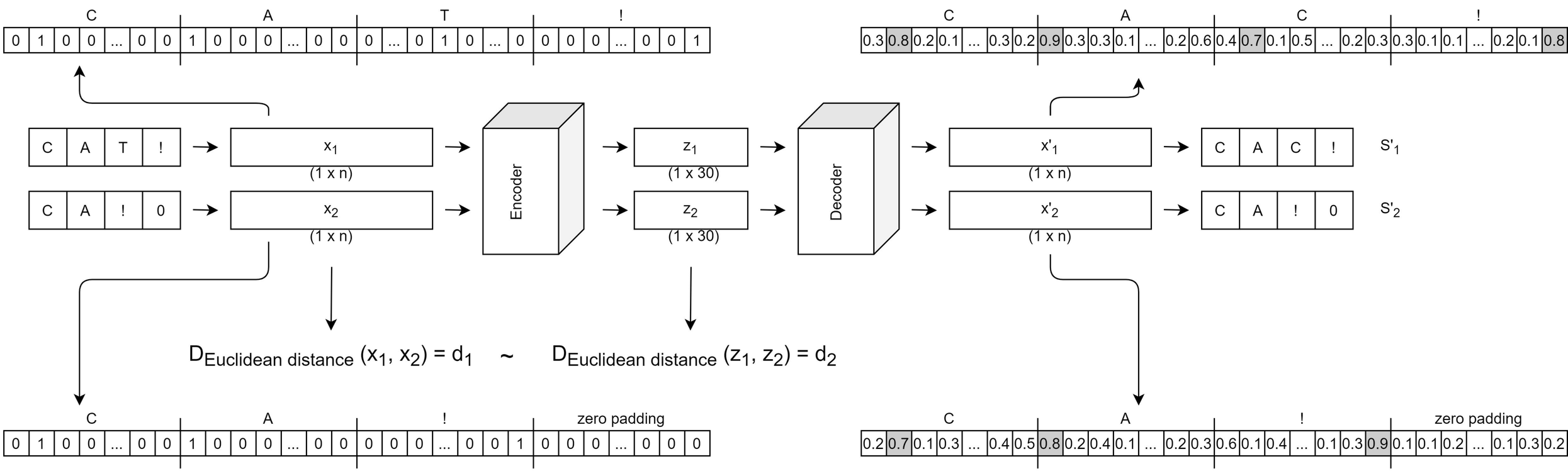
Dataset	Category	Samples	Clones	Number of different V genes	Frequencies
GLIPH	CTELKLSDY	1	25	X	X
	VTEHDTLLY	1	271		
	GLCTLVAML	1	739		
	GILGFVFTL	1	586		
	TPRVTGGGAM	1	65		
	NLVPMVATV	1	166		
	LPRRSGAAGA	1	214		
ImmunoMap	TRP2	3	488	36	V
	SIY	4	1,106		
TILs Immunotherapies	9H10	5	3,643	27	V
	RT	5	6,707		
	9H10+RT	5	7,826		
	Control	5	3,643		

Benny Chain	Memory	36	675,146	X	V
	Naive	20	1,557,365		
Vaccine	7 D Pre	17	16,786	56	V
	0 D Post	28	27,781		
	7 D Post	24	23,700		
	15 D Post	23	22,320		
	45 D Post	30	29,646		
	2 Y Post	3	2,260		
Cancer	Healthy	1	7,342	82	V
	Cancer	1	409		

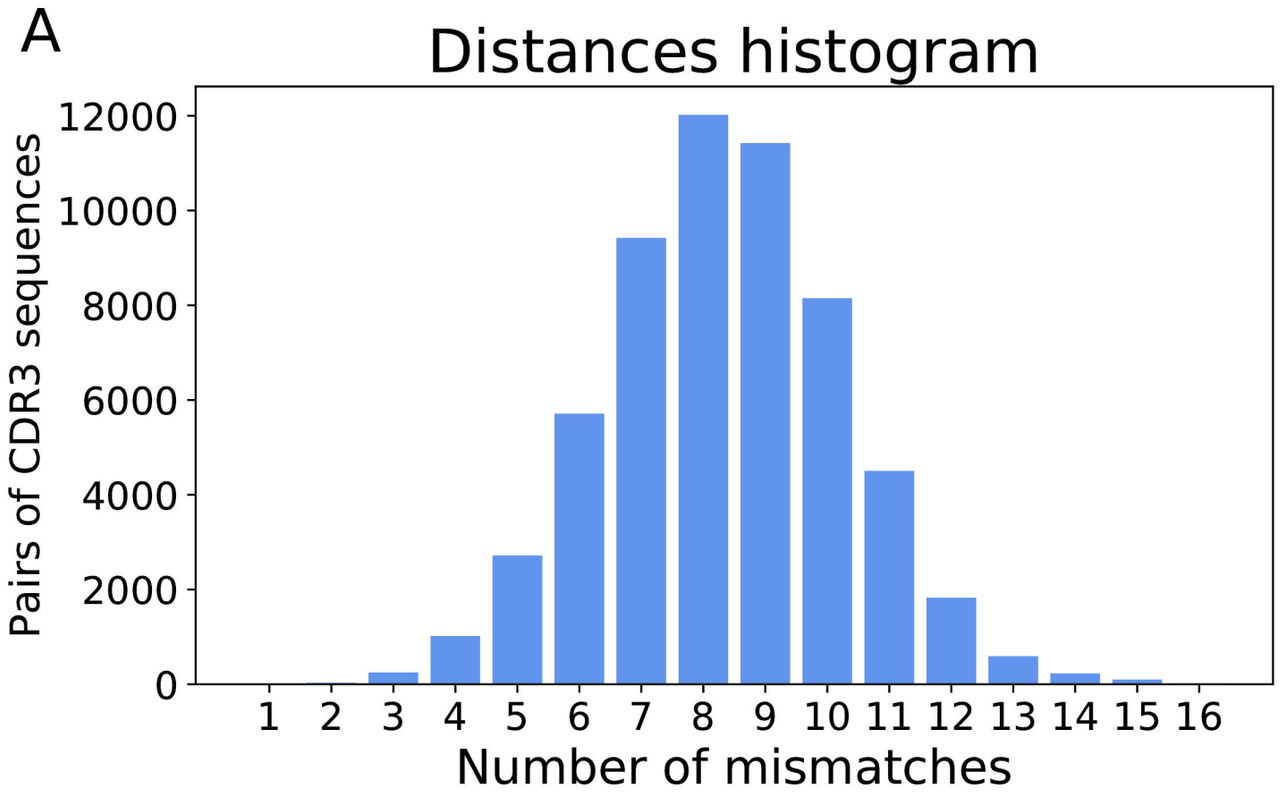
Table 4. *Glossary of terms used for the analysis.*

l	Length of CDR3 sequence
m	Maximal length of CDR3 sequence in an analyzed repertoire
n	Dimension of one-hot representation
n_V	Number of possible Vs in a repertoire
N	Size of the training set
b	Batch size
D_N	Distance matrix of the all Euclidean distances in the training set
D_b	Distance matrix of the all Euclidean distances in a batch
x	Encoder's input vector
z	Encoder's output vector- embedded vector
Z	Length of z
s	The sequence form of x
x'	Reconstructed vector from the decoder's output vector-
s'	The sequence form of x'
V	Set of all different V genes
R	Number of randomly sampled clones
E	R by R edit distance matrix


