Prediction of specific TCR-peptide binding from large dictionaries of TCR-peptide pairs

Ido Springer 1, Hanan Besser 1, Nili Tickotsky-Moskovitz 2, Shirit Dvorkin 1, Yoram Louzoun 1,*

1) Department of Mathematics, Bar Ilan University, Ramat Gan, Israel, 5290000.

2) The Goodman faculty of life sciences, Bar Ilan University, Ramat Gan, Israel, 5290000.

* To whom correspondence should be addressed: email louzouy@math.biu.ac.il

One Sentence Summary: The combination of advanced tools from natural language processing and large-scale dictionaries of T cell receptors and their target peptide precisely predicts whether a T cell would bind a specific target.

Abstract

The T cell repertoire is composed of T cell receptors (TCR) selected by their cognate MHC-peptides and naive TCR that do not bind known peptides. We here show that the task of distinguishing a peptide binding TCR from a naive TCR can be performed with an accuracy of more than 99% and is mainly the result of peptide specific sequence motifs in the TCR. However, distinguishing between TCRs binding different peptides is more complex. Such a prediction is the key for using TCR repertoires as disease-specific biomarkers.

We here used large scale TCR-peptide dictionaries with state-of-the-art natural language processing (NLP) methods to produce ERGO (pEptide tcR matchinG predictiOn), a highly specific classifier to predict which TCR binds to which peptide. We successfully employed ERGO for multiple related tasks: discrimination between specific peptide or antigen binding TCRs and naïve or memory TCRs, the more complicated task of distinguishing between TCRs that bind different peptides, and finally detecting TCRs binding to neo-antigen. We show that ERGO significantly outperforms all current methods for classification of TCRs that bind peptides, but more importantly can distinguish the specific target of a TCR among a large set of peptides.

The software implementation and data sets are available at: https://github.com/IdoSpringer/ERGO

Introduction

T lymphocytes (T cells) are pivotal in the cellular immune response [1,2]. The immense diversity of the T-cell receptor (TCR) enables specific antigen recognition (TCRs) [3,4]. Successful recognition of antigenic peptides bound to major histocompatibility complexes (pMHCs) requires specific binding of the TCR to these complexes [5–7], which in turn directly modulates the cell's fitness, clonal expansion and acquisition of effector properties [7]. The affinity of a TCR for a given peptide epitope and the specificity of the binding are governed by the heterodimeric αβ T-cell receptors [2]. While both chains have been reported to be important to affect binding, we show here that the TCR's binding to target MHC-peptide can be determined in most cases using the β-chain only (See also [8]).

Within the TCRβ chain, the complementarity-determining region 1 (CDR1) and CDR2 loops of the TCR contact the MHC alpha-helices while the hypervariable CDR3 regions interact mainly with the peptide [1,2]. In both TCRα and TCRβ chains, CDR3 loops have by far the highest sequence diversity and are the principal determinants of receptor binding specificity.

Following specific binding of T cell receptors to viral and bacterial-derived peptides bound to MHC [5], or from neo-antigens [8–10], the appropriate T cells expand, resulting in the increased frequency of T cells carrying such receptors. Recently, high-throughput DNA sequencing has enabled large-scale characterization of TCR sequences, producing detailed T cell repertoire (Rep-Seq) [11]. Expanded clones are more likely to be repeatedly sampled in Rep-Seq than non-expanded clones and may be thus used as biomarkers for the presence of their cognate target, especially if the sample is enriched for mature or activated cells, or if strict filtering on sampled clone size is applied.. Using these repertoires as large-scale biomarkers requires a precise enough distinction between TCRs binding distinct targets (this is often referred to as “reading the repertoire”). A classifier producing such a distinction is currently not available.

A direct approach for using TCR Rep-Seq as biomarkers has been proposed by Emerson et al. [12], who detected patients that have CMV based on their full repertoire and the choice of TCRs that differ between CMV positive and negative patients. This approach is based on the presence of public TCR that are highly specific and repetitively observed in the response of different hosts to the same peptide (often denoted public clones [13]). Moreover, such an approach requires extensive repertoire sequencing for every condition tested.

In contrast, many TCR responses are characterized by a high level of cross-reactivity with single TCRs binding large number of MHC-bound peptides, and single peptides binding a large number of TCRs [14,15]. TCRs binding the same MHC-peptide may share similarities but possess different CDR3 sequences. Thus, while for public clones the task of deciphering the relation between a peptide and the TCR binding it is based on tallying the candidate public TCR, for most TCRs which are highly cross-reactive, a probabilistic approach is required.

Important steps have been made in this direction by Glanville et al. [4] and Dash et al. [16], who detected the clear signature of short amino acid motifs in the CDR3 region of TCRβ and TCRα in response to specific peptides presented by specific MHC molecules. This work was then extended by two more recent efforts that combined such motifs with machine learning approaches

[17,18] to predict peptide specific TCRs vs. naïve TCRs, using either Convolutional Neural Networks [18], or Gaussian Processes [17]. These methods significantly outperform random classification in practically all tested datasets. These last methods did not incorporate the HLA type in the prediction.

The next required step for using the repertoire to develop specific biomarkers would be to distinguish between TCR binding different peptides. We here propose that TCRs binding any peptide can be easily separated from naive TCRs unlikely to bind any peptide (with an Area Under Curve of above 0.98). However, separating TCRs binding different peptides is a much more complex task.

Such tasks are now addressed in Natural Language Processing (NLP) using recurrent neural networks (RNN). We propose to adapt the method to the TCR-peptide classification task with RNN, artificial neural networks models that can handle variable length inputs, such as sequences and words. Recurrent connections are involved in computing a hidden cell state, which can be considered the network memory. Long short-term memory (LSTM) networks are a common RNN type that uses continuous gating functions for updating the cell state [19]. LSTM are often preferred, due to their sophisticated architecture that avoids the vanishing gradient problem. We extended such techniques to TCR-peptide binding and created ERGO (pEptide tcR matchinG predictiOn). We show that ERGO significantly outperforms all existing methods in multiple classification tasks, including: the detection of TCRs binding specific peptides in a full repertoire, the distinction between TCR binding different peptides, and the division of a full repertoire into TCR predicted to bind peptides vs TCRs not predicted any peptide in the full database.

