Interpretable deep learning to uncover the molecular binding patterns determining TCR–epitope interactions

Ceder Dens[1], **Wout Bittremieux**[2], **Fabio Affaticati**[1], **Kris Laukens**[1], **Pieter Meysman**[1]

[1]Adrem Data Lab, Department of Computer Science, University of Antwerp, 2020 Antwerpen, Belgium
[2]Dorrestein Laboratory, University of California San Diego, Skaggs School of Pharmacy and Pharmaceutical Sciences, La Jolla, CA 92093, USA

Abstract

Background

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 with novel epitopes, because the underlying patterns that drive the recognition are still largely unknown to both domain experts and machine learning models.

Results

The binding of a TCR and epitope sequence can only occur when amino acids from both sequences are in close contact with each other. We analyze the distance between interacting molecules of the TCR and epitope sequences and compare this to the amino acids that are important for TCR–epitope prediction models. Important residues are determined by using interpretable deep learning techniques or, more specifically, feature attribution extraction methods, on two state-of-the-art TCR–epitope prediction models: ImRex and TITAN. Highlighting feature attributions on the molecular complex reveals additional insights to the domain expert about why the prediction was made and can offer novel insights into the factors that determine TCR affinity on a molecular level. We also show which residues of the TCR and epitope sequences determine binding prediction for ImRex and TITAN and use those to explain model performance.

Conclusions

Extracting feature attributions is a useful way to verify your model and data for challenging problems where small hard-to-detect problems can accumulate to inaccurate results.

Keywords: T-cell epitope prediction, interpretable deep learning, immunoinformatics

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 has to recognize the epitope to be activated and to 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 would lead to many advances in healthcare, by aiding diagnostics, vaccine development, and cancer therapies. Although some machine learning tools that perform these predictions exist, their performance is relatively low. 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 (4–14). However, it is still unclear which underlying patterns or features of the CDR3 and epitope sequences lead to binding, even for domain experts. As a consequence, high-quality machine learning models that are able to learn relevant patterns of 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 learned 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 but patterns learned by those models are more interesting. 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 (15). When extracting feature attributions from neural networks, we can divide the extraction methods in two classes.

The first class consists of gradient-based methods, which make use of the internal weights of the neural network. Vanilla (16) is the oldest and simplest gradient-based method. It gives each input feature an attribution by determining how much the predicted output changes with a minor variation in the input by calculating the gradients of the neural network weights for the given input sample. Vanilla gradients are known to suffer from saturation, which is mainly caused by the use of the ReLU activation function in common neural network architectures (15,17). The ReLU function returns the input activation when it is above 0 and otherwise returns 0. In the latter case, the gradient of that neuron will also be 0. This can often be solved by using Integrated Gradients (IG) (18) instead. 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. The inputs closer to the baseline will suffer less from saturation. The third gradient-based method we applied, XRAI (19), is specifically developed for image-like input. XRAI first divides the input image in segments with Felzenszwalb's algorithm (20). This should capture perceptually important groupings or regions, which often reflect global aspects of the image. Afterwards, feature attributions are extracted with IG using a black and white baseline image and finally, each segment gets a feature attribution value.

The second class of feature attribution extraction methods are 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 are replaced by their baseline value. Multiple options are available for the baseline. The first 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. A second option is to use the average input feature values of (a subset of) the dataset. In this case, the same baseline is used for all samples.

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 the BLOSUM62 matrix 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.

Results

Molecular distances underlie recognition patterns in TCR–epitope complexes

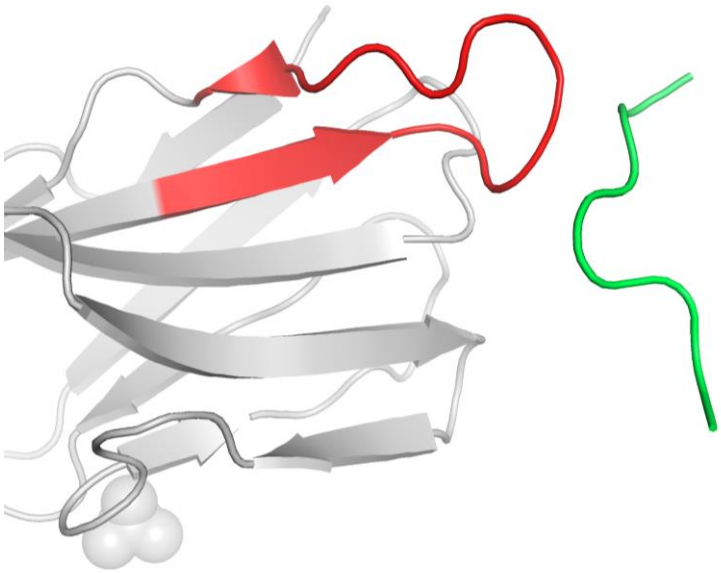

Figure 1. Molecular complex of a TCR–epitope interaction. The 3D image of the PDB complex 2P5W (22,23). Only the TCR beta chain and the epitope are shown, with the CDR3 region of the TCR beta chain colored red and the epitope colored green.