A
C
A
!
0
21 one-hot
21 one-hot
stop one-hot
zero padding
n neurons
300 neurons
100 neurons
30 neurons
Classifier
100 neurons
300 neurons
n neurons
21 neurons
21 neurons
21 neurons
21 neurons
reshape
21 neurons
21 neurons
21 neurons
21 neurons
softmax
n neurons
reshape
21 Ps
21 Ps
21 Ps
21 Ps
Cut here for accuracy + arg-max
21 one-hot
21 one-hot
C
A
Compare amino acids sequences for accuracy
loss_MSE
Fixed weights
Trained weights
loss_BCE
30 neurons
15 neurons
10 neurons
1 neuron
sigmoid


B
C
A
T
!
0 1 0 0 ... 0 0 1 0 0 0 ... 0 0 0 ... 0 1 0 ... 0 0 0 0 ... 0 0 1
C
A
C
!
0.3 0.8 0.2 0.1 ... 0.3 0.2 0.9 0.3 0.3 0.1 ... 0.2 0.6 0.4 0.7 0.1 0.5 ... 0.2 0.3 0.3 0.1 0.1 ... 0.2 0.1 0.8
C A T !
C A ! 0
x1
(1 x n)
x2
(1 x n)
Encoder
z1
(1 x 30)
z2
(1 x 30)
Decoder
x'1
(1 x n)
x'2
(1 x n)
C A C !
S'1
C A ! 0
S'2
D Euclidean distance (x1, x2) = d1 ~ D Euclidean distance (z1, z2) = d2
C
A
!
zero padding
0 1 0 0 ... 0 0 1 0 0 0 ... 0 0 0 0 0 ... 0 0 1 0 0 0 ... 0 0 0
C
A
!
zero padding
0.2 0.7 0.1 0.3 ... 0.4 0.5 0.8 0.2 0.4 0.1 ... 0.2 0.3 0.6 0.1 0.4 ... 0.1 0.3 0.9 0.1 0.1 0.2 ... 0.1 0.3 0.2

