ImmunoInformatics 11 (2023) 100027

Contents lists available at ScienceDirect

ImmunoInformatics

journal homepage: www.journals.elsevier.com/immunoinformatics

# Interpretable deep learning to uncover the molecular binding patterns determining TCR–epitope interaction predictions

Ceder Dens, Wout Bittremieux, Fabio Affaticati, Kris Laukens, Pieter Meysman *

*Adrem Data Lab, Department of Computer Science, University of Antwerp, Middelheimlaan 1, 2020 Antwerpen, Belgium*

ARTICLE INFO

*Keywords:*
T-cell epitope prediction
Interpretable deep learning
Immunoinformatics

ABSTRACT

The recognition of an epitope by a T-cell receptor (TCR) is crucial for eliminating pathogens and establishing immunological memory. Prediction of the binding of any TCR–epitope pair is still a challenging task, especially for novel epitopes, because the underlying patterns are largely unknown to domain experts and machine learning models. To achieve a deeper understanding of TCR–epitope interactions, we have used interpretable deep learning techniques to gain insights into the performance of TCR–epitope binding machine learning models. We demonstrate how interpretable AI techniques can be linked to the three-dimensional structure of molecules to offer novel insights into the factors that determine TCR affinity on a molecular level. Additionally, our results show the importance of using interpretability techniques to verify the predictions of machine learning models for challenging molecular biology problems where small hard-to-detect problems can accumulate to inaccurate results.

## 1. Background

When a pathogen enters the human body, antigen-presenting cells display short peptides of the pathogen (called epitopes) on their cell surface using a major histocompatibility complex (MHC). A T-cell receptor (TCR) sequence on the surface of a T-cell recognizes the epitope to activate and subsequently initiate the adaptive immune response. Quasi-random genetic rearrangements of the V, D, and J genes that express the TCR sequence make the recognition of a large variety of epitopes possible. The activation of the T-cell is crucial for eliminating the pathogen and creating the immunological memory to prevent severe symptoms with future infections of the same pathogen [1].

Predicting the binding of any given TCR–epitope pair can potentially lead to many advances in healthcare, by aiding diagnostics, vaccine development, and cancer therapies. Although some machine learning tools that perform TCR—epitope binding predictions exist, their performance is still insufficient for many applications. To increase their application potential, a more in-depth analysis of these models is required to gain a better understanding of how they work, which input features they use, and why their performance is not better. Because the CDR3 region of the TCR sequence is in close contact with the epitope [2,3], for simplicity this is often used by prediction tools instead of the full TCR sequence [4–15]. However, it is still unclear which underlying patterns within the CDR3 and epitope sequences lead to binding. Indeed, high-quality machine learning models that are able to learn relevant rules underlying TCR–epitope binding do not currently exist. Broadly, a distinction can be made between the seen-epitope and the unseen-epitope prediction task. For seen-epitope prediction, one attempts to predict if a TCR will bind a known epitope, which means that the training dataset includes TCR–epitope pairs that involve this epitope. This requires a much lower amount of generalization capabilities and pattern learning from the model because it can compare the CDR3 sequence from the test sample to previously seen samples with the same epitope. In contrast, the unseen-epitope prediction task is more difficult because the model is evaluated on samples with novel epitopes that have not been seen during training. This no longer allows matching against known sequences and requires the model to learn more general binding patterns to make a correct prediction. As a consequence, the state-of-the-art performance on the unseen-epitope task is much lower [4,5,12,16] and only marginally better than random. As these models continue to be published and utilized, it is essential to closely examine the feature explainability and predictions of these models to know why they are not doing better, e.g. if these models are even learning to look at the correct features. This is crucial for improving prediction models in the future. In addition, the patterns learned by unseen-epitope models are far more interesting because they model a more realistic setting.

* Corresponding author.
*E-mail address:* pieter.meysman@uantwerpen.be (P. Meysman).

https://doi.org/10.1016/j.immuno.2023.100027
Received 9 February 2023; Received in revised form 6 June 2023; Accepted 6 June 2023
Available online 15 June 2023
2667-1190/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

They also have a higher disruptive potential for healthcare as the possible epitope space far exceeds any feasible database. However, due to the increased complexity of the prediction problem, methods that have been developed to tackle the unseen-epitope task have turned out to be complex machine learning models. All current methods are exclusively based on neural networks, which are notoriously poor at explaining why a prediction is made the way it is, and thus what recognition patterns may underlie an interaction.

Feature attribution extraction methods provide information on which input features are mainly used by machine learning models to make a prediction for a single sample [17]. When extracting feature attributions from neural networks, we can divide the extraction methods into two classes. The first class consists of gradient-based methods. Neural networks typically consist of multiple hidden layers connecting the input layer with the output layer. The output of each inner layer becomes the input of its subsequent layer, passing the input through all layers. Each of these layers has weights that are learned when the model is trained. The weights define the relation between the input and output of a layer. The partial derivative with respect to one of the input features of all these weights combined is the gradient for that feature. Intuitively, one might think about the gradient of a feature as the amount of change in the predicted output for a minor change in the input feature. Vanilla [18] is the oldest and simplest gradient-based method. It gives each input feature an attribution by calculating the gradients of the neural network weights for the given input sample. Another method that is often used is Integrated Gradients (IG) [19]. IG is a path-attribution method: it integrates the Vanilla gradients over a range of input samples. The input samples are constructed by taking samples on the linear interpolation path between a baseline input (e.g., 0 for all features) and the original sample. A third gradient-based method, SmoothGrad (SG) [20], is designed to reduce noise in the feature attributions by adding noise to the input. SG samples similar inputs by adding random noise to the original input and averages the Vanilla feature attributions for those samples. The second class of feature attribution extraction methods is model-agnostic methods. These treat the model as a black box and only require the ability to make predictions with a chosen input. One example is SHAP [21], which computes feature attributions by looking at the change in the predicted output probability when part of the feature values is replaced by their baseline value. In our case, the baseline is a distribution of actual input samples: during each iteration, a random feature value is chosen from the input distribution as the baseline. This method is repeated multiple times to average the influence of the randomly chosen baseline.

In this study, we have used interpretable deep learning techniques to understand TCR–epitope binding by applying feature attribution extraction methods to two state-of-the-art TCR–epitope prediction models: ImRex [4] and TITAN [5]. ImRex is a convolutional neural network (CNN) that follows the general design of CNNs for image processing. ImRex converts CDR3 and epitope sequences into interaction maps by calculating the pairwise difference between selected physicochemical properties (hydrophobicity, hydrophilicity, mass, and isoelectric point) of the amino acids of both sequences. This interaction map can be interpreted as a multi-channel image, with each channel corresponding to a specific physicochemical property (Figure S1), after which TCR–epitope binding prediction is performed using a multi-layer CNN. TITAN is based on one-dimensional CNNs using a contextual attention mechanism. The CDR3 and epitope sequences are encoded using a trainable embedding and separately fed into multiple one-dimensional convolutional layers, followed by a context attention layer that uses the epitopes as context for the TCR sequences and vice versa. The attention weights of both sequences are concatenated, and a stack of dense layers is used to output the binding probability. Here, we have applied several feature attribution extraction methods to ImRex and TITAN to obtain insights into the performance of TCR–epitope binding prediction by state-of-the-art machine learning models and investigate the biological patterns that underlie TCR–epitope binding.