While the complete TCR 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 Figure 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 binding between two protein sequences can only occur when there is close contact between amino acids of both sequences. The distance between a pair of amino acids from both sequences, therefore, gives a good indication

of how important that pair of amino acids is for the binding. 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.

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% ($ROC\ AUC_{ImRex} = 0.55 \pm 0.027$; $ROC\ AUC_{TITAN} = 0.559 \pm 0.05$; $PR\ AUC_{ImRex} = 0.556 \pm 0.033$; $PR\ AUC_{TITAN} = 0.541 \pm 0.049$), which shows that TCR–epitope binding prediction is still a very difficult task, even for state-of-the-art machine learning models. Models that are able to learn which amino acids of the TCR and epitope sequences are most important for the binding could be better at capturing the relevant underlying patterns and as a result, perform better. Comparing the amino acid usage of the prediction tools with the actual distance between the amino acids may help to evaluate and to improve the performance and robustness of those 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 the ImRex model (Figure 2).

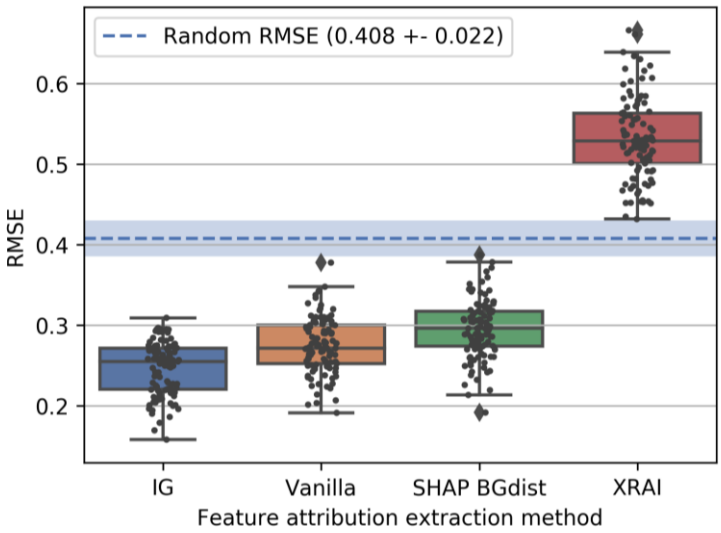

Figure 2. RMSE of feature attribution extraction methods applied to ImRex. The root-mean-square error (RMSE) is calculated between the feature attributions extracted with a specific method and the 3D distance between the amino acids. A boxplot is shown for each method giving the RMSE over all 105 complexes. The random RMSE is calculated by taking the RMSE between a random feature attribution matrix and the actual 3D distance for each sample, repeated multiple times. Thus, this represents the RMSE 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 RMSE is given as mean and standard deviation.

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 each sample, we computed the root-mean-square error (RMSE) between the feature attributions and the pairwise 3D distance between the amino acids. The 3D distance is inverted (by replacing its values by 1/value) because we expect an amino acid pair with a small distance to have a higher feature attribution. Afterwards, both the feature attribution and 3D distance matrices are normalized. Figure 2 shows the RMSE of each sample extracted with 4 different feature attribution extraction methods: Integrated Gradients (IG), Vanilla, SHAP using a background distribution (SHAP BGdist), and XRAI. The IG method has a slightly lower error than the Vanilla method ($RMSE_{IG} = 0.247 \pm 0.032$; $RMSE_{Vanilla} = 0.275 \pm 0.034$), which can be explained by the saturation problem of Vanilla. While being a model-agnostic method, SHAP still performs well ($RMSE_{SHAP\ BGdist} = 0.295 \pm 0.036$). XRAI performs worse than random on all our samples ($RMSE_{XRAI} = 0.532 \pm 0.05$; $RMSE_{random} = 0.408 \pm 0.022$). This can be explained by the segmentation step used by this algorithm, which is a good method for real-life images because regions of similar pixels often correspond to objects that might be important for the prediction. However, this assumption is often not valid for TCR–epitope binding. Adjacent amino acids can have very different properties and still be important. An overview of the RMSE of all 10 feature extraction methods we tested can be seen in Figure S2, a detailed feature attribution matrix for a single sample can be seen in Figure S3.