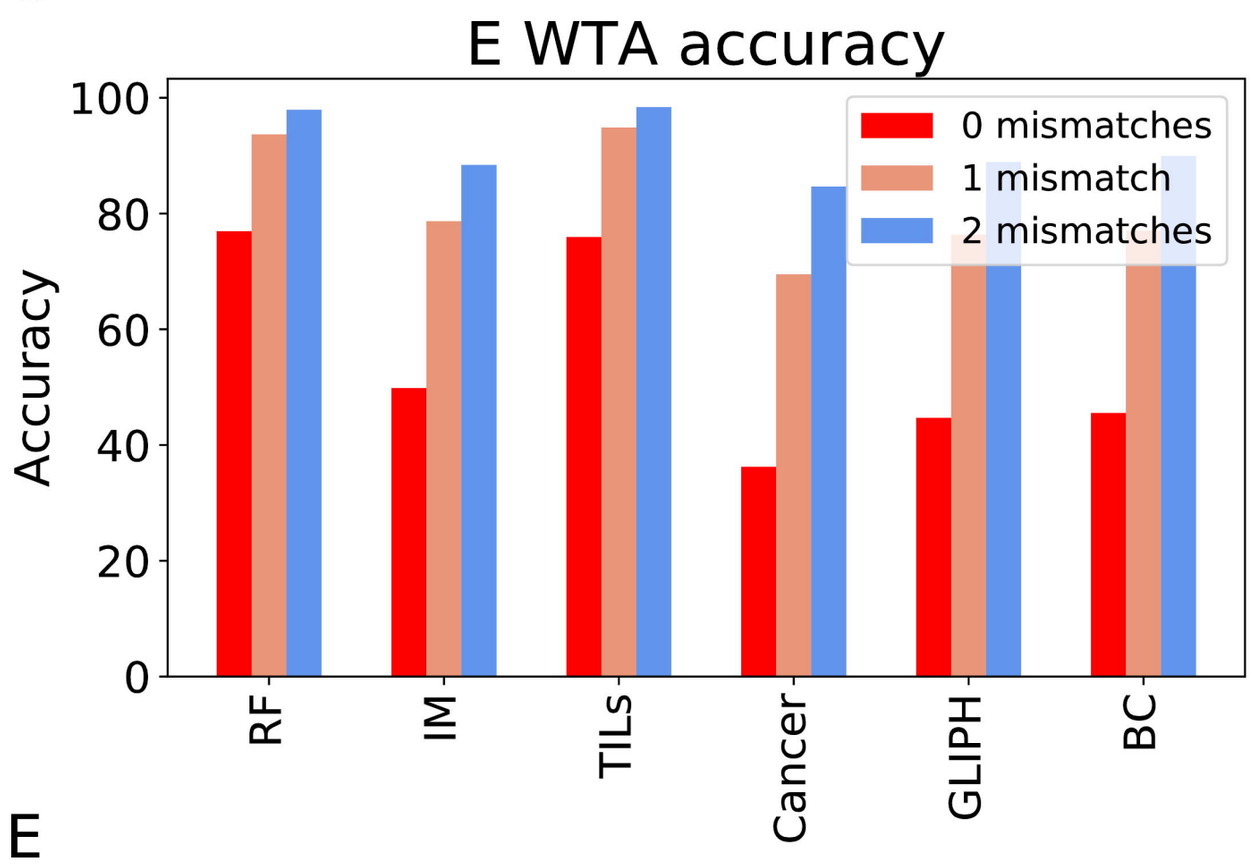
A



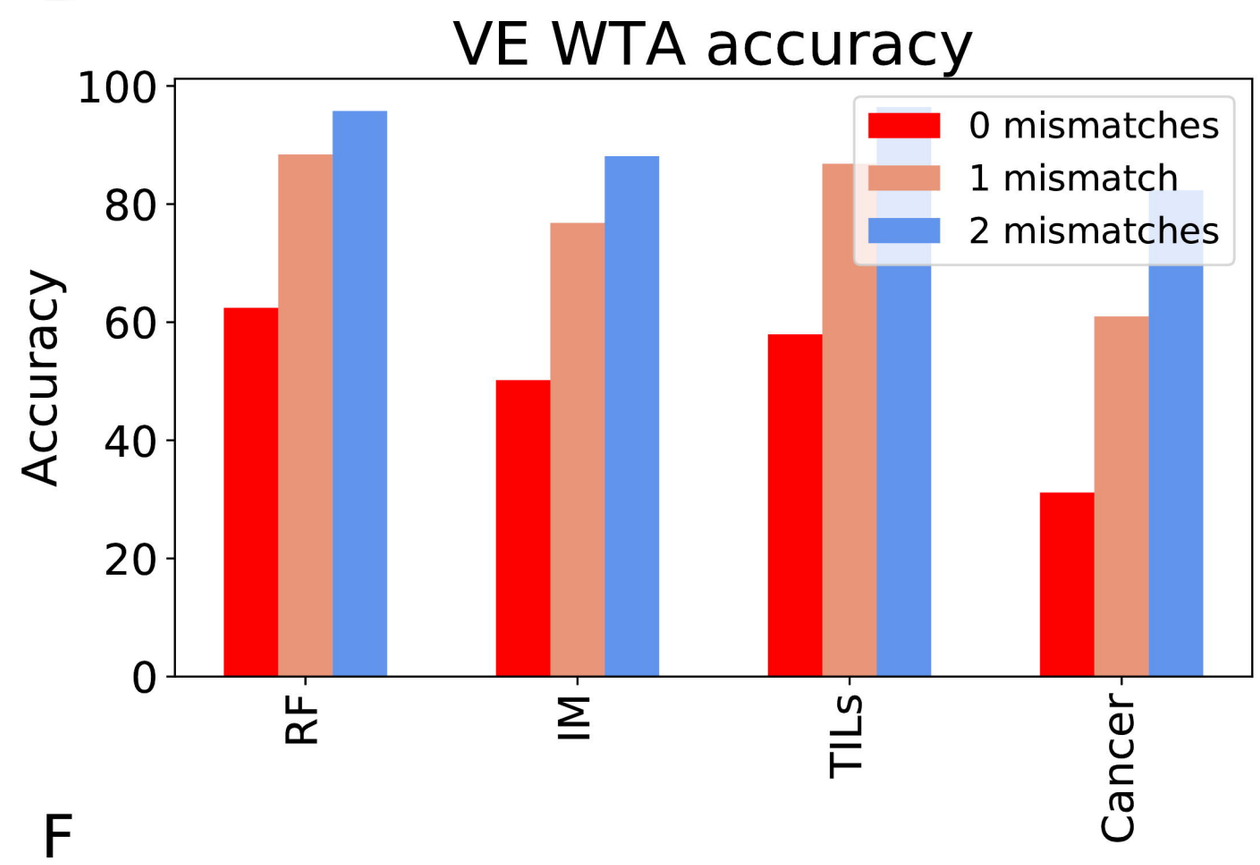
B

	Description
E	Encoder with MSE loss
EM	Embedding with MSE loss + DIS loss
VE	Encoder with MSE loss + V
VEM	Embedding with MSE loss + DIS loss + V