Results

ERGO outline

We address the TCR-peptide binding prediction issue using a dual LSTM formalism, where the TCR is fed to one LSTM and the peptide is fed to a different LSTM, and the output of these two LSTM is then used by a feed-forward neural network (FFN) to produce a binary output for the binding of a specific TCR to a specific peptide (Figure 1B). The details of the machine learning are provided in the Methods section.

Since target peptides and TCRs have different generation mechanisms (TCRs through VDJ recombination and junctional diversity [11], and peptides through antigen generation, trafficking, processing and MHC binding [20]), we expect their amino acid (AA) compositions to differ. Thus, different parallel encoders are used. At the broad level, we encode the CDR3 of each TCR and each peptide into numerical vectors. The encoded CDR3 and the encoded peptide are concatenated and used as an input to a network which should output 1 if the TCR and peptide bind and 0 otherwise (Figures 1A & 1B). At this stage, the MHC was not included, since it did not contribute significantly to the prediction accuracy.

For the peptides, we first use an initial random embedding to translate each amino acid (AA) in the CDR3 to a 10-dimensional AA Embedding Vector. Note that the encoding dimension

is of limited effect. In order to merge these encoding vectors into a single vector representing the peptide, each vector is used as input to a LSTM network. The LSTM network produces an output vector for every prefix of the peptide sequence. We use the last vector as the encoding of the sequence. The embedding matrix values, the weights of the LSTM and the weights of the FFN are trained simultaneously. For the TCR encoding, we either use a similar approach or an autoencoder (AE) (See methods and Figure 1A & 1B) with similar accuracies.

These models were trained on two large datasets of published TCR binding specific peptides [21,22]. McPAS-TCR [21] is a manually curated database of TCR sequences associated with various pathologies and antigens based on published literature. The database contains more than 11,000 TCRβ sequences matching to over 300 unique epitope peptides. These TCRs are associated with various pathologic conditions (including pathogen infections, cancer and autoimmunity) and their respective antigens in humans and in mice. Another large-scale database is VDJdb [22], an open, comprehensive database of TCR sequences and their cognate epitope and the restricting MHC allotype acquired by manual processing of published studies. For each TCR-peptide pair, a record confidence score is computed. The database (available at https://vdjdb.cdr3.net/) contains more than 20,000 records, and about 200 unique epitopes of various antigen origins.

ERGO can predict TCR binding to specific epitopes or antigen.

To test the performance of ERGO, we calculated the model AUC for specific frequent peptides in the test set (In McPAS-TCR database), we analyzed the 5 most frequent peptides in the dataset, and tested the possibility of detecting whether a randomly selected peripheral T cell clone would bind the peptide. The Area Under Curve (AUC) for the binary classifications ranged from 0.946 for the worse peptides to above 0.99 for the best peptides. The AE classifier typically outperformed the LSTM (Table 1). Note that the results are not very sensitive to the number of TCRs reported for the peptides, and all peptides with more than 20 reported TCRs had similar values (Supplementary Materials Table S1).

A more important task from a diagnostic point of view would be to distinguish between TCRs binding different peptides. A diagnostic tool should not only detect a response but be specific about the precise target of the response (specific peptide/antigen or disease). To test the specificity of the prediction (i.e. not only detecting that the TCR binds something, but whether it binds this specific peptide or another peptide), we replaced the baseline of the comparison to be TCR binding other peptides in the McPAS dataset. This task is much more complex, but still reaches AUC values of 0.795-0.984). Finally, we compared the accuracy vs memory clones from the Chain et al analysis [23]. This is the most challenging tasks with AUCs of 0.635-0.984.

Given the high peptide binding accuracy, we tested the accuracy of ERGO in the antigen predicting task, where the goal is to predict whether a TCR binds a specific antigen (i.e. any peptide within this antigen. The accuracy in this task is even higher than in the peptide epitope binding

predictions, with test AUCs reaching 0.999 for the best antigen (PB1 in influenza) (Table 2). Again, the analysis was performed on the 5 most frequent antigen in the Mc-PAS dataset. (More results can be found in Supplementary Materials Table S2). When all antigens in both datasets were tested, an average AUC of above 0.97 is obtained in the McPAS and above 0.99 in the VdJdb (Table 3).

Comparison to existing models

In order to show that ERGO significantly outperforms all current approaches, we tested its precision on previous attempts to predict TCR-peptide binding. We first compared it to the work of Jokinen et al. [17] who compared TCRs found by Dash et al. [16] to bind three human epitopes and seven mice epitopes with TCRs from VDJdb database [22], which bind additional 22 epitopes. These peptide-TCR pairs were compared with naïve TCRs not expected to recognize the epitopes. They evaluated their model performance by comparing TCRs binding each peptide separately with naïve TCRs (Table 4).

To compare our models to Jokinen et al.'s model, TCRGP, we used all TCR-peptide pairs in our datasets as the positive examples while the negative examples were defined as random peptides from the positive set and a random background of TCRs gathered from Dash et al. [16]. We tested the ERGO classifier on these datasets and found it classified peptide binding TCRs vs. naive TCRs and it significantly outperformed all existing methods, with an AUC of over 0.98 (Table 4, Figure 1C).