Feature attributions reveal important residues for each prediction

The remainder of all experiments were run using the SHAP feature attribution extraction method on both ImRex and TITAN. This method showed comparable performance to the best scoring approach and is the only method applicable to the categorical input of the TITAN model.

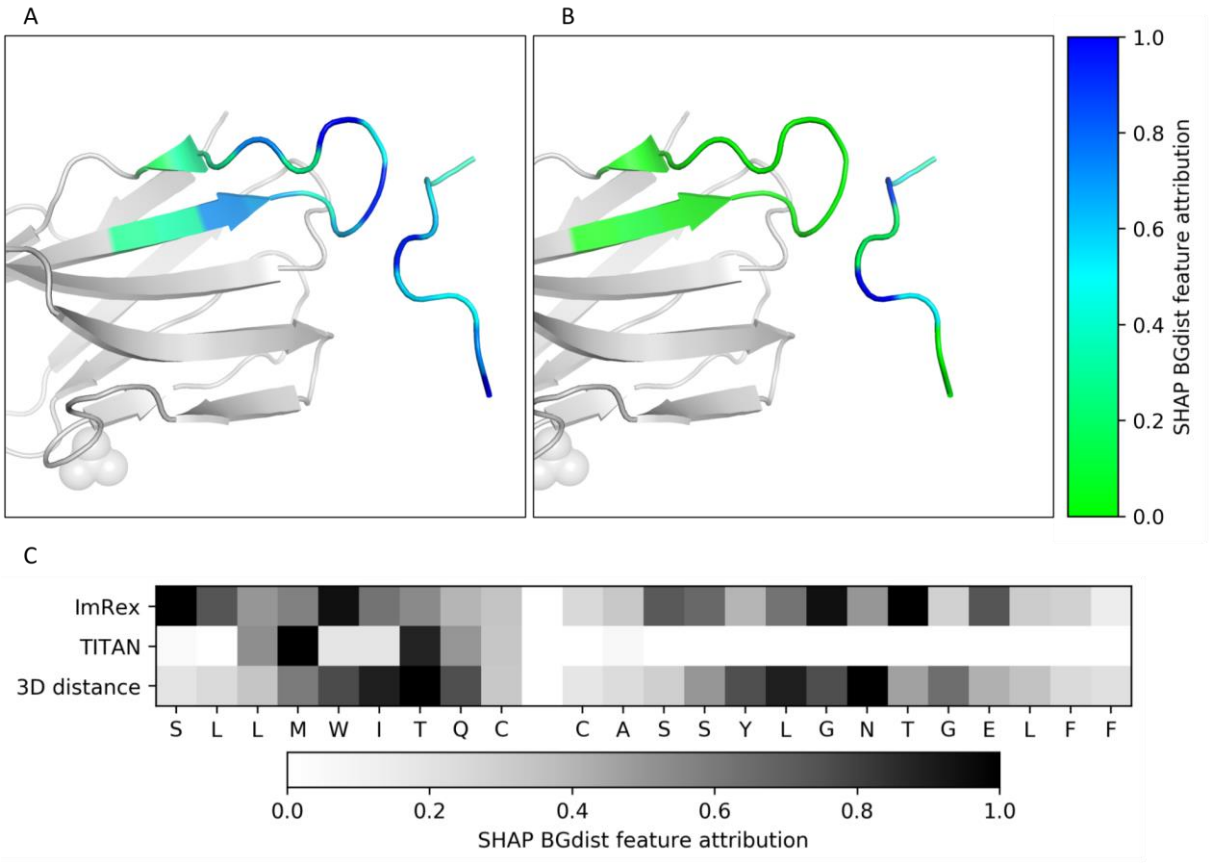

Figure 3. Feature attributions extracted with SHAP from PDB complex 2P5W. Feature attributions of (A) ImRex and (B) TITAN extracted with SHAP from the prediction for the PDB complex 2P5W (22,23) and shown on its molecular complex. Only the TCR beta chain and the epitope are shown. The TCR beta chain 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 3D distance. For TITAN and ImRex, a higher value represents a larger feature attribution, for the 3D distance, a higher value represents a smaller distance.

The pairwise feature attributions from ImRex were merged per amino acid. This was necessary for the visualizations and the comparison to TITAN, as TITAN only uses the amino acid sequences as input, which results in one feature attribution value per amino acid. The pairwise feature attributions were merged by using the maximum of all feature attribution values for that 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 3D distance 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 to the amino acids from the other sequence is taken. Afterwards, all values from this distance array are inverted by replacing them by 1/value to make them comparable to the feature attributions, a higher value for the 3D distance means that the amino acid is closer to the other sequence and a higher feature attribution is expected. At 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 for the PDB complex 2P5W (22). For the epitope, we can see that TITAN uses only some of the amino acids, while the attributions of ImRex are more evenly distributed. The amino acids that are most important for TITAN are also in close contact with the CDR3 region, which is not the case for all important amino acids according to ImRex. 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.