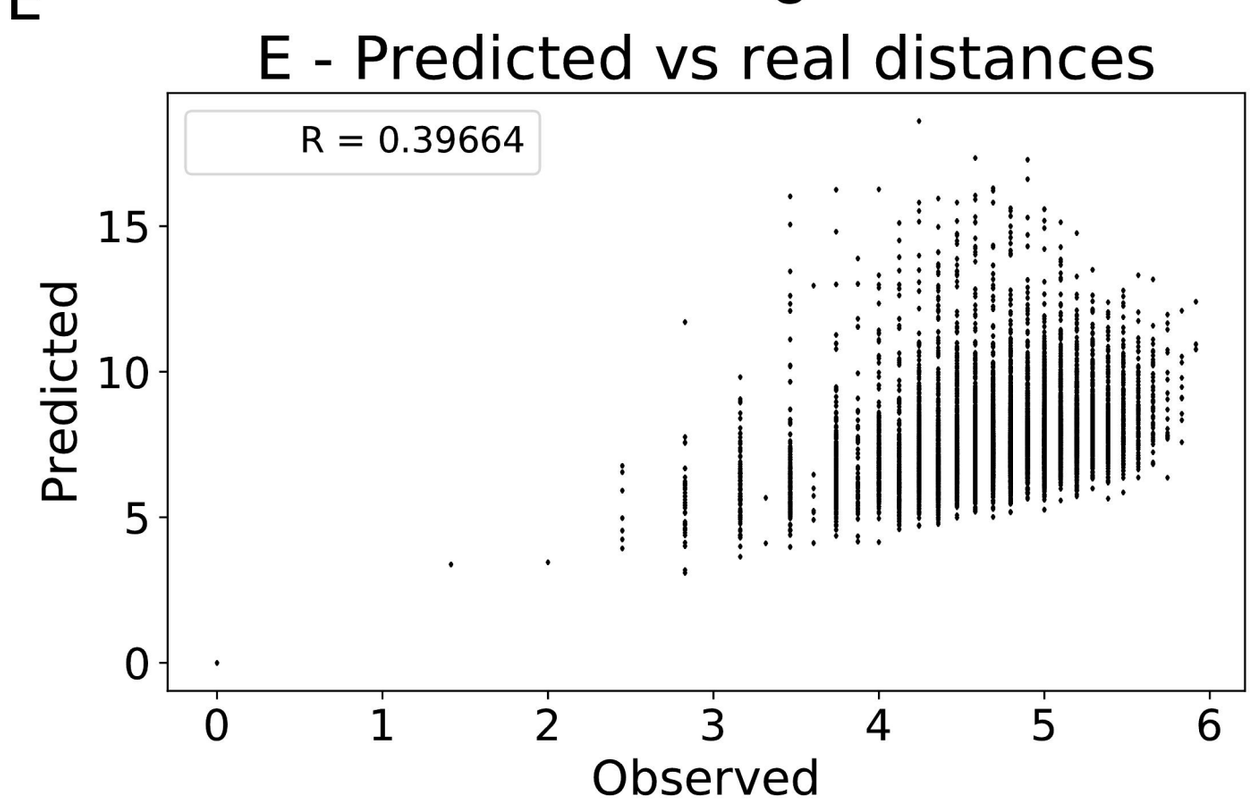
C



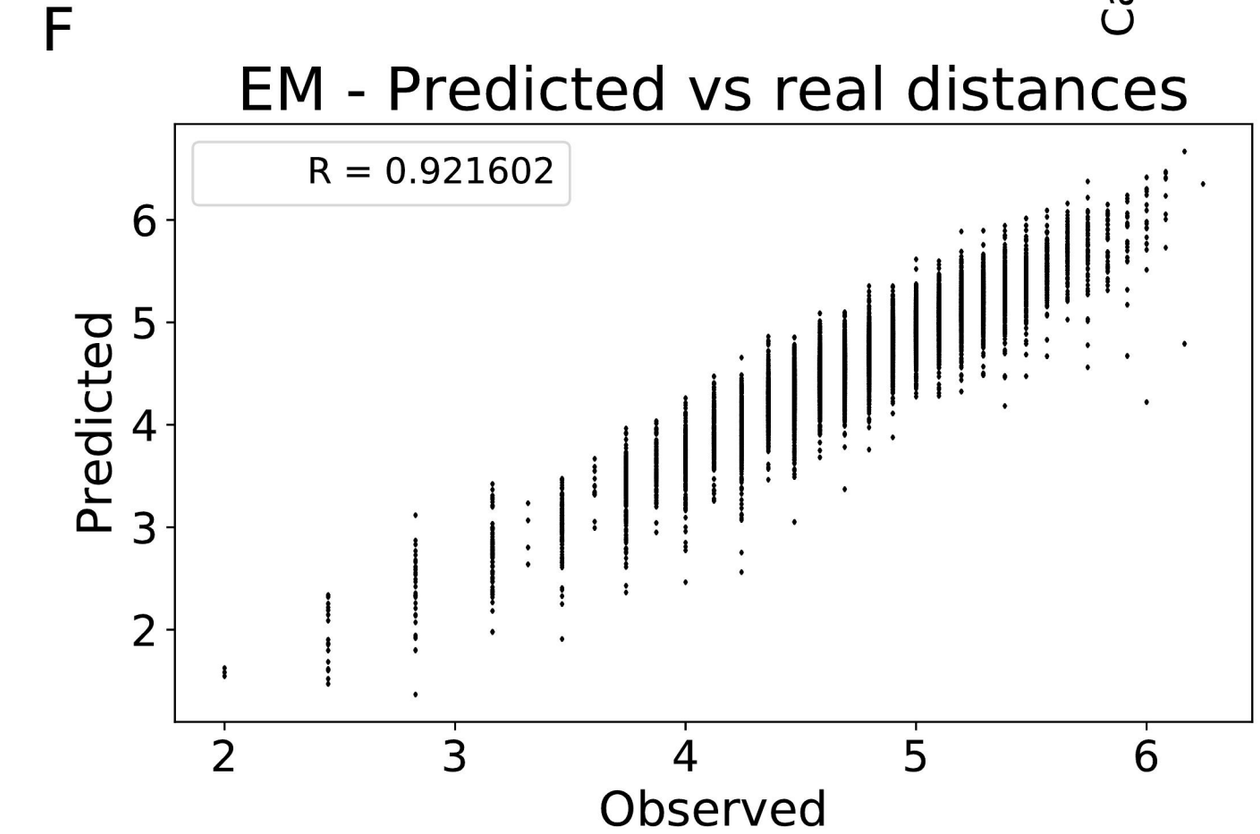
D



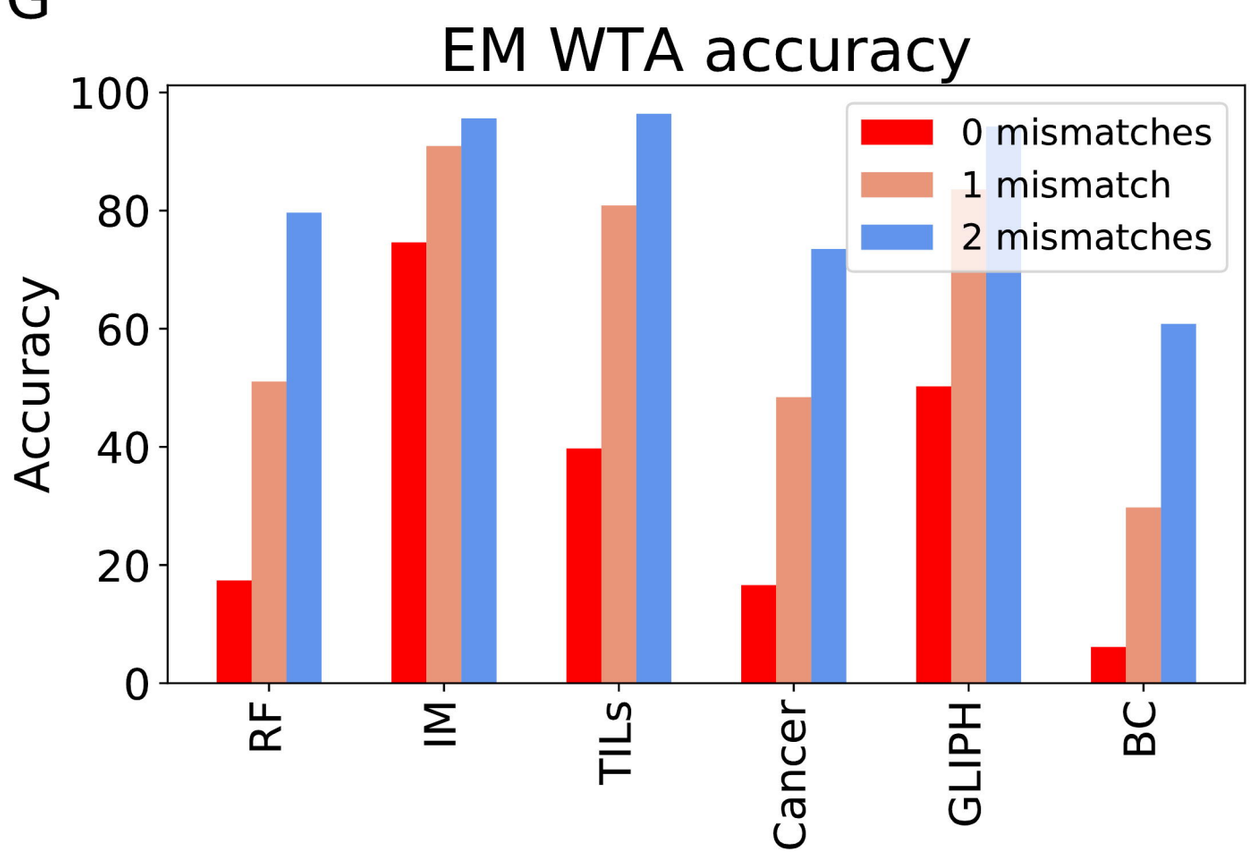
E



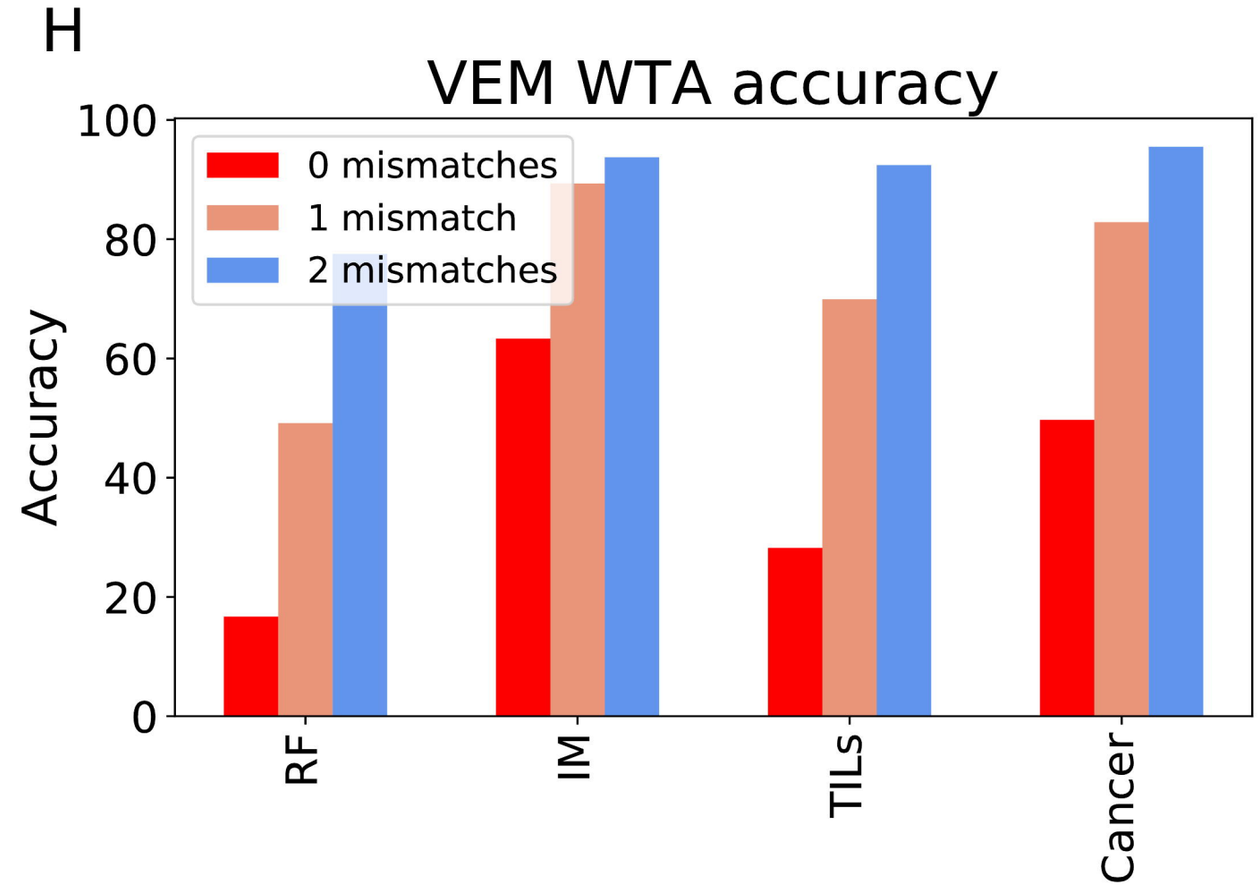
F

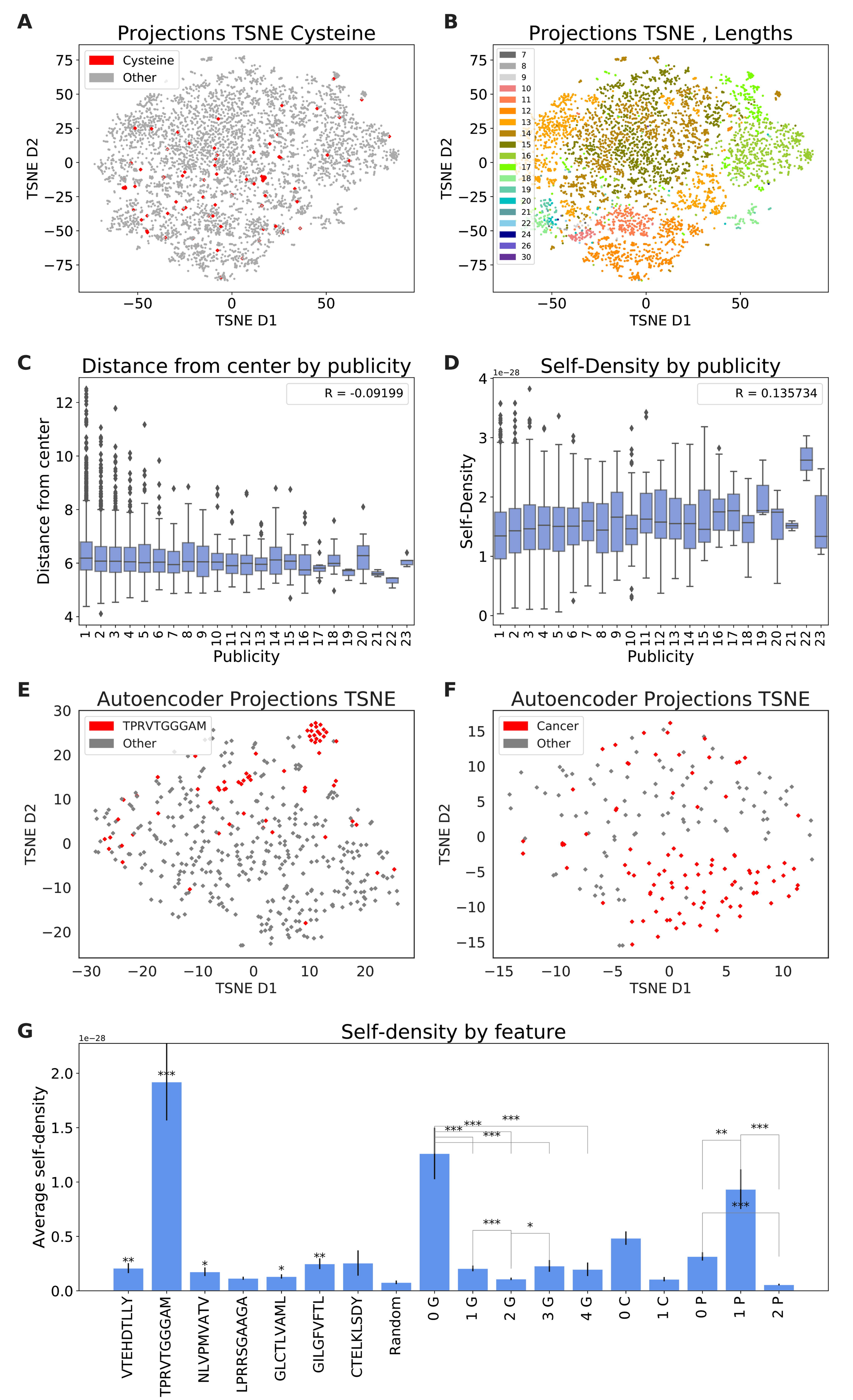


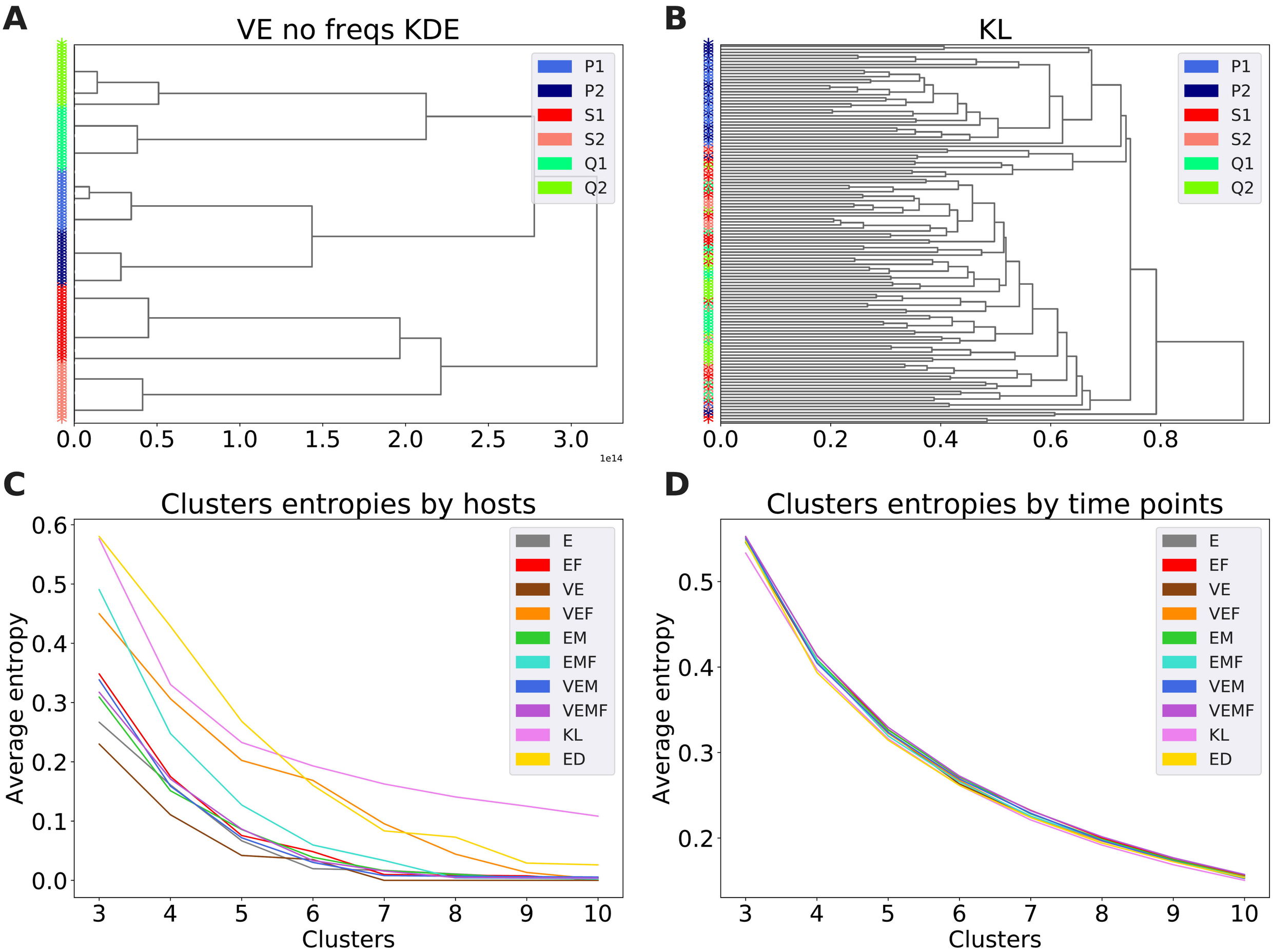
G

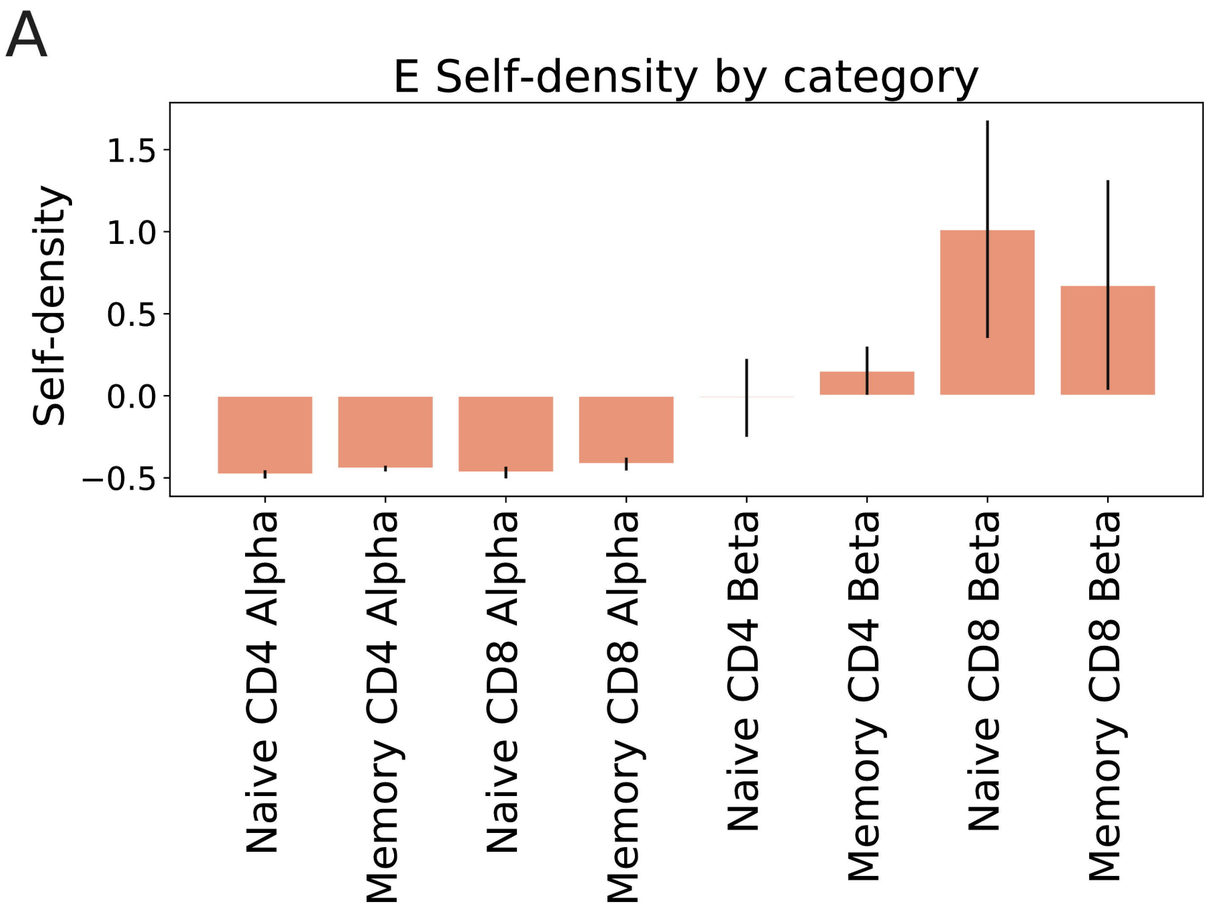


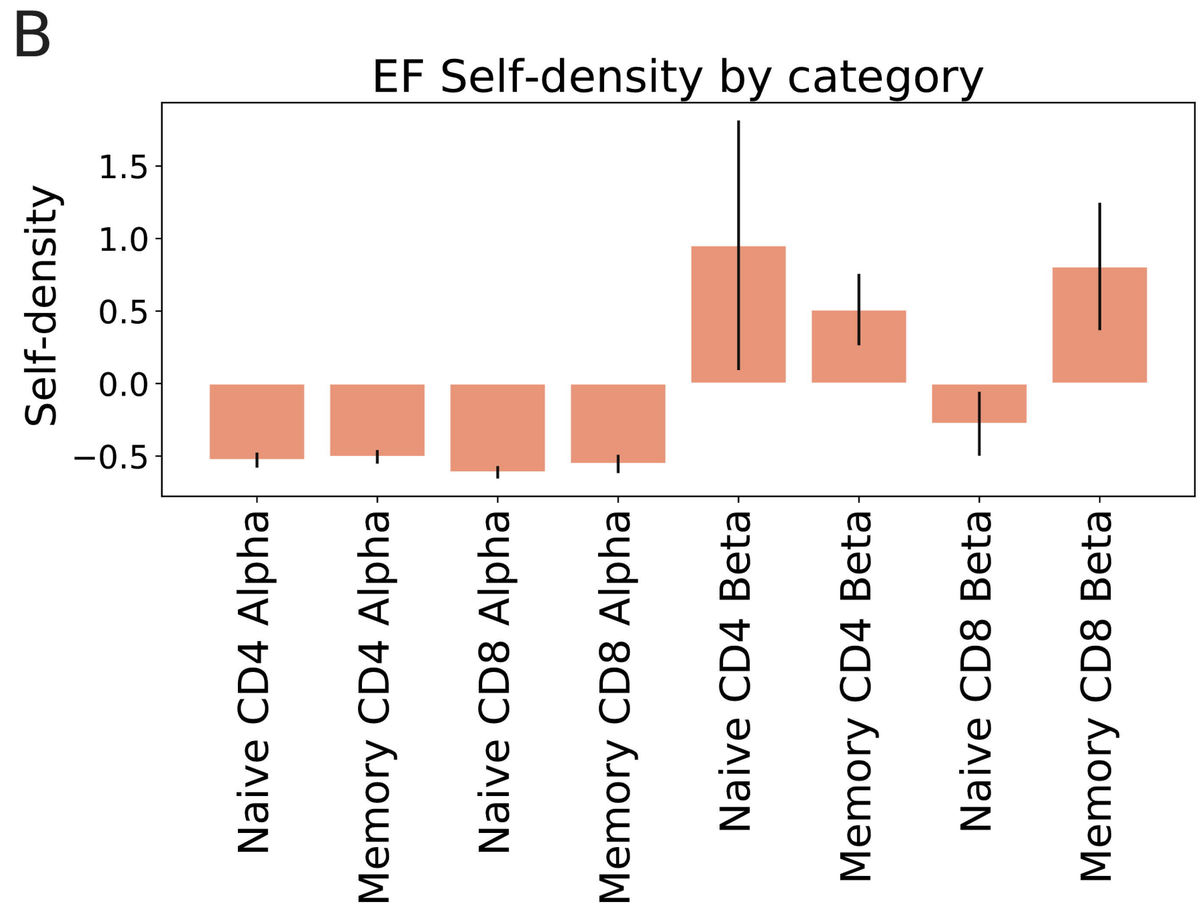
H

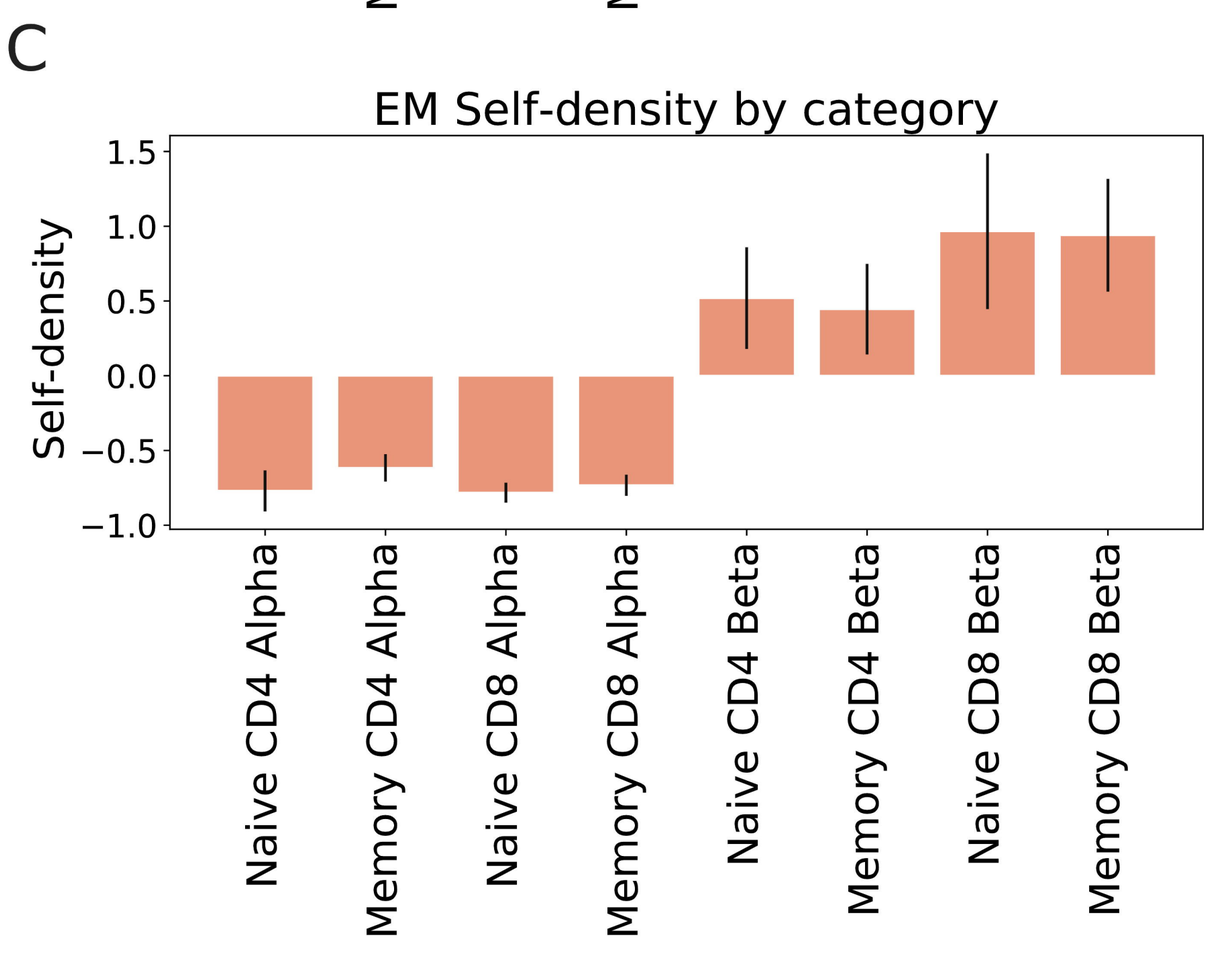


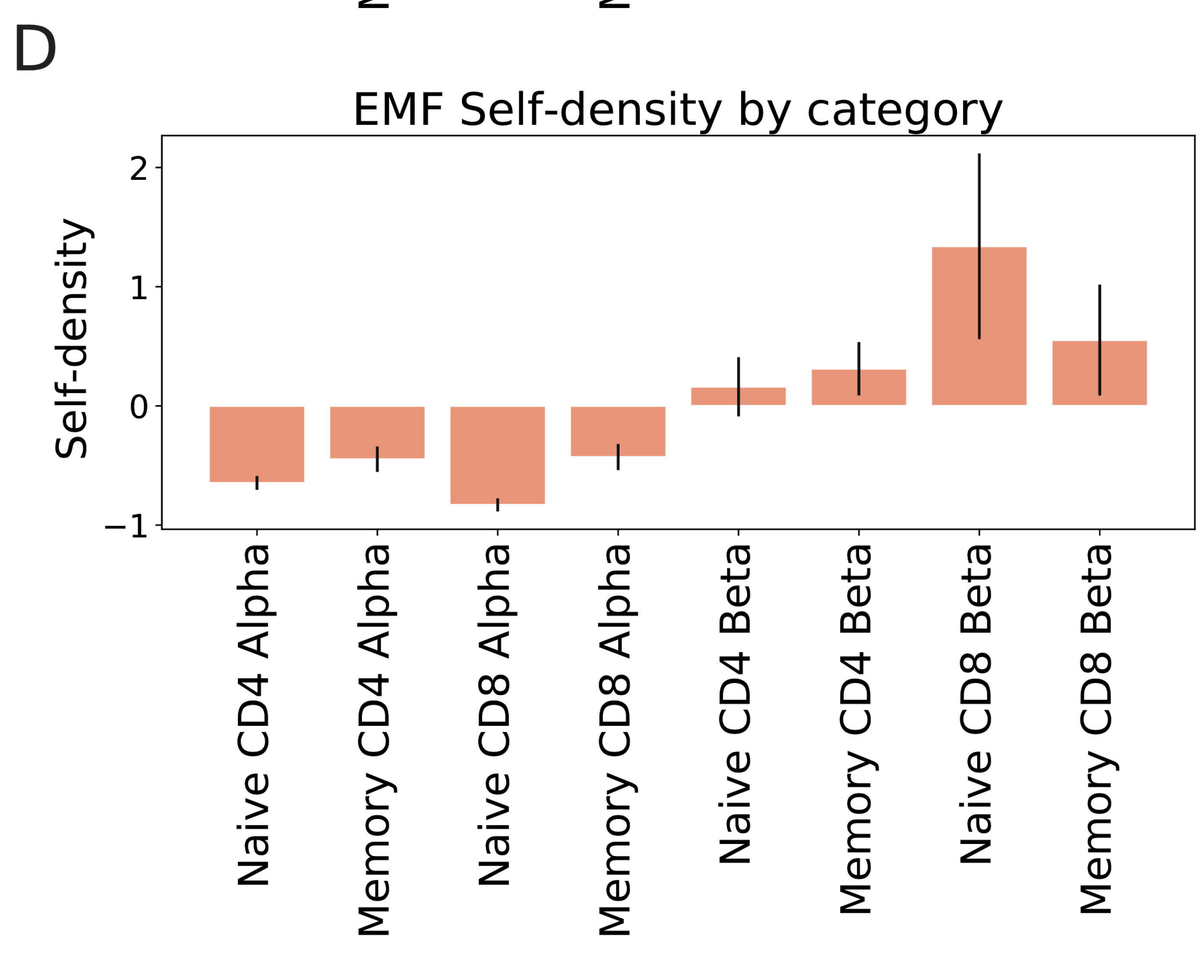