We suggest that the classification tasks reach such a high precision since the properties of peptide binding CDR3s differ from those of naïve TCRs in multiple respects, including length distribution (as defined by the number of amino acids between the conserved cysteine at the 5' end of the V segment and the conserved phenylalanine at the 3' end of the J segment (Figure 1D), the transition probability between different amino acids (defined as the probability of observing a given amino acid following another amino acid - see Methods) - (Figure 1E), or even biochemical properties for the total CDR3, as apparent by the Kidera factors [24] (Supp. Mat. Figure S1). These factors represent orthogonal properties of sequences to avoid the high correlations between different sequence biochemical properties. The first 3 amino acids and the last amino acid (typically 'CAS' and 'F' respectively) were removed from the analysis. To highlight the large difference between naïve TCRs and all peptide binding TCRs studied here, the sequence of TCRs in the McPAS-TCR dataset [21] (available at http://friedmanlab.weizmann.ac.il/McPAS-TCR/) and the background TCRs in Dash et al. [16] data was compared using a two sample logo [25] (at http://www.twosamplelogo.org/cgi-bin/tsl/tsl.cgi) (Figure 1F). For the sake of representation, we limited the analysis to 10 mers (Figure 1F), but results are similar for other lengths (Supplementary materials Figure S2). Only amino acids whose distributions differ significantly between the two databases are shown. A motif of tripe Glycines is apparent in CDR3 positions 1-3 in the McPAS-TCR data (Figure 1F), which clearly separates those from naïve peptides.

Train and test for specific TCR -peptide binding

Again, we applied the ERGO classifier to predict for TCR-peptide pairs whether the TCR binds the peptide. In contrast with the results in Figure 1 and with previous methods, our control was the same TCR and peptides as the true binders but randomly paired. We tested both autoencoder and LSTM approaches and obtained accuracies of 0.85 for the McPAS-TCR dataset and 0.8 for the VDJdb dataset (Figure 1C and Tables 3,4). When subsampling the existing datasets, the accuracy increases with sample size and does not seem to saturate at current sample size (Figure 2A, McPAS TCR) Some peptides have many reported TCRs binding them, while some have a single reported binding TCR. (Figure 2D) We tested whether a larger number of reported binding TCR improves the accuracy (Figure 2F). Up to a 100 TCRs, a higher number of bound TCRs induce higher prediction accuracy. However, for the few most promiscuous peptides, the accuracy decreases, suggesting that larger datasets would further improve ERGO's accuracy Note that ERGO does not use the HLA information. Including the HLA of the peptide had practically no impact on the accuracy (data not shown).

CDR3 sequence characteristics

To test which position along the CDR3 has the strongest effect on the binding prediction, we trained ERGO ignoring one TCR amino-acid position at a time, by nullifying the position in the autoencoder based model or by skipping that position input in the LSTM based model. We tested ERGO's performance for every position separately. Figure 2C shows that missing central (positions 7 to 15) amino-acids of the TCRs impair the model's performance, especially in the LSTM-based model. The autoencoder-based model seems to be more stable than the LSTM based model, perhaps due to exposure to a variety of TCRs in the TCR autoencoder pre-training.

We used again two-sample-logos to compare the CDR3 sequences of cognate TCRs for two peptides that have the highest number of cognate TCRs in the McPAS TCR database (Figure 2E). Both peptides, 'LPRRSGAAGA' and 'GILGFVFTL' are Influenza A virus epitopes. Interestingly, only the 2nd position in the CDR3 (10 mers) differed significantly between the two TCR groups, although the peptides are derived from different viral proteins, nucleoprotein (NP) and matrix protein (M1). Both are linear peptides, LPRRSGAAGA has one aromatic ring and GILGFVFTL has two. Both M1 and NP have highly conserved sequences, and a combined vaccine against both is remarkably immunogenic, leading to frequencies of responding T cells that appear to be much higher than those induced by any other influenza vaccination approaches [26,27].

Prediction of TCR-neoantigen binding

ERGO would be essential for the development of future TCR-based diagnostic tools. However, it can already be used for detection of TCRs that bind specific tumor antigens. Given a neoantigen extracted from full genome sequencing of tumors [28,29] and a target TCR, one could estimate the binding probability of the TCR to such a neoantigen. To test for that, we applied ERGO to neoantigen binding prediction;

We used a positive dataset of cancer neoantigen peptides and their matching TCRs, published by Zhang et al. [30], and expended it with TCR-matching neoantigens in the McPAS TCR and VDJdb databases. Again, we tested two setups: The basic setup in which neoantigens binding TCRs are compared with naive TCRs, showed an AUC of 0.97 (Table 4). The more complex setup that distinguishes between different TCRs binding different neo-antigens performed with an AUC of 0.69, (Figure 2B and Table 4). Again, the performance of the autoencoder-based classifier was better than the LSTM one. Note that the analysis was performed on a comparison of a datasets of TCRs binding neo-antigens and T cells from repertoires of healthy donors. Thus, this is not a direct measurement of the possibility of detecting neo-antigen specific TCRs within a donor.

ERGO's performance on TCRs of unknown specificities

Finally, we compared ERGO's performance in a mixed setting of TCRs of unknown-specificities recently proposed by Jurtz et al. [18], who used a convolutional neural network (CNN) based model, NetTCR, for predicting binding probability of TCR- HLA-A*02:01 restricted peptide pairs. Jurtz et al. experimented on two datasets, one downloaded from IEDB was used to train the model and an additional dataset, generated using the MIRA assay provided by Klinger et al. [31], was used for evaluating the model, by testing the model performance on shared IEDB and MIRA peptides (Table 5 and Table 6). Jurtz et al. used two models in their experiments. One was trained with positive IEDB examples and only negative examples made from the IEDB dataset itself (no additional sources) while another model had also additional naïve negatives [32]. We used the united IEDB and MIRA positive examples dataset provided by Jurtz et al. and created also negative examples from that dataset. We trained ERGO models with 80% of the united data (positive and negative examples), and evaluated the model performance on the rest of the data (20%) (Table 5). Again, ERGO significantly outperformed the current results (Table 5).