## 2. Results

### 2.1. Molecular distances correlate with recognition patterns in TCR–epitope complexes

While the complete TCR sequence might influence the recognition of a given epitope, previous studies have shown that the CDR3 is the main binding region [2,3]. This is illustrated in Fig. 1 showing a bound TCR–epitope complex, with the CDR3 region a sequence of 14 amino acids and the epitope a sequence of 9 amino acids. A molecular interaction between two protein sequences can only occur when there is close contact between amino acids of both sequences. The distance between residues from both sequences can therefore be used as a measure for indicating which amino acids are likely responsible for the interaction. Furthermore, it can be expected that any model that attempts to predict this interaction must make use of these residues, and thus the pairwise distances between amino acids of the TCR and the epitope can be used as a metric for evaluating the learned patterns within a model. To this end, we collected 105 solved TCR–epitope structures from the public RCSB Protein Data Bank (PDB) database [23] as ground truth data.

### 2.2. Feature attribution extraction methods reveal interacting residues in the ImRex model

We retrained the ImRex and TITAN models and evaluated them with epitope-grouped cross-validation, i.e. the train and test datasets do not share samples with the same epitope sequence. The performance is always given in the format: $metric_{model} = mean \pm standard\ deviation$. Both models achieve an average receiver-operating characteristic (ROC) area under the curve (AUC) and precision-recall (PR) AUC of less than 56% ($ROC\ AUC_{ImRex} = 0.550 \pm 0.027$; $ROC\ AUC_{TITAN} = 0.559 \pm 0.050$; $PR\ AUC_{ImRex} = 0.556 \pm 0.033$; $PR\ AUC_{TITAN} = 0.541 \pm 0.049$), which shows that TCR–epitope binding prediction is still a very difficult task, even for state-of-the-art machine learning models. Given the low performance of these methods, it is unclear whether these models are truly learning the interaction. While it can be expected that these models may perform better with an increase in training data, a thorough evaluation of their current capacities is warranted. If these models are capturing the relevant molecular patterns, then they should be making use of those residues within the TCR and epitope sequences that are driving the interaction. Comparing the amino acid usage of the prediction tools with

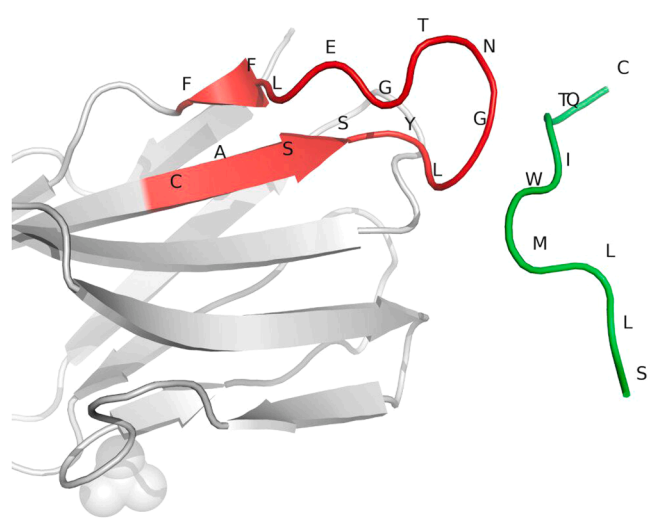


**Fig. 1. Molecular complex of a TCR–epitope interaction.** The 3D image of the PDB complex 2P5W [22,23]. Only the TCR beta chain and the epitope are shown, with the CDR3 region of the TCR beta chain colored red and the epitope colored green. The amino acids of both sequences are labeled with their one-letter abbreviation.

the actual distance between the amino acids may help to evaluate and to improve the performance and robustness of these tools.

To explore whether feature attribution extraction methods can be applied to these models, and which one is the most relevant, four common attribution extraction methods were applied to ImRex and TITAN. Extracting feature attributions for a sample from the ImRex model results in a 4-channel 2D feature attribution matrix with the same dimensions as the input sample. The attributions of the four physicochemical properties are summed per amino acid pair. For each pairwise combination of amino acids from the CDR3 and epitope sequence, the feature attribution extraction method returns a value that represents how much that input feature contributed to the prediction for the given input sample. For comparison to TITAN, the pairwise feature attributions from ImRex were merged per amino acid by taking the maximum of all feature attribution values for that amino acid. TITAN only uses the amino acid sequences as input, which results in one feature attribution value per amino acid. For each sample, we computed the Pearson correlation coefficient between the feature attributions and the residue distance of the amino acids. The correlation coefficient is multiplied by $-1$ to represent the correlation with the residue proximity, as we expect an amino acid pair with a small distance to have a higher feature attribution. Also the pairwise residue distance was merged per amino acid by taking the maximum value for each residue.

Figure 2 shows the correlation of each sample extracted with 4 different feature attribution extraction methods: Vanilla, Integrated Gradients (IG), SmoothGrad (SG), and SHAP from ImRex (Fig. 2a and b) and TITAN (Fig. 2c). For ImRex, there is a lower correlation when comparing pairwise feature attributions and residue proximity (Fig. 2a) compared to per residue attributions and proximity (Fig. 2b) for all four methods, but the variation increases as well. Although less significant in the per residue case, feature attributions extracted with SG are most correlated with the residue proximity ($Corr_{ImRex\ pairwise\ SG} = 0.435 \pm 0.175$; $Corr_{ImRex\ per\ residue\ SG} = 0.516 \pm 0.186$). This can be explained by the de-noising effect of SG, as it will give higher attributions to the most important features and less to the other features. An additional explanation is that SG extracts the feature attributions multiple times with a slightly different input, which can make the algorithm more robust against high small-scale fluctuations in the gradient. For ImRex, we find that IG results in a lower correlation than Vanilla ($Corr_{ImRex\ pairwise\ IG} = 0.229 \pm 0.117$; $Corr_{ImRex\ per\ residue\ IG} = 0.384 \pm 0.230$; $Corr_{ImRex\ pairwise\ Vanilla} = 0.318 \pm 0.146$; $Corr_{ImRex\ per\ residue\ Vanilla} = 0.474 \pm 0.205$). Feature attributions extracted by SHAP from ImRex are very weakly correlated with the residue proximity ($Corr_{ImRex\ pairwise\ SHAP} = 0.096 \pm 0.102$; $Corr_{ImRex\ per\ residue\ SHAP} = 0.214 \pm 0.205$), our results show that feature attributions extracted with gradient-based methods have a higher correlation with residue proximity. The feature attribution extraction methods on TITAN all give a similar result and the correlation is only a limited amount higher than the random correlation ($Corr_{TITAN\ Vanilla} = 0.299 \pm 0.148$; $Corr_{TITAN\ IG} = 0.297 \pm 0.154$; $Corr_{TITAN\ SG} = 0.342 \pm 0.128$; $Corr_{TITAN\ SHAP} = 0.293 \pm 0.116$; $Corr_{random} = -0.001 \pm 0.212$). An overview of the correlation of all 9 feature attribution extraction methods we tested can be seen in Figure S2, a detailed ImRex feature attribution matrix for a single sample can be seen in Figure S3 and the feature attributions from different methods for TITAN can be seen in Figure S4.

### 2.3. Feature attributions reveal important residues for each prediction

The remainder of all experiments were run using the SG feature attribution extraction method because it gives the highest correlation for both ImRex and TITAN.