A
Projections TSNE Cysteine
Cysteine
Other
TSNE D1
TSNE D2
B
Projections TSNE , Lengths
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
24
26
30
TSNE D1
TSNE D2
C
Distance from center by publicity
R = -0.09199
Distance from center
Publicity
D
Self-Density by publicity
1e−28
R = 0.135734
Self-Density
Publicity
E
Autoencoder Projections TSNE
TPRVTGGGAM
Other
TSNE D1
TSNE D2
F
Autoencoder Projections TSNE
Cancer
Other
TSNE D1
TSNE D2
G
Self-density by feature
1e−28
Average self-density
VTEHDTLLY
TPRVTGGGAM
NLVPMVATV
LPRRSGAAGA
GLCTLVAML
GILGFVFTL
CTELKLSDY
Random
0 G
1 G
2 G
3 G
4 G
0 C
1 C
0 P
1 P
2 P


A
VE no freqs KDE
P1
P2
S1
S2
Q1
Q2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e14
B
KL
P1
P2
S1
S2
Q1
Q2
0.0
0.2
0.4
0.6
0.8
C
Clusters entropies by hosts
Average entropy
Clusters
E
EF
VE
VEF
EM
EMF
VEM
VEMF
KL
ED
D
Clusters entropies by time points
Average entropy
Clusters
E
EF
VE
VEF
EM
EMF
VEM
VEMF
KL
ED


A
E Self-density by category
Self-density
1.5
1.0
0.5
0.0
−0.5
Naive CD4 Alpha
Memory CD4 Alpha
Naive CD8 Alpha
Memory CD8 Alpha
Naive CD4 Beta
Memory CD4 Beta
Naive CD8 Beta
Memory CD8 Beta


B
EF Self-density by category
Self-density
1.5
1.0
0.5
0.0
−0.5
Naive CD4 Alpha
Memory CD4 Alpha
Naive CD8 Alpha
Memory CD8 Alpha
Naive CD4 Beta
Memory CD4 Beta
Naive CD8 Beta
Memory CD8 Beta


C
EM Self-density by category
Self-density
1.5
1.0
0.5
0.0
−0.5
−1.0
Naive CD4 Alpha
Memory CD4 Alpha
Naive CD8 Alpha
Memory CD8 Alpha
Naive CD4 Beta
Memory CD4 Beta
Naive CD8 Beta
Memory CD8 Beta


D
EMF Self-density by category
Self-density
2
1
0
−1
Naive CD4 Alpha
Memory CD4 Alpha
Naive CD8 Alpha
Memory CD8 Alpha
Naive CD4 Beta
Memory CD4 Beta
Naive CD8 Beta
Memory CD8 Beta