Discussion

We have shown that a combination of state-of-the-art NLP methods with curated datasets can lead to high accuracy predictions of TCR-peptide binding. Two different levels of accuracy can be obtained; while distinguishing naive TCR from TCR binding a specific peptide can be performed with an accuracy of over 98%, distinguishing TCRs binding different peptides can be performed with an accuracy of around 85% in the current setting.

The classification tasks reach such a high precision since the properties of peptide binding CDR3 differ from those of naïve TCRs by multiple aspects, including length distribution, probability of transition between different amino acids and biochemical properties of the CDR3 sequence, as demonstrated by the Kidera factors. This difference shows that the T cells selected through cognate interactions with target peptides have a very clear signature. Indeed when peptide binding TCRs were compared with memory TCRs, the classification accuracy was similar and even lower than the one of TCRs binding different peptides.

Many elements can affect the obtained precision. Two such elements tested here include the size of the dataset used for training and its curation level. The curated McPAS TCR has a higher AUC for the distinction between different peptide binding TCRs than the open VDJdb dataset. However, the opposite happens for the distinction of specific binding peptides the VDJdb produces a better accuracy(Tables 3 and 4). This suggests that curating the dataset is important for the precision of the target, but enlarging the database improves the distinction between peptide binding and naïve TCRs. In addition, when subsampling the existing datasets, the accuracy increases with sample size and does not seem to saturate at current sample size (Figure 2A), suggesting that better curation and larger datasets can further improve the algorithm's accuracy.

However, several other elements can affect the results, such as the V and J gene used and the alpha chain. In general, TCR-sequencing have been commonly limited to the TCR β chain due to its greater combinatorial and junctional diversity [8] and to the fact that a single TCRβ chain can be paired with multiple TCRα chains [33]. Pogorelyy et al. [34] have shown concordance between TCRα and TCRβ chain frequencies specific for a given epitope and suggested this justifies the exclusive use of TCRβ sequences in analyzing the antigen-specific landscape of heterodimeric TCRs. Currently, most proposed classifiers have used only CDR3 beta chains [4,35] but some attempts have been made to include alpha chains [16]. Only recently, with single-cell techniques that enable pairing of α and β chains sequences, more data on alpha-beta TCRs is accumulating [36]. Once large-scale curated alpha-beta TCR-peptide datasets are available, their integration into the current method is straight forward.

ERGO is based on LSTM networks to encode sequential data. Previous models by Jurtz et al. [18] used convolutional neural networks (CNN) for the similar task. While CNN are good at extracting position-invariant features, RNN (in particular LSTM) can catch a global representation of a sequence, in various NLP tasks [37].

ERGO randomly initializes our amino-acid embeddings and trains the embeddings with the model parameters. Using word-embedding algorithms such as Word2Vec [38] or GloVe [39] can give a good starting point to the embeddings. Special options for amino-acids pre-trained embeddings include the use of BLOSUM matrix [40] or Kidera-factors-based manipulations. Pre-trained embedding usually provides better model results. We plan to further test such encodings.

The prediction method presented here can serve as a first step in identifying neoantigen-reactive T cells for adoptive cell transfer (ACT) of tumor-infiltrating lymphocytes (TILs) targeting neoantigens [41]. The ERGO algorithm can accelerate the preliminary selection of valid target epitopes and corresponding TCRs for adoptive cell transfer. Finally, an important future implication would be to predict TCR-MHC binding, such prediction can be crucial for improving mismatched bone marrow transplants [42].

Materials and Methods

The Data

Three TCR-peptide datasets were used in the attachment prediction task. McPAS-TCR dataset was downloaded from http://friedmanlab.weizmann.ac.il/McPAS-TCR/ and VDJdb dataset was downloaded from https://vdjdb.cdr3.net/. We used a dataset of cancer neoantigen peptides and their matching TCRs, published by Zhang et al. [30]. A set of cancerous peptides was made for extracting TCRs matching to these peptides also in McPAS-TCR and VDJdb databases. We extended the original cancer dataset to include all TCRs-cancerous peptide pairs in all datasets. The data were processed into TCR-peptide pair files, using only TCRβ chains and valid TCR and peptide sequences.

The TCR autoencoder was trained on a data which was derived from a prospective clinical study (NCT00809276) by Kanakry et al. [43] The dataset is freely available at the Adaptive database (www.adaptivebiotech.com) that provides open access to a variety of datasets of TCRs next generation sequencing.

Datasets studied

In each model, training data was loaded as batches of positive and negative examples. For the positive examples, we took the existing TCR-peptide pairs in the training set. For the negative examples for binding of a specific TCR to a specific peptide, we first chose a peptide randomly from the peptides in the training set. Then we chose five random TCRs from the training set that are not reported to bind this peptide to create five internal wrong pairs. We then removed duplicate pairs from the training and validation sets.

For the negative example for distinguishing a peptide-binding TCR from naive TCRs, we chose a random peptide in similar way and five random background TCRs from the Dash et al. dataset. We again removed duplicates and used equal numbers of negatives and positives.

Correlation map

For computing a correlation map of a given dataset, we index all amino acids, including 'start' and 'end' symbols. Then we counted the presence of every two consecutive amino acids for every TCR, when the first amino acid follows 'start' and 'end' follows the last amino acid. We set a matrix with the relevant counts and normalized each row to sum up to 1. The result is a stochastic matrix P, where $P(i,j)$ is the probability of moving from amino acid with index i to the amino acid with index j. Notice that $P(\text{'}end\text{'},k) = 0$ for every k, so the last row does not sum up to 1.

The Models

We used two models for predicting TCR-peptide binding. The models use deep-learning architectures to encode the TCR and the peptide. Then the encodings are fed into multi linear perceptron (MLP) to predict the binding probability. Two encoding methods are applied – LSTM acceptor encoding and Autoencoder-based encoding. The peptide is always encoded using the LSTM acceptor method, so the two models differ in the TCR encoding method.