The pairwise feature attributions from ImRex were merged per amino acid. For both ImRex and TITAN, all feature attributions were normalized per input sample by dividing them by the highest feature attribution value for that sample. This results in feature attributions ranging between 0 and 1, where the amino acid with the highest attribution gets a normalized value of 1, although the amino acid with the lowest attribution will not necessarily get a value of 0. The pairwise residue proximity is calculated for all pairs of amino acids from both sequences. A single value for each amino acid is derived in a similar way as for the pairwise feature attributions from ImRex: for each amino acid, the minimal distance (maximal proximity) to the amino acids from the other sequence is taken. The residue proximity is derived from the residue distance by taking 1/distance. It the end, the distance array is normalized per sample in the same way as the feature attributions. Figure 3 shows the feature attributions extracted from ImRex and TITAN with SG for the PDB complex 2P5W [22]. For the epitope, we can see that the amino acids that are most important for TITAN are also in close

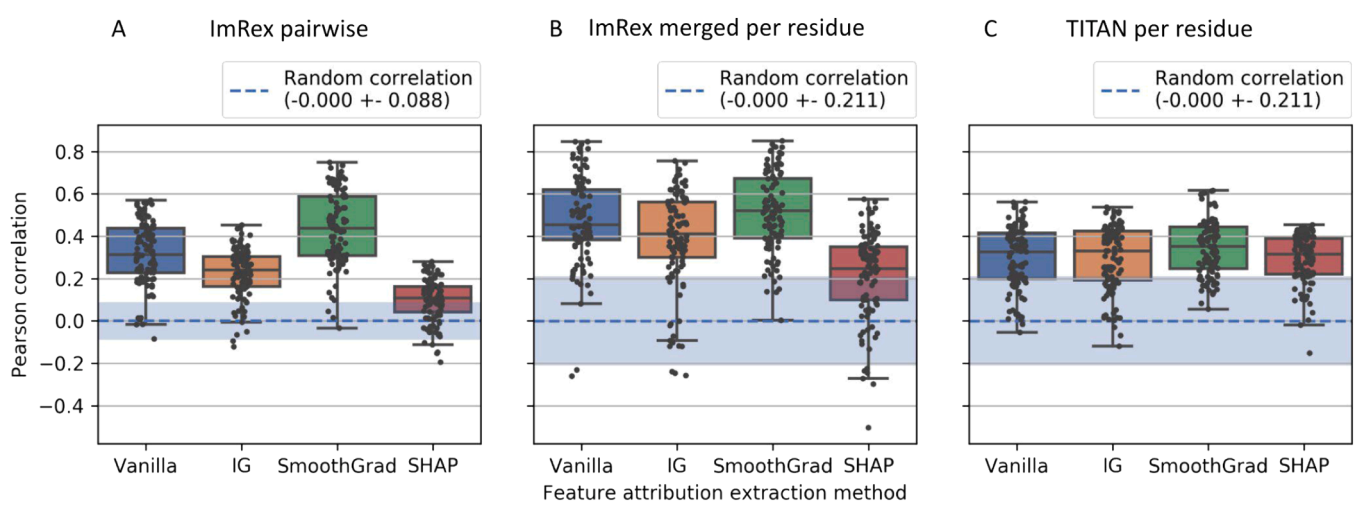


**Fig. 2. Pearson correlation coefficient of feature attributions and residue proximity extracted with different methods from ImRex and TITAN.** The Pearson correlation is calculated between the feature attributions extracted with a specific method and the residue proximity between the amino acids of the TCR and epitope sequences. A boxplot is shown for each method giving the correlation over all 105 complexes. The random correlation is calculated by taking the Pearson correlation between a random feature attribution matrix and the actual residue proximity for each sample, repeated multiple times. Thus, this represents the correlation when the feature attribution extraction method would give a random output. Boxplots are constructed as follows: the box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to the last datum before 1.5 times the interquartile range above/below the box. The random correlation is given as mean and standard deviation. **(A)** Correlation for ImRex using pairwise feature attributions and residue proximity. **(B)** Correlation for ImRex using feature attributions and residue proximity merged per residue. **(C)** Correlation for TITAN using feature attributions and residue proximity per residue.

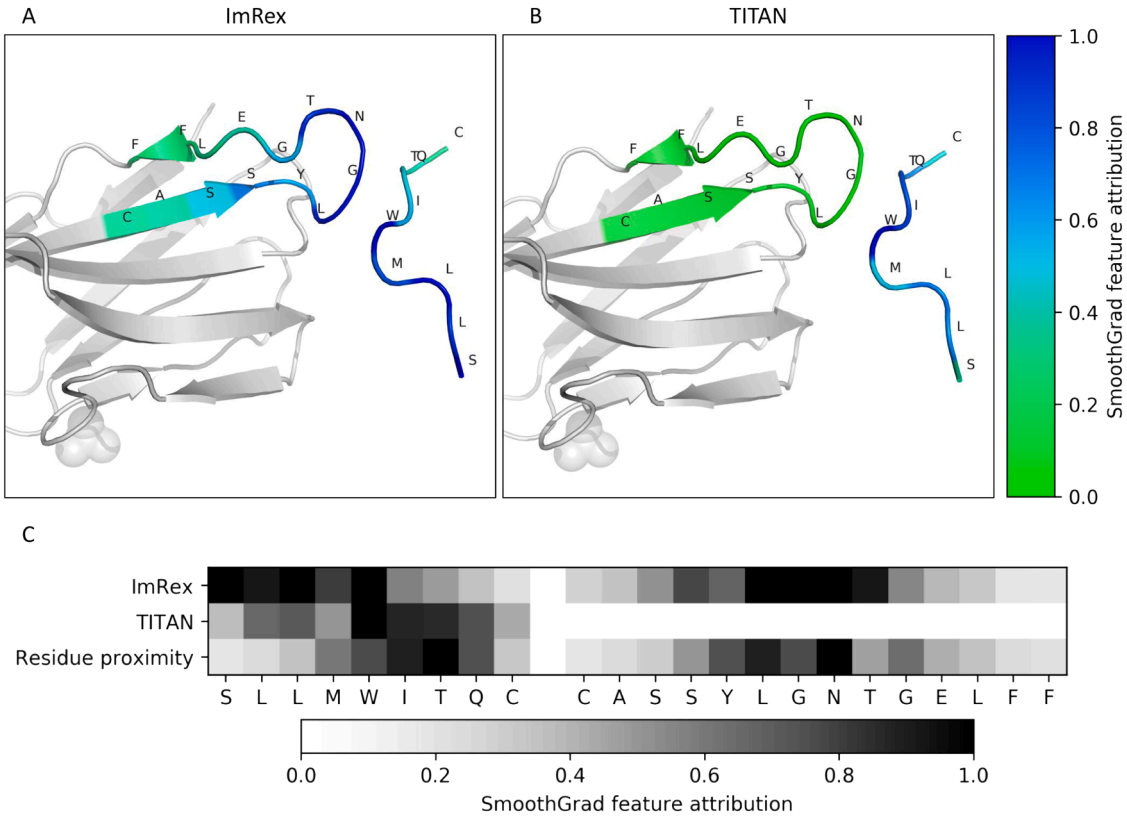


**Fig. 3. Feature attributions extracted with SG from PDB complex 2P5W.** Feature attributions of **(A)** ImRex and **(B)** TITAN extracted with SG from the prediction for the PDB complex 2P5W [22,23] and shown on its molecular complex. Only the TCR beta chain and the epitope are shown. The TCR beta chain is colored gray, except for the CDR3 region which is colored according to a color range derived from the normalized feature attributions. The epitope is colored in the same way. **(C)** Feature attributions per position and model for the same complex together with the residue proximity. A higher value represents a larger feature attribution or residue proximity.

contact with the CDR3 region, which is not the case for all important amino acids according to ImRex. ImRex focuses more on the residues at the beginning of the sequence. The attributions for the CDR3 region are very different, where ImRex mainly focuses on the middle part of the CDR3 sequence while for TITAN all attributions are almost zero.