Important residues are distinct for each TCR-epitope model

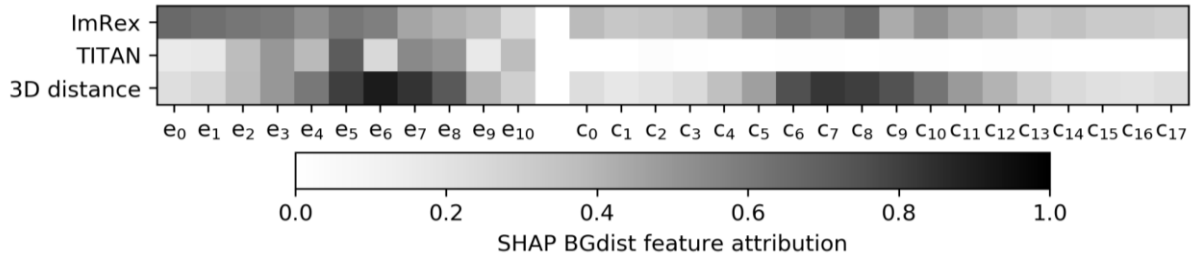

Figure 4. Average feature attributions per position. The average feature attribution per position and model, also the average 3D distance is given. Both the epitope and CDR3 sequence are padded left and right separately. The average is calculated by only looking at the feature attributions from sequences that do not have padding on that position. A higher value represents a higher feature attribution for ImRex and TITAN and a smaller distance for the 3D distance.

The findings derived from a single complex are representative for the full dataset, as can be seen in figure 4. We calculated the average attributions of the 105 samples and the average 3D distance. This shows that the amino acids of the epitope on position 6 to 9 (e_5 - e_8) are on average the closest to the CDR3 region. ImRex focuses more on the first part of the epitope and the attributions are in general evenly distributed. TITAN does not really consider the first positions but focuses more on the middle part and the average attributions per position vary more. For the CDR3, the attributions from ImRex are similar to the 3D distance but more evenly spread across the full sequence. 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.

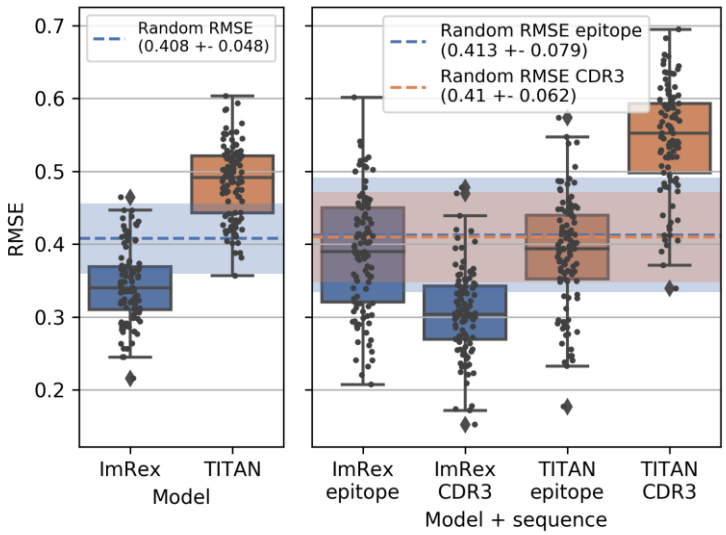

Figure 5. RMSE between attributions and 3D distance for models and sequences. The RMSE is calculated between the feature attributions extracted from different models with SHAP and the 3D distance between amino acid pairs from both sequences. For ImRex, the attributions were first merged per amino acid to allow comparison with TITAN. On the right, the RMSEs for both the epitope and CDR3 sequences separately are shown. The random RMSE is calculated by taking the RMSE between a random feature attribution array and the actual 3D distance for each sample, repeated multiple times. Thus, this represents the RMSE when the feature

attribution extraction method would give a random output. Note that these results for ImRex are different from those in figure 2 because the feature attributions and 3D distance are now first merged per amino acid. 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 RMSE is given as mean and standard deviation.