LSTM Acceptor

First the amino acids were embedded using an embedding matrix. We set each amino acid an embedding vector, randomly initialized. Next, the TCR or the peptide was fed into a LSTM network as a sequence of vectors. The LSTM network outputs a vector for every prefix of the sequence; we used the last output as the encoding of the whole sequence. We used two different embedding matrices and LSTM parameters for the TCRs and the peptides encodings. The embedding dimension of the amino acids was 10. We use two layered stacked LSTM, with 30 units at each layer. A dropout rate of 0.1 was set between the layers.

TCR Autoencoder

The TCR autoencoder was trained before training the Autoencoder-based attachment prediction model. In order to train the TCR autoencoder, first we added a 'stop-codon' at the end of every TCR CDR3 sequence. Each amino acid was represented as one-hot vector of 21 numbers (20 possible amino acids and an additional stop codon) where all values were set to zeros except one index of the corresponding amino acid which was set to 1. Each of the CDR3 vector representations one-hot vectors were joined and, terminated with a 'stop codon' one-hot vector. Zero padding was then added to the CDR3 vectors, completing the vectors to the maximum lengths chosen according to the data lengths distribution. Each zero codon was represented as fully zeroed one-hot vector.

The concatenated TCR vectors were fed into the Autoencoder network, which was based on a combination of linear layers, creating a similar 'encoder' and 'decoder' networks. In the encoder the TCRs were first put into a layer with 300 units, then into a layer with 100 units, and then into the encoding layer with 30 units. This layer output was used to encode the TCR in the trained autoencoder model. We used Exponential Linear Unit (ELU) activation between the linear layers and dropout with rate of 0.1. The decoding layers were similar to the encoding layers in the

reverse order – first the encoded TCR vectors were fed into a layer with 100 units, then into a layer with 300 units, and then into a layer with the original TCR concatenated one-hot vector length units. We used softmax on the last decoder layer output on every sub-vector matching to an input amino acid one-hot vector position.

We used Mean Squared Error (MSE) loss (when the decoder output should be like the concatenated one-hot input). The autoencoder was trained using Adam optimizer with learning rate of 1e-4, We used batching with batch size 50. The autoencoder was trained for 300 epochs.

In order to read the TCR from the decoding vector, first we split the long vector into 'one-hot' like vectors. We back-translated the one-hot vectors into amino-acids by taking the amino acid matching to the maximal value index in the vector (which should be 1). We dropped all amino acids from the stop codon and forward to get a sequence of amino acid which should be the TCR.

The autoencoder was trained with 80% of the data and was evaluated with the rest of it.

The accuracy of the autoencoder was tested using the predictions of the test set. The number of mismatches between a CDR3 input vector to its prediction after trimming both vectors according to the start position of the codon vector in the predicted representation, removing the stop vector along with the zeros chain, leaving only the CDR3 representation to be accurate. Three types of accuracies were considered in this project, considering not only exact match as true prediction, but also allowing for one or two mismatches (Supplementary materials Table S3).

MLP Classifier

In both models, the TCR encoding was concatenated to the peptide encoding and fed into the MLP. The MLP contains one hidden layer with 30 units (half of the concatenated vector) and sigmoid is used on the output of the last layer to get a probability value. In both models the activation in the MLP is Leaky ReLU. Dropout with rate of 0.1 was set between layers.

Model configurations

As mentioned, we used two models, the LSTM based model and the autoencoder based model. We trained the embeddings, the LSTM parameters and the MLP in the first model, and the TCR autoencoder, peptide LSTM encoder and MLP parameters in the second model. The trained TCR autoencoder parameters were loaded to the autoencoder based model and are trained again within all model parameters.

We used Binary Cross Entropy (BCE) loss. The optimizer was Adam with learning rate of 1e-3 and weight decay 1e-5. We used batching with batch size 50. The model was trained for 200 epochs. The models used 80% of the data for training and 20% for evaluation for all datasets.
All models were implemented with PyTorch library in Python programming language.
The prediction models were evaluated using Area Under the Curve (AUC) score.

Supplementary Materials

Figure S1. Average Kidera factors distribution In McPAS and in the naïve negative TCRGP TCRs.

Figure S2. McPAS-TCR vs. Dash et al. [16] naïve negatives CDR3β sequences comparative logos in different lengths.

Figure S3. McPAS-TCR two Influenza A peptides matching CDR3β sequences comparative logos in different lengths.

Table S1. AUC of RCR binding to peptides**.**

Table S2. AUC of RCR binding to antigens**.**

Table S3. TCR autoencoder accuracy per number of mismatches allowed in sequence decoding.