### 2.4. Important residues are distinct for each TCR-epitope model

The findings derived from a single complex are representative of the full dataset, as can be seen in Fig. 4. We calculated the average attributions of the 105 samples and the average residue proximity. The epitope and CDR3 sequences were padded left and right after centering them, the padded positions are ignored when calculating the average feature attributions or residue proximity. This shows that the amino acids of the epitope on positions 6 to 9 are on average the closest to the CDR3 region. ImRex focuses more on the first part of the epitope while TITAN uses mostly the middle part. For the CDR3, the attributions from ImRex are similar to the 3D distance but more focused on the middle region. The average attributions from TITAN are very close to zero for each position of the CDR3 sequence, which suggests that TITAN primarily uses the epitope sequence to make its predictions.

We calculated the Pearson correlation coefficient between the feature attributions extracted from ImRex and TITAN and the proximity of the amino acids of the CDR3 and epitope sequence. The attributions of ImRex and the residue proximities were first merged per amino acid. This results in a higher correlation for ImRex when looking at both sequences together ($Corr_{ImRex} = 0.516 \pm 0.186$; $Corr_{TITAN} = 0.342 \pm 0.128$) (Fig. 5). When only considering the epitope, the feature attributions from ImRex are not more correlated with the residue proximity than random attributions. This is caused by ImRex mainly focusing on the first residues of the epitopes. TITAN mainly uses the residues close to the CDR3 sequence ($Corr_{ImRex\ ep} = 0.045 \pm 0.446$; $Corr_{TITAN\ ep} = 0.639 \pm 0.181$). On the other hand, the correlation for the attributions of the CDR3 sequence is very high for ImRex and similar to random for TITAN ($Corr_{ImRex\ CDR3} = 0.807 \pm 0.104$; $Corr_{TITAN\ CDR3} = -0.187 \pm 0.301$). This can be explained by the fact that TITAN always gives a near-

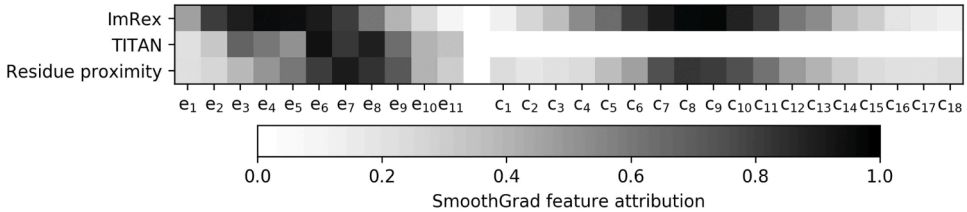


**Fig. 4. Average feature attributions per position.** The average feature attribution per position and model extracted with SG, also the average residue proximity is given. Both the epitope and CDR3 sequence is centered and padded left and right separately. The average is calculated by only looking at the feature attributions from sequences that do not have padding on that position.

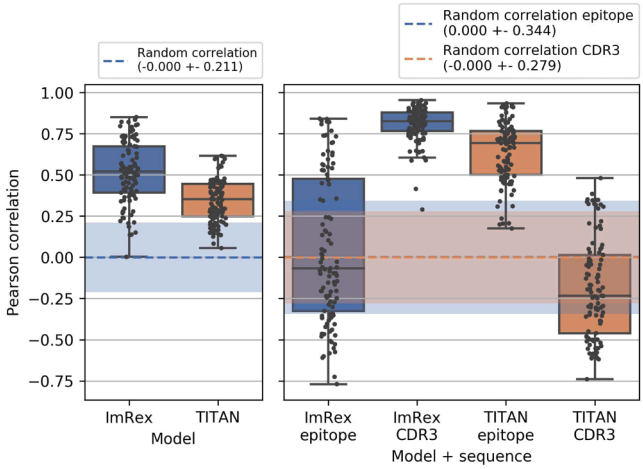


**Fig. 5. Pearson correlation between attributions and residue proximity for models and sequences.** The correlation is calculated between the feature attributions extracted from ImRex and TITAN with SG and the proximity of the amino acid pairs from both sequences. For ImRex, the attributions were first merged per amino acid to allow comparison with TITAN. On the right, the correlation for both the epitope and CDR3 sequences separately is shown. The random correlation is calculated by taking the correlation between a random feature attribution array and the actual residue proximity for each sample, repeated multiple times. Thus, this represents the correlation when the feature attribution extraction method would give a random output. Boxplots are constructed as follows: the box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to the last datum before 1.5 times the interquartile range above/below the box. The random correlation is given as mean and standard deviation.

zero attribution to the entire CDR3 sequence.

### 2.5. Feature attributions explain model performance

As already mentioned in Section 2.2, we evaluated both the ImRex and TITAN models on the task they were designed for: TCR–epitope interaction prediction. We use epitope-grouped cross-validation for all models which means that none of the epitopes will occur in samples of both the train and test split. Thus, for these performance metrics, all test samples concern an unseen epitope. The default TITAN model has a slightly better average ROC AUC and ImRex has a slightly better average PR AUC ($ROC\ AUC_{ImRex} = 0.550 \pm 0.027$; $ROC\ AUC_{TITAN} = 0.559 \pm 0.05$; $PR\ AUC_{ImRex} = 0.556 \pm 0.033$; $PR\ AUC_{TITAN} = 0.541 \pm 0.049$). The variance of both metrics is higher for TITAN, which also exhibited cross-validation splits with a performance below 0.500 (Fig. 6). We previously found that TITAN primarily considers the epitope sequence to make its prediction. Here we see that it is still able to get decent performance when evaluated with epitope-grouped cross-validation, even though a model that only uses the epitope sequence is not expected to achieve a good performance in this setting. We investigated this in more detail by training and evaluating the TITAN model on exactly the same data and cross-validation splits as ImRex (the 'TITAN on ImRex data' model in Fig. 6, also see figures S5 and S6 for average feature attributions and correlation of this model). When the TITAN model is trained on the same data as ImRex, the performance on both metrics drops ($ROC\ AUC_{TITAN\ on\ ImRex\ data} = 0.523 \pm 0.013$; $PR\ AUC_{TITAN\ on\ ImRex\ data} = 0.514 \pm 0.012$). This indicates that there might be an information leakage issue in the TITAN training data, which explains why a model that completely focuses on the epitope is still able to get a good performance on an unseen-epitope task. At last, we trained and evaluated the TITAN model on a third dataset with scrambled CDR3 sequences. This dataset is the same as the original dataset (including cross-validation train-test splits), but the CDR3 sequences are randomized by sampling random amino acids to create a new sequence of the same length for each original sequence while ensuring that the distribution of the amino acids from the original CDR3 sequences is retained. The performance of the TITAN model trained on this random data is very similar to the performance of the original TITAN model ($ROC\ AUC_{TITAN\ scrambled\ TCRs} = 0.559 \pm 0.045$; $PR\ AUC_{TITAN\ scrambled\ TCRs} = 0.540 \pm 0.046$), which is only possible when the CDR3 sequence does not contribute to the predictions. Thus, this confirms that our feature attribution extraction method made correct conclusions about the usage of the CDR3 sequence.