We calculated the RMSE between the feature attributions extracted from ImRex and TITAN and the distance between the amino acids of the CDR3 and epitope sequence. The attributions of ImRex and the distances were first merged per amino acid. This results in a lower RMSE for ImRex when looking at both sequences together ($RMSE_{ImRex\ AA} = 0.344 \pm 0.051$; $RMSE_{TITAN} = 0.485 \pm 0.051$) (Figure 5). When only considering the epitope, the RMSE is similar for both models ($RMSE_{ImRex\ AA\ ep} = 0.387 \pm 0.081$; $RMSE_{TITAN\ ep} = 0.387 \pm 0.075$) and not clearly better than random ($RMSE_{random\ AA\ ep} = 0.413 \pm 0.079$). On the other hand, the CDR3 sequence RMSE is much better for ImRex than for TITAN ($RMSE_{ImRex\ AA\ CDR3} = 0.306 \pm 0.06$; $RMSE_{TITAN\ CDR3} = 0.543 \pm 0.072$). This can be explained by the fact that TITAN always gives a near-zero attribution to the entire CDR3 sequence.

Feature attributions explain model performance

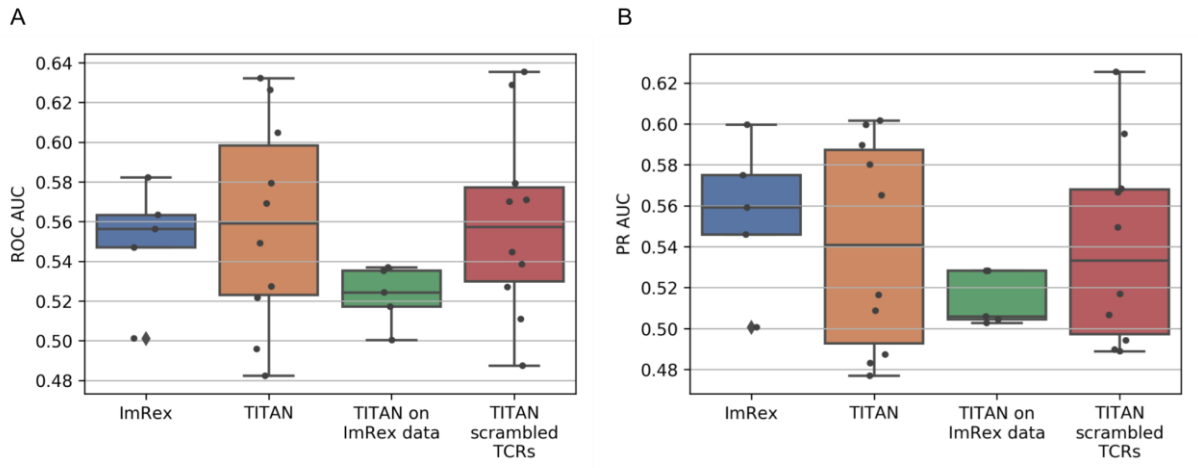

Figure 6. Comparison of model performance. The performance of the different models measured with (A) ROC AUC and (B) PR AUC. 'ImRex' and 'TITAN on ImRex data' were both trained on the ImRex dataset and evaluated with 5-fold epitope-grouped cross-validation. 'TITAN' was trained on the original TITAN dataset 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.

We re-evaluated both the ImRex and TITAN models using epitope-grouped cross-validation. 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 (Figure 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 a 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 figure 6, also see figures S4 and S5 for average feature attributions and RMSE 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 due to which 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.

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). 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 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 in the factors that determine TCR affinity on a molecular level.

Using the actual distance between the amino acids of the TCR and epitope sequences as ground truth, we were able to compare the performance of different feature attribution extraction methods and prediction models. We found that IG is the best method for ImRex. This is not unexpected, because it suffers less from saturation compared to Vanilla and it is not specifically designed for image classification neural networks, like XRAI. The poor performance of XRAI shows the importance of understanding how an interpretability method works and which use cases it is designed for before applying it to biomedical research questions. Additionally, it is encouraging that the model-agnostic method SHAP does not perform significantly worse than the best gradient-based methods, because it can be easily applied to any model (although it has substantially longer runtimes).

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 position 6 to 9 (out of 11). This points to a first limitation of our study, as we only considered the distance between the TCR and epitope. However, one must also consider the interaction between the epitope and the MHC molecule, which is known to occur close to either end of the epitope for MHC class I. ImRex is revealed to use the amino acids of the CDR3 region more equally with somewhat more attribution to the middle part, this is similar to the average amino acid distance. The amino acid usage of the epitope sequence by TITAN is similar to the 3D distance apart from position 6; although this position is closest to the CDR3 on average, surprisingly it gets a much lower attribution from TITAN.

By extracting feature attributions from TITAN, we found that it only looks at the epitope and 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 in 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. We therefore 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). At 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. Afterwards, the performance could have been given by applying the model from that epoch on the test set.