References

1. Davis, M. M. & Bjorkman, P. J. T-cell antigen receptor genes and T-cell recognition. *Nature* **334**, 395–402 (1988).
2. Krogsgaard, M. & Davis, M. M. How T cells 'see' antigen. *Nat. Immunol.* **6**, 239–245 (2005).
3. Rowen, L. *et al.* The Complete 685-Kilobase DNA Sequence of the Human beta T Cell Receptor Locus. *Science (80-.).* **272**, 1755–1762 (1999).
4. Glanville, J. *et al.* Identifying specificity groups in the T cell receptor repertoire. *Nat. Publ. Gr.* **547**, (2017).
5. Rudolph, M. G., Stanfield, R. L. & Wilson, I. A. HOW TCRS BIND MHCS, PEPTIDES, AND CORECEPTORS. *Annu. Rev. Immunol.* **24**, 419–466 (2006).
6. Rossjohn, J. *et al.* T Cell Antigen Receptor Recognition of Antigen-Presenting Molecules. *Annu. Rev. Immunol.* **33**, 169–200 (2015).
7. Zhang, S.-Q. *et al.* Direct measurement of T cell receptor affinity and sequence from naïve antiviral T cells. *Sci. Transl. Med.* **8**, 341ra77 (2016).
8. Schrama, D., Ritter, C. & Becker, J. C. T cell receptor repertoire usage in cancer as a surrogate marker for immune responses. *Semin. Immunopathol.* **39**, 255–268 (2017).
9. Cohen, C. J. *et al.* Isolation of neoantigen-specific T cells from tumor and peripheral lymphocytes. *J. Clin. Invest.* **125**, 3981–3991 (2015).
10. Page, D. B. *et al.* Deep Sequencing of T-cell Receptor DNA as a Biomarker of Clonally Expanded TILs in Breast Cancer after Immunotherapy. *Cancer Immunol. Res.* **4**, 835–844 (2016).
11. Benichou, J., Ben-Hamo, R., Louzoun, Y. & Efroni, S. Rep-Seq: uncovering the immunological repertoire through next-generation sequencing. *Immunology* **135**, 183–91 (2012).
12. Emerson, R. O. *et al.* Immunosequencing identifies signatures of cytomegalovirus exposure history and HLA-mediated effects on the T cell repertoire. *Nat. Genet.* **49**, 659–665 (2017).
13. Madi, A. *et al.* T-cell receptor repertoires share a restricted set of public and abundant CDR3 sequences that are associated with self-related immunity. *Genome Res.* **24**, 1603–12 (2014).
14. Wooldridge, L. *et al.* A Single Autoimmune T Cell Receptor Recognizes More Than a Million Different Peptides. *J. Biol. Chem.* **287**, 1168–1177 (2012).
15. Sewell, A. K. Why must T cells be cross-reactive? *Nat. Rev. Immunol.* **12**, 669–677 (2012).
16. Dash, P. *et al.* Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* **547**, 89–93 (2017).
17. Jokinen, E., Heinonen, M., Huuhtanen, J., Mustjoki, S. & Lähdesmäki, H. TCRGP: Determining

epitope specificity of T cell receptors. doi:10.1101/542332
18. Jurtz, V. I. *et al.* NetTCR: sequence-based prediction of TCR binding to peptide-MHC complexes using convolutional neural networks. *bioRxiv* 433706 (2018). doi:10.1101/433706
19. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9**, 1735–1780 (1997).
20. Louzoun, Y., Vider, T. & Weigert, M. T-cell epitope repertoire as predicted from human and viral genomes. *Mol. Immunol.* **43**, 559–569 (2006).
21. Tickotsky, N., Sagiv, T., Prilusky, J., Shifrut, E. & Friedman, N. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. doi:10.1093/bioinformatics/btx286
22. Shugay, M. *et al.* VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Res.* **46**, D419–D427 (2018).
23. Bosselut, R. *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. *Immunol* **8**, 1267 (2017).
24. Kidera, A., Konishi, Y., Oka, M., Ooi, T. & Scheraga, H. A. *Statistical Analysis of the Physical Properties of the 20 Naturally Occurring Amino Acids*. *Journal of Protein Chemistry* **4**, (1985).
25. Vacic, V., Iakoucheva, L. M. & Radivojac, P. Two Sample Logo: a graphical representation of the differences between two sets of sequence alignments. *Bioinformatics* **22**, 1536–1537 (2006).
26. Zheng, M., Luo, J. & Chen, Z. Development of universal influenza vaccines based on influenza virus M and NP genes. *Infection* **42**, 251–62 (2014).
27. Antrobus, R. D. *et al.* Coadministration of Seasonal Influenza Vaccine and MVA-NP+M1 Simultaneously Achieves Potent Humoral and Cell-Mediated Responses. *Mol. Ther.* **22**, 233–238 (2014).
28. Jia, J. *et al.* Genome-scale search of tumor-specific antigens by collective analysis of mutations, expressions and T-cell recognition. *Mol. Immunol.* **46**, 1824–9 (2009).
29. Laumont, C. M. *et al.* Noncoding regions are the main source of targetable tumor-specific antigens. *Sci. Transl. Med.* **10**, eaau5516 (2018).
30. Zhang, S.-Q. *et al.* High-throughput determination of the antigen specificities of T cell receptors in single cells. *Nat. Biotechnol.* **36**, 1156–1159 (2018).
31. Klinger, M. *et al.* Multiplex Identification of Antigen-Specific T Cell Receptors Using a Combination of Immune Assays and Immune Receptor Sequencing. *PLoS One* **10**, e0141561 (2015).
32. Savola, P. *et al.* Somatic mutations in clonally expanded cytotoxic T lymphocytes in patients with newly diagnosed rheumatoid arthritis. *Nat. Commun.* **8**, 15869 (2017).
33. Birnbaum, M. E., Dong, S. & Garcia, K. C. Diversity-oriented approaches for interrogating T-cell receptor repertoire, ligand recognition, and function. *Immunol. Rev.* **250**, 82–101 (2012).
34. Pogorelyy, M. V *et al.* Exploring the pre-immune landscape of antigen-specific T cells. *Genome Med.* **10**, 68 (2018).
35. De Neuter, N. *et al.* On the feasibility of mining CD8+ T cell receptor patterns underlying immunogenic peptide recognition. *Immunogenetics* **70**, 159–168 (2018).
36. De Simone, M., Rossetti, G. & Pagani, M. Single Cell T Cell Receptor Sequencing: Techniques and Future Challenges. *Front. Immunol.* **9**, 1638 (2018).
37. Yin, W., Kann, K., Yu, M. & Schütze, H. Comparative Study of CNN and RNN for Natural Language Processing. (2017).
38. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient Estimation of Word Representations in Vector Space. (2013).
39. Pennington, J., Socher, R. & Manning, C. D. *GloVe: Global Vectors for Word Representation*.
40. Henikoff, S. & Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci.* **89**, 10915–10919 (1992).
41. Hammerl, D., Rieder, D., Martens, J. W. M., Trajanoski, Z. & Debets, R. Adoptive T Cell Therapy: New Avenues Leading to Safe Targets and Powerful Allies. *Trends Immunol.* **39**, 921–

936 (2018).
42. Kollman, C. *et al.* The effect of donor characteristics on survival after unrelated donor transplantation for hematologic malignancy. *Blood* **127**, 260–7 (2016).
43. Kanakry, C. G. *et al.* Origin and evolution of the T cell repertoire after posttransplantation cyclophosphamide. *JCI Insight* **1**, (2016).

Acknowledgments

We wish to thank Prof. Luning Prak for helpful critiques and suggestions.

IS developed the formalism and implemented it.

HB designed the initial formalism.

YL supervised the work and wrote the manuscript.

NT developed the libraries and wrote the manuscript.

SD designed the TCR autoencoder formalism.

Figures

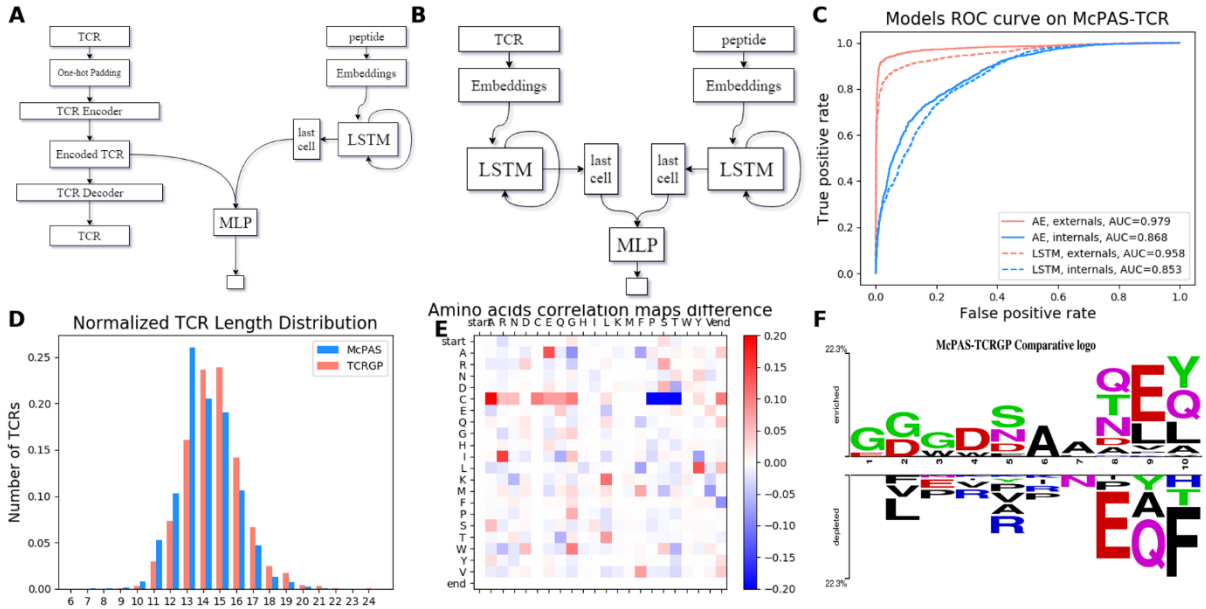

Figure 1: *A flow chart of our model is shown in Figures 1A & 1B: A) Autoencoder based model architecture. B) LSTM based model architecture. Differences between TCRβ chains in McPAS-TCR and Dash et al. background TCRs are shown in Figures 1C-1F: C) ROC curve of LSTM and autoencoder based models performance on McPAS-TCR with naïve (external) and intentional (internal) negatives. D) Normalized length distribution of TCRβ chains in both datasets. E) Difference matrix of correlation maps of the 2 datasets (see Supp. Mat. methods). F) We compared amino acids of CDR3 beta sequences in McPAS and in Dash et al.* [16] *naïve negatives databases (logos were created with Two-Sample-Logos); The height of symbols within the stack in each logo indicates the relative frequency of each amino acid at that position. Only amino acid whose distribution differ significantly between the two databases are shown. A motif of tripe Glycines is apparent only in the McPAS data.*

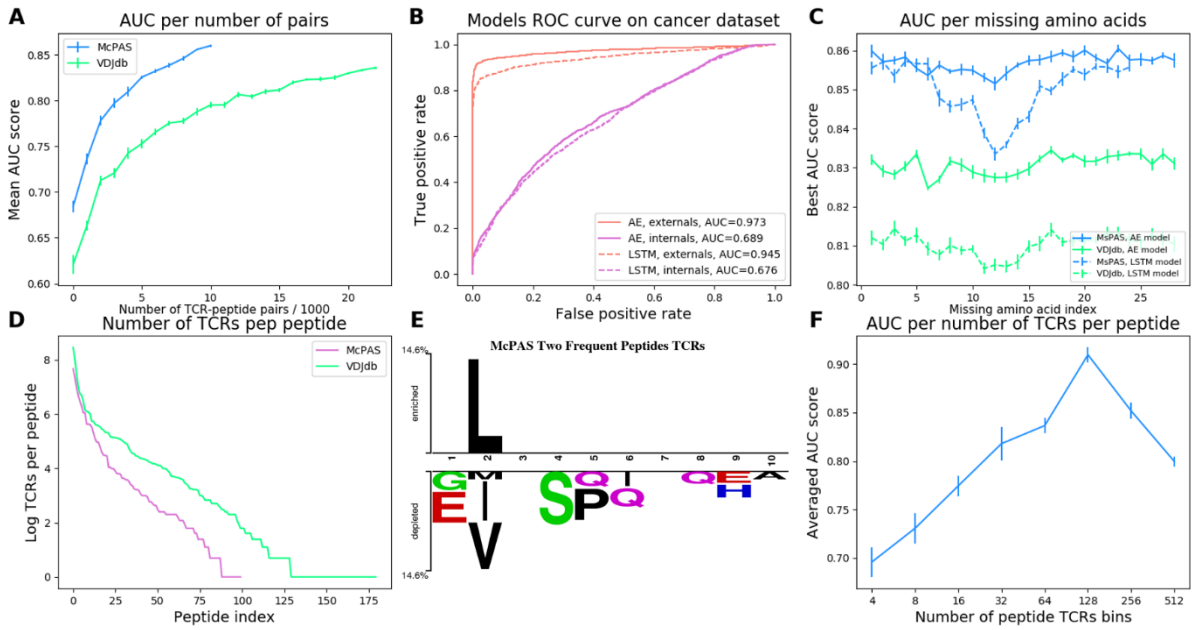

Figure 2: *A) AUC per number of TCR-peptide pairs in McPAS-TCR and VDJdb datasets. B) ROC curve of models performance on cancer dataset with intentional (internal) and naïve (external) negatives. C) AUC per missing amino-acids index. D) Number of TCRs per peptide distribution in McPAS-TCR and VDJdb datasets, logarithmic scale. E) A comparison of the amino acids of the CDR3 beta sequences that target the two most common antigenic peptides in McPAS shows high similarity, although there is no apparent similarity between the peptides. The height of letters within the stack in each logo indicates the relative frequency of each amino acid at that position. F) AUC per number of TCRs per peptide bins (bins are union of all TCRs that match peptides with total number of TCRs in a specific range).*

Peptide	Pathology	Model					
		AE			LSTM		
		Naïve vs binder	Specific binding	Memory vs binder	Naïve vs binder	Specific binding	Memory vs binder
LPRRSGAAGA	Influenza	0.933	0.761	0.610	0.904	0.713	0.568
GILGFVFTL	Influenza	**0.962**	0.820	0.790	**0.956**	0.789	0.760
NLVPMVATV	Cytomegalovirus (CMV)	0.940	0.793	0.751	0.908	0.777	0.799
GLCTLVAML	Epstein Barr virus (EBV)	0.917	0.777	0.734	0.920	0.737	0.683
SSYRRPVGI	Influenza	0.927	**0.932**	**0.965**	0.865	**0.951**	**0.938**

Table 1. *AUC of RCR binding to peptides with 3 possible definitions of non-binders: Naïve cells are non-sorted blood peripheral T cells, specific binders are T cells in the same database binding different peptides, and memory. We used either the autoencoder based (AE) or LSTM based (LSTM) encoders. All values are test AUC values.*

Protein	Pathology	Model					
		AE			LSTM		
		Naïve vs binder	**Specific binding**	**Memory vs binder**	**Naïve vs binder**	**Specific binding**	**Memory vs binder**
NP177	Influenza	0.959	0.785	0.633	0.882	0.742	0.591
Matrix protein (M1)	Influenza	0.977	0.808	0.758	0.910	0.772	0.721
pp65	Cytomegalovirus (CMV)	0.955	0.792	0.793	**0.935**	0.750	0.808
BMLF-1	Epstein Barr virus (EBV)	0.962	0.797	0.817	0.923	0.770	0.772
PB1	Influenza	**0.999**	**0.901**	**0.910**	0.935	**0.904**	**0.899**

Table 2. *AUC of RCR binding to antigens with 2 possible definitions of non-binders. Either Naïve or memory cells as in table 1. Here, the classification is binding to a target protein antigen and not to a specific peptide. We used either the AE based (AE) or LSTM based (LSTM) encoders. All values are test AUC values.*

	McPAS			VDJdb		
Model	**Naïve vs binder**	**Specific binding**	**Memory vs binder**	**Naïve vs binder**	**Specific binding**	**Memory vs binder**
AE	**0.981**	**0.841**	**0.833**	**0.994**	**0.847**	**0.801**
LSTM	0.954	0.834	0.830	0.981	0.822	0.793

Table 3. *AUC of RCR binding to antigens with 2 possible definitions of non-binders. Either Naïve or memory cells as in table 1. Here, the classification is binding to a target protein antigen and not to a specific peptide.. We used either the AE based (AE) or LSTM based (LSTM) encoders. All values are test AUC values.*

	McPAS		VDJdb		Tumor	
Model	**Naïve vs binder**	**Specific binding**	**Naïve vs binder**	**Specific binding**	**Naïve vs binder**	**Specific binding**
VAE	**0.982**	**0.855**	**0.993**	**0.833**	**0.971**	**0.704**
LSTM	0.972	0.850	0.980	0.796	0.935	0.686
Average TCRGP (Dash et al. peptides, CDR3β)	0.828					
Average TCRGP (VDJdb peptides, CDR3β)			0.805			

Table 4: *ERGO Autoencoder-based and LSTM-based models performance on McPAS, VDJdb and tumor datasets with naïve and intentional negatives. TCRGP (CDR3β) column shows Jokinen et al.'s TCRGP* [17] *average model performance.*

Model	Train data	Test data	Test AUC score
ERGO Autoencoder based model	IEDB + MIRA (80%)	IEDB + MIRA (20%)	0.766
ERGO LSTM based model	IEDB + MIRA (80%)	IEDB + MIRA (20%)	0.754
NetTCR INT_NEG	IEDB	MIRA (shared IEDB peptides)	0.697
NetTCR ADD_NEG	IEDB + additional negatives	MIRA (shared IEDB peptides)	0.727

Table 5: *ERGO significantly outperformed the current results of Jurtz et al.* [18]*, who used two models, NetTCR, for their experiments. One was trained with positive IEDB examples and only intentional negative examples made from the IEDB dataset, and another model, the MIRA dataset, had also additional naïve negatives* [32]*.*

Model	Positives	Negatives
NetTCR INT_NEG	IEDB + MIRA	IEDB + MIRA internal wrong combinations
NetTCR ADD_NEG	IEDB + MIRA	IEDB + MIRA internal wrong combinations, with additional negatives from other sources
ERGO	IEDB + MIRA	IEDB + MIRA internal wrong combinations

Table 6: *Jurtz el al. NetTCR* [18] *and ERGO (made for comparison) positive and negative examples creation.*