## 3. Discussion

Although recently multiple improvements have been made in unseen-epitope TCR interaction prediction, the performance of the current state-of-the-art models is still limited [4,5,12,16]. One of the reasons is that the determining factors for these molecular interactions are still unknown to both human experts and machine learning models. We presented a method that can extract which amino acids of the input were mainly used by the models to make their prediction. Highlighting

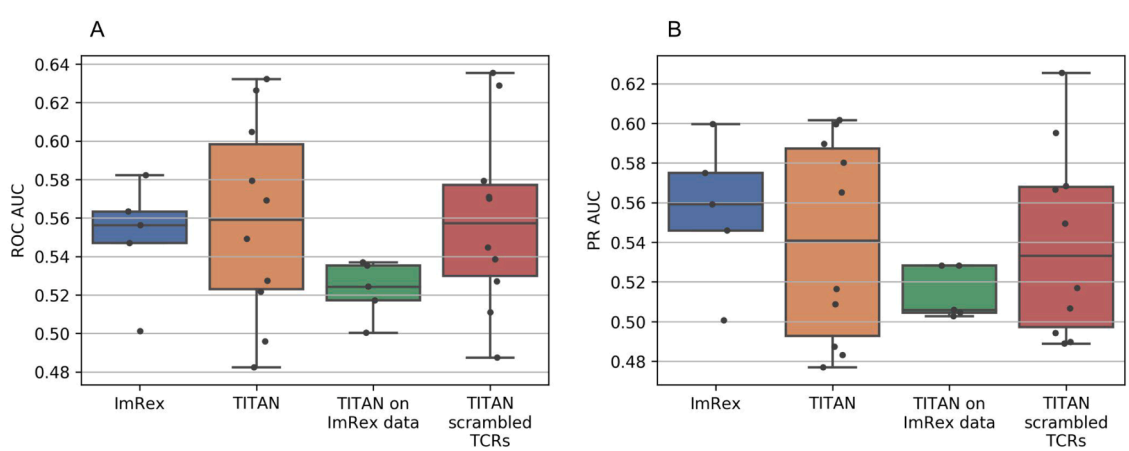


**Fig. 6. Comparison of model performance.** The TCR–epitope interaction prediction performance of the different models measured with **(A)** ROC AUC and **(B)** PR AUC. 'ImRex' and 'TITAN on ImRex data' were both trained on the ImRex dataset and evaluated with 5-fold epitope-grouped cross-validation. 'TITAN' was trained on the original TITAN dataset and evaluated with 10-fold epitope-grouped cross-validation, 'TITAN scrambled TCRs' was trained on the TITAN dataset with scrambled CDR3 sequences and also evaluated with 10-fold epitope-grouped cross-validation. Boxplots are constructed as follows: the box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to the last datum before 1.5 times the interquartile range above/below the box.

those feature attributions on the molecular complex gives additional information to the domain expert about why the prediction was made and can give new insights into the factors that determine TCR affinity on a molecular level. However, these results should also be interpreted in the context of the models' performances.

Using the actual distance between the amino acids of the TCR and epitope sequences as ground truth, we were able to compare the different feature attribution extraction methods and prediction models. We found that SG gives the highest correlation for ImRex and TITAN. This is not unexpected because it is designed to reduce noise, making it focus more on the most important residue pairs and less on the others. This is similar to the residue proximity; a limited number of residues from both sequences are very close to each other and the proximity between the other residues decreases quickly. Another reason can be that SG samples multiple similar inputs with random noise, this should make the algorithm more robust against high small-scale fluctuations in the gradient. On ImRex, all gradient-based methods have a significantly higher correlation with the residue proximity than SHAP. We reason that these methods are better at finding more detailed attributions because they have access to the gradients of the models while SHAP only compares the predicted output for altered inputs.

We found that, on average, ImRex uses mainly the start and middle amino acids of the epitope, while the distance is smaller for amino acids on positions 6 to 9 (out of 11). We repeated our feature attribution analysis on an ImRex model trained on samples with scrambled epitope sequences and found that the epitope attributions are now more centered (Figure S5). The remaining feature attributions after scrambling can possibly be explained by the padding that is added to the start and the end of epitope sequences shorter than 11 residues. These positions have lower variability in input values during training, which might make them less interesting for the model. This can also partially explain the epitope feature attributions of the regular ImRex model but the difference with the model trained on scrambled epitopes suggests that ImRex might also find different patterns. ImRex is revealed to use the amino acids of the CDR3 region similarly to the average residue proximity with somewhat more attribution to the middle part and less attribution to the outer parts of the sequence. We repeated our feature attribution extraction again on an ImRex model trained on samples with scrambled TCRs (Figure S5). The difference in the feature attributions of the middle and outer positions of the CDR3 is now much less extreme, the remaining attributions can possibly also be explained by the padding. If these attributions are influenced by the residue variability when training, then we hypothesize that this can also be the case for the CDR3 feature attributions of the regular ImRex model. Next to the reduced variability from padding, the center of the CDR3 sequence has much more variability in the input samples than the start and end positions, which makes it more important for the model. This is not necessarily a problem, but one needs to take into account that padding and training input distributions might influence the focus of a machine learning model.

For TITAN, we found that the amino acid usage of the epitope sequence is similar to the residue proximity, but it does not consider the CDR3 sequence. However, when testing TITAN using epitope-grouped cross-validation, unexpectedly its performance was still similar to ImRex. Training TITAN on the ImRex data resulted in a much lower performance, which leads us to hypothesize that the unexpected performance can be explained by how the TITAN training and evaluation data was constructed. TITAN creates its negative samples in a different way than ImRex. TITAN uniformly samples a random epitope for each TCR sequence, while ImRex samples a random epitope with the same probability as in the positive dataset. This can be seen in supplementary table S1: for each epitope, the number of positive samples is between 15 and 400 (which is expected due to the data preprocessing), but the number of negative samples is similar for all epitopes. This leads to a large imbalance between the number of positive and negative samples for most epitopes. Note that the same imbalance is also present in the training data (although with other epitopes). This imbalance is learned by the TITAN model that thereby completely focuses on the epitope sequence, as we found by extracting the feature attributions. The last column of supplementary table S1 shows that TITAN almost always gives a negative prediction for most epitopes except for a few (TLIGDCATV, RQLLFVVEV, EPLPQGQLTAY, and LSDDAVVCFNSTY in this specific cross-validation split) for which it almost always gives a positive prediction.

Even though the dataset is imbalanced, this does not fully explain the good test performance. TITAN splits its data into two datasets, a train and test set, where no epitopes are shared among each. The model is trained on the train dataset and at the end of each epoch tested on the test dataset. When retraining the model ourselves, we saw that the performance on the test dataset is very unstable across multiple epochs and does not converge. After training for a given number of epochs, the test performance of the best epoch is selected and reported as the final test performance. Therefore, we hypothesize that the model tries to give a high prediction to random epitopes or patterns in the train data, repeatedly changing this every epoch (which results in the very high variation in performance across epochs). In the end, the epoch where the patterns selected from the train dataset gave (by chance) the best performance on the test dataset is chosen. If true, this could have been avoided by using an independent validation dataset to determine the best epoch. Afterward, the performance could have been given by applying the model from that epoch on the test set.

## 4. Conclusions

We have shown that applying feature attribution extraction methods are useful to improve the explainability for protein interaction prediction, as applied here to the TCR–epitope problem. The attributions from the gradient-based methods are closer to what we expect than the attributions from the model-agnostic method SHAP. Showing feature attributions on the TCR–epitope molecular complex can be useful to gain more knowledge about why specific protein sequences interact. Extracting feature attributions is a good way to verify a model and data and to check that it works as expected. This is especially true for challenging problems, where small hard-to-detect issues in the dataset balance or evaluation methods can compound to inaccurate results.

## 5. Methods

### 5.1. Data

#### 5.1.1. Molecular complex data

A collection of TCR–epitope MHC class I complexes with links to their RCSB Protein Data Bank (PDB) [23] entry was downloaded from the TCR3d database [24] and all non-human entries were removed. The TCR beta chain, epitope chain, and location of the CDR3 region were manually selected from the PDB complexes with the help of the IMGT/3Dstructure-DB [25–27] online tool. Finally, we ended up with 105 unique complexes consisting of 51 unique CDR3 sequences and 62 unique epitopes.

#### 5.1.2. ImRex training data

We used the data that was also used in the ImRex [4] paper. It uses the VDJdb dataset from August 2019 [28] and is filtered on samples with human TCR beta sequences. All samples from the 10x Genomics study [29] were excluded and only samples with a length between 10 and 20 and 8–11 for respectively the CDR3 and epitope were kept. The data was downsampled to have at most 400 samples per epitope and negative data was generated by shuffling. We performed one additional filtering step: all samples that are also present in the molecular complex data (based on the CDR3 sequence) were removed, which reduced the positive dataset size from 6,702 to 6,656 samples.

We also created two additional datasets: one with scrambled epitope

sequences and one with scrambled CDR3 sequences. We used the final ImRex dataset but replaced all epitope/CDR3 sequences with a random combination of amino acids of equal length. The amino acid distribution of the epitope/CDR3 sequences of the original dataset was kept.

Each model trained on the ImRex data was evaluated with 5-fold epitope-grouped cross-validation, as per the original study. This divides the data into five groups with about the same amount of epitopes and a similar distribution of number of samples per epitope. Samples with the same epitope are all put in the same group.

#### 5.1.3. TITAN training data

For TITAN [5], we use their 'strictsplit' data which is a combination of two datasets: the VDJdb [28] and a COVID-19 specific dataset published by the ImmuneCODE project [30]. Both datasets were filtered on human TCR beta sequences, all epitopes with less than 15 associated TCRs were removed and the data was downsampled to have at most 400 samples per epitope. For the COVID-19 dataset, only samples with a single unique epitope were kept and unproductive samples were excluded. After merging both datasets, negative samples were generated by shuffling. This means that a negative sample is generated for each positive sample, pairing the original CDR3 sequence with a random, different epitope from the positive dataset. The only additional filtering step we performed on the data was removing all samples that are also present in the molecular complex data, which reduced the positive dataset size from 23,145 to 23,125 samples.

We also created an additional dataset with scrambled CDR3 sequences. We used the final TITAN dataset but replaced all CDR3 sequences with a random combination of amino acids of equal length. The amino acid distribution of the CDR3 sequences of the original dataset was kept.

Models trained on any of the TITAN datasets are always evaluated with 10-fold epitope-grouped cross-validation. The 10 groups created by TITAN were kept.

### 5.2. ImRex model training

We retrained ImRex with the same parameters as the final published model [31]. This is the default 'padded model', has a batch size of 32, is trained for 20 epochs, uses ReLU activation functions, has convolutional layers with depth 128, 64, 128, and 64, has a dropout rate of 0.25 for the convolutional layers, uses a learning rate of 1e-4, a regularization of 0.01 on all layers and uses the RMSProp optimizer [32].

### 5.3. TITAN model training

TITAN was trained with the parameter configuration of the AA CDR3 case as explained in their paper (the configuration on their online repository led to similar results). We chose this setting because it has the same performance as its alternative using SMILES strings [33] and/or the full TCR sequence on the 10-fold cross-validation strictsplit test [5], while the feature attributions can be directly compared to the "ground truth" molecular distance data. We set the padding of the CDR3 and epitope to 25 instead of 500, otherwise, we could not achieve a better than random performance. The epitope is not encoded as SMILES, the size of the dense hidden layers is 368 and 184, a ReLU activation function is used, a dropout of 0.5, a batch size of 512 without normalization, a learning rate of 1e-4, the attention size for both sequences is 16, the embedding size for both sequences is 26, the epitope and CDR3 kernel sizes are both [3, 26], [5, 26], and [11, 26] and the epitope and CDR3 embeddings are both learned during training. Note that the pretrained version of the TITAN model might give different results.

Retraining TITAN on its own data was done with the original 10-fold epitope-grouped cross-validation splits provided by the authors. Retraining TITAN on the ImRex data was done with the 5-fold epitope-grouped cross-validation splits generated by ImRex.

### 5.4. Feature attribution extraction

The input for ImRex can be represented as an image, so it is possible to apply feature attribution extraction methods made for image classification CNNs. These methods give an attribution to each of the input pixels for the prediction of a single sample. In our case, every pixel is a pairwise combination of an amino acid from both sequences. The value of the attribution represents how much each pixel contributed to the prediction.

In this research, we focused on two kinds of feature attribution extraction methods: a set of methods based on the neural network gradients and a model-agnostic method: SHAP [21]. Gradient-based methods can only be used on models with a differentiable input, whereas SHAP can be used for any type of black box model. We compared 8 different feature attribution extraction methods that are based on gradients (Vanilla [18], Integrated Gradients (IG) [19], SmoothGrad [20], SmoothGradIG [19,20], GuidedIG [19,34], BlurIG [19,35], SmoothGradBlurIG [19,20,35] and XRAI [36]) and the SHAP method using random sampling from the dataset as background. Of these, for conciseness, four representative methods were used in the results section (Vanilla, IG, SmoothGrad, and SHAP), while results for all methods are available as supplementary information. The Python package *shap* (version 0.40.0) [21] was used for the implementation of the SHAP algorithm and the other methods were implemented with the Python package *saliency* (version 0.1.3) [18,20,34–37].

TITAN uses a 1D categorical input: a concatenated list of the amino acids from the epitope and CDR3. Therefore, there will only be one feature attribution value for each amino acid. Because gradient-based feature attribution extraction methods require a differentiable input, the embedded input was used with TITAN.

### 5.5. Feature attribution evaluation

We evaluated the feature attribution extraction methods by comparing them to the distance between the amino acids of both sequences in the molecular complex. The distance between two amino acids is the minimal distance between two atoms of each amino acid in ångströms. This results in a distance matrix with values between 2.3 Å and 30 Å. The amino acids that are closer to the other sequence are also more important for the interaction. Therefore, we inverted each value of the distance matrix by taking 1/value to get the residue proximity. The feature attributions and the residue proximity matrix are both normalized by dividing them by their maximum value. This results in all values being between 0 and 1 and the largest value is always equal to 1.

$$residue_proximity_{ij} = p_{ij}$$

$$p_{ij} = \frac{1}{d_{ij}},\ i \in \{1, ..., n\},\ j \in \{1, ..., m\}$$

$$normalized_residue_proximity_{ij} = np_{ij}$$

$$np_{ij} = \frac{p_{ij}}{\max(p)},\ i \in \{1, ..., n\},\ j \in \{1, ..., m\}$$

$$normalized_feature_attribution_{ij} = nfa_{ij}$$

$$nfa_{ij} = \frac{fa_{ij}}{\max(fa)},\ i \in \{1, ..., n\},\ j \in \{1, ..., m\}$$

With n the length of the epitope, m the length of the CDR3 sequence, $d_{ij}$ the distance between amino acid i of the epitope and amino acid j of the CDR3 sequence and $fa_{ij}$ the feature attribution of the amino acid pair (i, j).

To quantify the correlation between the feature attributions and the residue proximity the Pearson correlation coefficient was calculated between the feature attributions and the residue distance for each sample and multiplied by −1. This results in a higher, positive value when there is a higher correlation between the feature attributions and

the residue proximity.

### 5.6. 1D feature attributions

To compare the feature attributions and the correlation with the residue proximity between ImRex and TITAN we converted the 2D ImRex attributions and distance matrices to a 1D array by merging the values per amino acid and concatenating the result of both sequences. For each amino acid, the maximum feature attribution and minimum distance with respect to every amino acid from the other sequence was taken. Those 1D arrays (of length $n + m$, with n the length of the epitope and m the length of the CDR3) were inverted and normalized in the same way as the 2D matrices.

### 5.7. Molecular complex highlighting

The normalized 1D feature attributions from both models are shown on the molecular complex with PyMol (version 2.3.0) [38]. For clarity, we only show the TCR beta sequence and the epitope. The TCR beta chain is colored gray, except the CDR3 region which is colored according to a color range derived from the normalized feature attributions, green for a feature attribution of zero to blue for a feature attribution of one. The epitope is colored with the same color range. The amino acids of both sequences are labeled with their one-letter abbreviation.

## Declarations

### Data and code availability

The datasets analyzed during the current study and all scripts used to obtain the results are licensed under the Apache License 2.0 and are available on GitHub at https://github.com/PigeonMark/McFAE [39] and on Zenodo at https://doi.org/10.5281/zenodo.6500495 [40]. All code is written in Python 3.7 [41]. PyTorch (version 1.10.0) [42], TensorFlow (GPU version 2.6.0) [43] and Keras (version 2.6.0) [44] were used for implementing the neural network architectures and model training. NumPy (version 1.18.1) [45] and pandas (version 1.0.1) [46, 47] were used for data processing. SHAP (version 0.40.0) [21] was used to extract SHAP feature attributions and Saliency (version 0.1.3) [37] was used to extract the other feature attributions. Matplotlib (version 3.1.3) [48], seaborn (version 0.10.0) [49], and Pillow (version 7.0.0) [50] were used to create the figures. PyMol (version 2.3.0) [38] was used to create the molecular complex figures.

### Funding

This work was supported by the Flemish Government (AI Research Program); and the iBOF Modulating Immunity and the Microbiome for Effective CRC Immunotherapy (MIMICRY) Project. The computational resources and services used in this work were provided by the HPC core facility CalcUA of the Universiteit Antwerpen, and VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government. The authors acknowledge support from Biomina, the Biomedical Informatics Core Facility of the Universiteit Antwerpen.

### Materials & correspondence

Correspondence to Pieter Meysman.

## CRediT authorship contribution statement

**Ceder Dens:** Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Wout Bittremieux:** Conceptualization, Writing – review & editing, Supervision. **Fabio Affaticati:** Investigation. **Kris Laukens:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition. **Pieter Meysman:** Conceptualization, Writing – review & editing, Supervision.

## Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

KL and PM hold shares in ImmuneWatch BV, an immunoinformatics company.

## Data availability

All datasets and scripts are available on GitHub at https://github.com/PigeonMark/McFAE and on Zenodo at https://doi.org/10.5281/zenodo.6500495.

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.immuno.2023.100027.

## References

[1] Alberts B. Molecular biology of the cell. 6th ed. New York, NY: Garland Science, Taylor and Francis Group; 2015.
[2] Krogsgaard M, Davis MM. How T cells "see" antigen. Nat. Immunol. 2005;6(3). https://doi.org/10.1038/ni1173. Art. no. 3, Mar.
[3] Davis MM, Bjorkman PJ. T-cell antigen receptor genes and T-cell recognition. Nature 1988;334(6181). https://doi.org/10.1038/334395a0. Art. no. 6181, Aug.
[4] Moris P, et al. Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. Brief Bioinform 2021;22(4):bbaa318. https://doi.org/10.1093/bib/bbaa318. Jul.
[5] Weber A, Born J, Rodriguez Martínez M. TITAN: t-cell receptor specificity prediction with bimodal attention networks. Bioinformatics 2021;37(Supplement_1):i237–44. https://doi.org/10.1093/bioinformatics/btab294. Jul.
[6] Bi J, Zheng Y, Yan F, Hou S, Li C. Prediction of epitope-associated TCR by using network topological similarity based on deepwalk. IEEE Access 2019;7:151273–81. https://doi.org/10.1109/ACCESS.2019.2948178.
[7] De Neuter N, et al. On the feasibility of mining CD8+ T cell receptor patterns underlying immunogenic peptide recognition. Immunogenetics 2018;70(3):159–68. https://doi.org/10.1007/s00251-017-1023-5. Mar.
[8] Dash P, et al. Quantifiable predictive features define epitope-specific T cell receptor repertoires. Nature 2017;547(7661). https://doi.org/10.1038/nature22383. Art. no. 7661, Jul.
[9] Glanville J, et al. Identifying specificity groups in the T cell receptor repertoire. Nature 2017;547(7661). https://doi.org/10.1038/nature22976. Art. no. 7661, Jul.
[10] Jokinen E, Huuhtanen J, Mustjoki S, Heinonen M, Lähdesmäki H. Determining epitope specificity of T cell receptors with TCRGP. bioRxiv 2019. https://doi.org/10.1101/542332. Aug. 21.
[11] Gielis S, et al. Detection of enriched T cell epitope specificity in full T cell receptor sequence repertoires. Front Immunol 2019;10:2820. https://doi.org/10.3389/fimmu.2019.02820.
[12] Springer I, Besser H, Tickotsky-Moskovitz N, Dvorkin S, Louzoun Y. Prediction of specific TCR-peptide binding from large dictionaries of TCR-peptide Pairs. Front Immunol 2020;11. Accessed: Apr. 15, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/fimmu.2020.01803.
[13] Chronister WD, et al. TCRMatch: predicting T-cell receptor specificity based on sequence similarity to previously characterized receptors. Front Immunol 2021;12. Accessed: Apr. 15, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/fimmu.2021.640725.
[14] Montemurro A, et al. NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. Commun Biol 2021;4(1). https://doi.org/10.1038/s42003-021-02610-3. Art. no. 1, Sep.
[15] Fischer DS, Wu Y, Schubert B, Theis FJ. Predicting antigen specificity of single T cells based on TCR CDR3 regions. Mol Syst Biol 2020;16(8). https://doi.org/10.15252/msb.20199416. Aug.
[16] Jurtz VI, et al. NetTCR: sequence-based prediction of TCR binding to peptide-MHC complexes using convolutional neural networks. Bioinformatics 2018. https://doi.org/10.1101/433706. preprint, Oct.
[17] C. Molnar, *Interpretable machine learning: a guide for making black box models explainable*, 2nd ed. 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book.
[18] K. Simonyan, A. Vedaldi, and A. Zisserman, 'Deep inside convolutional networks: visualising image classification models and saliency maps', *ArXiv13126034 Cs*, Apr. 2014, Accessed: Mar. 31, 2022. [Online]. Available: http://arxiv.org/abs/1312.6034.

[19] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: ICML'17: Proceedings of the 34th international conference on machine learningVol. 70. JMLR.org; Aug. 2017. p. 3319–28.
[20] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, 'SmoothGrad: removing noise by adding noise', ArXiv170603825 Cs Stat, Jun. 2017, Accessed: Mar. 31, 2022. [Online]. Available: http://arxiv.org/abs/1706.03825.
[21] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Advances in neural information processing systems. Curran Associates, Inc.; 2017. Accessed: Mar. 30, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.
[22] Sami M, et al. Crystal structures of high affinity human T-cell receptors bound to peptide major histocompatibility complex reveal native diagonal binding geometry. Protein Eng Des Sel 2007;20(8):397–403. https://doi.org/10.1093/protein/gzm033. Aug.
[23] Berman HM, et al. The protein data bank. Nucleic Acids Res 2000;28(1):235–42. https://doi.org/10.1093/nar/28.1.235. Jan.
[24] Gowthaman R, Pierce BG. TCR3d: the T cell receptor structural repertoire database. Bioinforma Oxf Engl 2019;35(24):5323–5. https://doi.org/10.1093/bioinformatics/btz517. Dec.
[25] Ehrenmann F, Lefranc M-P. IMGT/3Dstructure-DB: querying the IMGT database for 3D structures in immunology and immunoinformatics (IG or Antibodies, TR, MH, RPI, and FPIA). Cold Spring Harb. Protoc. 2011;2011(6). https://doi.org/10.1101/pdb.prot5637. p. pdb.prot5637Jan.
[26] Ehrenmann F, Kaas Q, Lefranc M-P. IMGT/3Dstructure-DB and IMGT/DomainGapAlign: a database and a tool for immunoglobulins or antibodies, T cell receptors, MHC, IgSF and MhcSF. Nucleic Acids Res 2010;38(suppl_1):D301–7. https://doi.org/10.1093/nar/gkp946. Jan.
[27] Kaas Q, Ruiz M, Lefranc M. IMGT/3Dstructure-DB and IMGT/StructuralQuery, a database and a tool for immunoglobulin, T cell receptor and MHC structural data. Nucleic Acids Res 2004;32(suppl_1):D208–10. https://doi.org/10.1093/nar/gkh042. Jan.
[28] Bagaev DV, et al. VDJdb in 2019: database extension, new analysis infrastructure and a T-cell receptor motif compendium. Nucleic Acids Res 2020;48(D1):D1057–62. https://doi.org/10.1093/nar/gkz874. Jan.
[29] 10x Genomics: a new way of exploring immunity digital. Accessed: Mar. 30, 2022. [Online]. Available: https://pages.10xgenomics.com/rs/446-PBO-704/images/10x_AN047_IP_A_New_Way_of_Exploring_Immunity_Digital.pdf.
[30] J.N. Dines et al., 'The ImmuneRACE Study: a prospective multicohort study of immune response action to COVID-19 events with the ImmuneCODE™ open access database'. medRxiv, p. 2020.08.17.20175158, Aug. 21, 2020. doi:10.1101/2020.08.17.20175158.
[31] P. Moris, 'Pretrained ImRex model', GitHub. https://github.com/pmoris/ImRex/tree/master/models/pretrained/2020-07-24_19-18-39_trbmhcidown-shuffle-padded-b32-lre4-reg001 (accessed Jun. 01, 2022).
[32] G. Hinton, 'RMSProp'. 2021. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.
[33] Weininger D, Weininger A, Weininger JL. SMILES. 2. Algorithm for generation of unique SMILES notation. J Chem Inf Comput Sci 1989;29(2):97–101. https://doi.org/10.1021/ci00062a008. May.
[34] J.T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, 'Striving for simplicity: the all convolutional net', *ArXiv14126806 Cs*, Apr. 2015, Accessed: Mar. 31, 2022. [Online]. Available: http://arxiv.org/abs/1412.6806.
[35] S. Xu, S. Venugopalan, and M. Sundararajan, 'Attribution in scale and space', *ArXiv200403383 Cs*, Apr. 2020, Accessed: Mar. 31, 2022. [Online]. Available: http://arxiv.org/abs/2004.03383.
[36] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry, 'XRAI: better attributions through regions', *ArXiv190602825 Cs Stat*, Aug. 2019, Accessed: Mar. 31, 2022. [Online]. Available: http://arxiv.org/abs/1906.02825.
[37] 'saliency: framework-agnostic saliency methods'. Accessed: Apr. 28, 2022. [Online]. Available: https://github.com/pair-code/saliency.
[38] Schrödinger, L.L.C., 'The PyMOL molecular graphics system, version 2.3.0'.
[39] C. Dens, 'McFAE: molecular complex feature attribution extraction'. Apr. 2022. [Online]. Available: https://github.com/PigeonMark/McFAE.
[40] Dens C, Bittremieux W, Affaticati F, Laukens K, Meysman P. Interpretable deep learning to uncover the molecular binding patterns determining TCR–epitope interactions, 28. Zenodo; 2022. https://doi.org/10.5281/zenodo.7115860. Apr.
[41] Van Rossum G, Drake FL. Python 3 reference manual. Scotts Valley, CA: CreateSpace; 2009.
[42] A. Paszke et al., 'Automatic differentiation in PyTorch', 2017.
[43] Martín Abadi et al., 'TensorFlow: large-scale machine learning on heterogeneous systems'. 2015. [Online]. Available: https://www.tensorflow.org/.
[44] F. Chollet and others, 'Keras'. 2015. [Online]. Available: https://keras.io.
[45] Harris CR, et al. Array programming with NumPy. Nature Sep. 2020;585(7825):357–62. https://doi.org/10.1038/s41586-020-2649-2.
[46] pandas-dev/pandas: pandas. Zenodo; 2020. https://doi.org/10.5281/zenodo.3509134. Feb.
[47] McKinney W. Data structures for statistical computing in Python. In: van der Walt S, Millman J, editors. Proceedings of the 9th Python in science conference; 2010. p. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a.
[48] Hunter JD. Matplotlib: a 2D graphics environment. Comput Sci Eng 2007;9(3):90–5. https://doi.org/10.1109/MCSE.2007.55.
[49] Waskom ML. seaborn: statistical data visualization. J Open Source Softw 2021;6(60):3021. https://doi.org/10.21105/joss.03021.
[50] A. Clark, 'Pillow (PIL Fork) documentation. Readthedocs, 2015. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.