Conclusions

We showed 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 model-agnostic method SHAP works almost as well as IG but has the advantage that it is easily applicable on any type of model and is not dependent on the input type. Showing feature attributions on their 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.

Methods

Data

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. For each of these PDB complexes, a set of additional data was manually assembled with the help of IMGT/3Dstructure-DB (25–27). The final set of PDB complexes all consist of 5 chains: the MHC class I alpha, the MHC class I beta$_2$-microglobulin, the TCR alpha, the TCR beta, and the epitope chain. The naming and numbering of the TCR beta and epitope chain is not consistent across all complexes so we manually selected the correct ones with the help of the IMGT/3Dstructure-DB online tool. The location of the CDR3 region in the TCR beta sequence was also selected manually using the same tool. Finally, we ended up with 105 unique complexes.

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 10-20 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 and epitope amino acid sequences) were removed, which reduced the positive dataset size from 6702 to 6656 samples. 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 in 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. Every cross-validation iteration, the model is tested on one of the five groups and trained on the other four.

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). The VDJdb database was 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. The COVID-19 dataset was also filtered on TCR beta sequences and only samples with a single unique epitope were kept. Unproductive samples were excluded, epitopes with less than 15 TCRs were again removed and the data was also downsampled to 400 samples per epitope. 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 when generating the random sequences.

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.

ImRex model training

We retrained ImRex with the same parameters as the final published model.[1] 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 (31).

TITAN model training

TITAN was always trained with the parameter configuration of the AA CDR3 case as explained in their paper (a different configuration was given on their online repository but this configuration led to similar results). We set the padding of the CDR3 and epitope to 25 instead of 500, because otherwise we could not achieve a better than random performance. The epitope is not encoded as SMILES (32), the size of the dense hidden layers is 368 and

[1] https://github.com/pmoris/ImRex/tree/master/models/pretrained/2020-07-24_19-18-39_trbmhcidown-shuffle-padded-b32-lre4-reg001

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.

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. This means that both models were trained and evaluated on exactly the same samples.

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 (16), Integrated Gradients (IG) (18), SmoothGrad (33), SmoothGradIG (18,33), GuidedIG (18,34), BlurIG (18,35), SmoothGradBlurIG (18,33,35) and XRAI (19)) and 2 variations of SHAP (SHAP with the average dataset as background and SHAP with random sampling from the dataset as background (denoted as SHAP BGdist)). Of these, for conciseness, four representative methods were used in the results section (IG, Vanilla, SHAP BGdist and XRAI), while results for all methods are available as supplementary information. We implemented the IG method ourselves, the Python package *shap* (version 0.40.0) (21) was used for the implementation of the different SHAP methods, and the other feature attribution extraction methods were implemented with the Python package *saliency* (version 0.1.3) (16,19,33–36).

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. The categorical input makes that only 1 of the 9 methods can be used on TITAN: SHAP using a background distribution dataset. Gradient based methods can not be used because they require models with a differentiable input and the inner layers are not accessible for feature attribution extraction. SHAP using the average input as background data can also not be used because categorical inputs can not be averaged.

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 calculated by taking the minimal distance between two atoms of each amino acid in ångströms. This results in a distance matrix with a value for each combination of amino acids from the CDR3 and epitope sequence between 2.3Å for the closest amino acids and 30Å for the most distant amino acids. Our reasoning is that 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. The feature attributions and the inverted distance 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.

$$inverted\_distance_{ij} = inv\_d_{ij} = \frac{1}{d_{ij}}, i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

$$normalized\_distance_{ij} = nd_{ij} = \frac{inv\_d_{ij}}{\max\limits_{k\in\{1,\dots,m\},l\in\{1,\dots,n\}} inv\_d_{kl}}, i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

$$normalized\_feature\_attribution_{ij} = nfa_{ij} = \frac{fa_{ij}}{\max\limits_{k\in\{1,\dots,m\},l\in\{1,\dots,n\}} fa_{kl}}, i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$$

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).

For each sample the root-mean-square error (RMSE) is calculated by looking at the difference between the normalized feature attribution and the normalized distance for each amino acid combination.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(nd_{ij} - nfa_{ij}\right)^2}{m \cdot n}}$$

The RMSE of a sample for a prediction model and extraction method represents how close the actual amino acid distance is related to the input features used by the model to make the prediction.

1D feature attributions

To compare the feature attributions and the RMSE of the feature attributions 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 $n + m$) were inverted and normalized in the same way as the 2D matrices. The RMSE was also calculated in the same way as it was on the 2D data.

Molecular complex highlighting

The normalized 1D feature attributions from both models are shown on the molecular complex with PyMol (version 2.3.0) (37). 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.

Declarations

Availability of data and materials

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 (38) and on Zenodo at https://doi.org/10.5281/zenodo.6500496 (39). All code is written in Python 3.7 (40). PyTorch (version 1.10.0) (41), TensorFlow (GPU version 2.6.0) (42) and Keras (version 2.6.0) (43) were used for implementing the neural network architectures and model training. NumPy (version 1.18.1) (44) and pandas (version 1.0.1) (45,46) were

used for data processing. SHAP (version 0.40.0) (21) was used to extract SHAP feature attributions and Saliency (version 0.1.3) (36) was used to extract the other feature attributions. Matplotlib (version 3.1.3) (47), seaborn (version 0.10.0) (48), and Pillow (version 7.0.0) (49) were used to create the figures.

Competing interests

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

Funding

This research received funding from 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.

Authors' contributions

CD performed the study and wrote the manuscript. FA performed a preliminary study on applying integrated gradients on ImRex. WB, KL and PM conceived and supervised the study and revised the manuscript. All authors read and approved the final manuscript.

References

1. Alberts B. Molecular biology of the cell. Sixth edition. New York, NY: Garland Science, Taylor and Francis Group; 2015.
2. Krogsgaard M, Davis MM. How T cells 'see' antigen. Nat Immunol. 2005 Mar;6(3):239–45.
3. Davis MM, Bjorkman PJ. T-cell antigen receptor genes and T-cell recognition. Nature. 1988 Aug;334(6181):395–402.
4. Moris P, De Pauw J, Postovskaya A, Gielis S, De Neuter N, Bittremieux W, et al. Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. Brief Bioinform. 2021 Jul 1;22(4):bbaa318.
5. Weber A, Born J, Rodriguez Martínez M. TITAN: T-cell receptor specificity prediction with bimodal attention networks. Bioinformatics. 2021 Jul 1;37(Supplement_1):i237–44.
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.
7. De Neuter N, Bittremieux W, Beirnaert C, Cuypers B, Mrzic A, Moris P, et al. On the feasibility of mining CD8+ T cell receptor patterns underlying immunogenic peptide recognition. Immunogenetics. 2018 Mar 1;70(3):159–68.
8. Dash P, Fiore-Gartland AJ, Hertz T, Wang GC, Sharma S, Souquette A, et al. Quantifiable predictive features define epitope-specific T cell receptor repertoires. Nature. 2017 Jul;547(7661):89–93.
9. Glanville J, Huang H, Nau A, Hatton O, Wagar LE, Rubelt F, et al. Identifying specificity groups in the T cell receptor repertoire. Nature. 2017 Jul;547(7661):94–8.
10. Jokinen E, Huuhtanen J, Mustjoki S, Heinonen M, Lähdesmäki H. Determining epitope specificity of T cell receptors with TCRGP [Internet]. bioRxiv; 2019 [cited 2022 Apr 15]. Available from: https://www.biorxiv.org/content/10.1101/542332v2
11. Gielis S, Moris P, Bittremieux W, De Neuter N, Ogunjimi B, Laukens K, et al. Detection of Enriched T Cell Epitope Specificity in Full T Cell Receptor Sequence Repertoires. Front Immunol. 2019;10:2820.

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 [cited 2022 Apr 15];11. Available from: https://www.frontiersin.org/article/10.3389/fimmu.2020.01803
13. Chronister WD, Crinklaw A, Mahajan S, Vita R, Koşaloğlu-Yalçın Z, Yan Z, et al. TCRMatch: Predicting T-Cell Receptor Specificity Based on Sequence Similarity to Previously Characterized Receptors. Front Immunol. 2021 [cited 2022 Apr 15];12. Available from: https://www.frontiersin.org/article/10.3389/fimmu.2021.640725
14. Montemurro A, Schuster V, Povlsen HR, Bentzen AK, Jurtz V, Chronister WD, et al. NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. Commun Biol. 2021 Sep 10;4(1):1–13.
15. Molnar C. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. 2nd ed. 2022. Available from: https://christophm.github.io/interpretable-ml-book
16. Simonyan K, Vedaldi A, Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. ArXiv13126034 Cs. 2014 Apr 19 [cited 2022 Mar 31]; Available from: http://arxiv.org/abs/1312.6034
17. Shrikumar A, Greenside P, Kundaje A. Learning Important Features Through Propagating Activation Differences. ArXiv170402685 Cs. 2019 Oct 12 [cited 2022 Apr 14]; Available from: http://arxiv.org/abs/1704.02685
18. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. Sydney, NSW, Australia: JMLR.org; 2017. p. 3319–28. (ICML'17).
19. Kapishnikov A, Bolukbasi T, Viégas F, Terry M. XRAI: Better Attributions Through Regions. ArXiv190602825 Cs Stat. 2019 Aug 20 [cited 2022 Mar 31]; Available from: http://arxiv.org/abs/1906.02825
20. Felzenszwalb PF, Huttenlocher DP. Efficient Graph-Based Image Segmentation. Int J Comput Vis. 2004 Sep;59(2):167–81.
21. Lundberg SM, Lee SI. A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2017 [cited 2022 Mar 30]. Available from: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html
22. Sami M, Rizkallah PJ, Dunn S, Molloy P, Moysey R, Vuidepot A, 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 Aug;20(8):397–403.
23. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res. 2000 Jan 1;28(1):235–42.
24. Gowthaman R, Pierce BG. TCR3d: The T cell receptor structural repertoire database. Bioinforma Oxf Engl. 2019 Dec 15;35(24):5323–5.
25. Ehrenmann F, Lefranc MP. 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 Jun 1;2011(6):pdb.prot5637.
26. Ehrenmann F, Kaas Q, Lefranc MP. 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 Jan 1;38(suppl_1):D301–7.
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 Jan 1;32(suppl_1):D208–10.
28. Bagaev DV, Vroomans RMA, Samir J, Stervbo U, Rius C, Dolton G, et al. VDJdb in 2019: database extension, new analysis infrastructure and a T-cell receptor motif compendium. Nucleic Acids Res. 2020 Jan 8;48(D1):D1057–62.
29. 10x Genomics: A New Way of Exploring Immunity Digital. [cited 2022 Mar 30]. Available from: https://pages.10xgenomics.com/rs/446-PBO-

704/images/10x_AN047_IP_A_New_Way_of_Exploring_Immunity_Digital.pdf

30. Dines JN, Manley TJ, Svejnoha E, Simmons HM, Taniguchi R, Klinger M, et al. The ImmuneRACE Study: A Prospective Multicohort Study of Immune Response Action to COVID-19 Events with the ImmuneCODE™ Open Access Database. medRxiv; 2020 [cited 2022 Mar 30]. p. 2020.08.17.20175158. Available from: https://www.medrxiv.org/content/10.1101/2020.08.17.20175158v2
31. Hinton G. RMSProp. 2021. Available from: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
32. Weininger D, Weininger A, Weininger JL. SMILES. 2. Algorithm for generation of unique SMILES notation. J Chem Inf Comput Sci. 1989 May 1;29(2):97–101.
33. Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M. SmoothGrad: removing noise by adding noise. ArXiv170603825 Cs Stat. 2017 Jun 12 [cited 2022 Mar 31]; Available from: http://arxiv.org/abs/1706.03825
34. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving for Simplicity: The All Convolutional Net. ArXiv14126806 Cs. 2015 Apr 13 [cited 2022 Mar 31]; Available from: http://arxiv.org/abs/1412.6806
35. Xu S, Venugopalan S, Sundararajan M. Attribution in Scale and Space. ArXiv200403383 Cs. 2020 Apr 8 [cited 2022 Mar 31]; Available from: http://arxiv.org/abs/2004.03383
36. saliency: Framework-agnostic saliency methods. [cited 2022 Apr 28]. Available from: https://github.com/pair-code/saliency
37. Schrödinger, LLC. The PyMOL Molecular Graphics System, Version 2.3.0.
38. Dens C. McFAE: Molecular Complex Feature Attribution Extraction. 2022. Available from: https://github.com/PigeonMark/McFAE
39. Ceder D, Wout B, Fabio A, Kris L, Pieter M. Interpretable deep learning to uncover the molecular binding patterns determining TCR–epitope interactions. Zenodo; 2022 [cited 2022 Apr 29]. Available from: https://zenodo.org/record/6500496
40. Van Rossum G, Drake FL. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace; 2009.
41. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic differentiation in PyTorch. 2017;
42. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available from: https://www.tensorflow.org/
43. Chollet F, others. Keras. 2015. Available from: https://keras.io
44. Harris CR, Millman KJ, Walt SJ van der, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020 Sep;585(7825):357–62.
45. pandas-dev/pandas: Pandas. Zenodo; 2020. Available from: https://doi.org/10.5281/zenodo.3509134
46. McKinney W. Data Structures for Statistical Computing in Python. In: Walt S van der, Millman J, editors. Proceedings of the 9th Python in Science Conference. 2010. p. 56–61.
47. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007;9(3):90–5.
48. Waskom ML. seaborn: statistical data visualization. J Open Source Softw. 2021;6(60):3021.
49. Clark A. Pillow (PIL Fork) Documentation. readthedocs; 2015. Available from: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf