BATMAN: Improved T cell receptor cross-reactivity prediction benchmarked on a comprehensive mutational scan database

Amitava Banerjee [1], David J Pattinson [2], Cornelia L. Wincek [1,3], Paul Bunk [4], Sarah R. Chapin [1], Saket Navlakha [1✉], and Hannah V. Meyer [1✉]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, USA
[2]Department of Pathobiological Sciences, School of Veterinary Medicine, University of Wisconsin-Madison, Madison, WI 53711, USA
[3]Heidelberg University, 69117 Heidelberg, Germany
[4]School of Biological Sciences, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, USA
✉To whom correspondence should be addressed. E-mail: hmeyer@cshl.edu, navlakha@cshl.edu

ABSTRACT

Predicting T cell receptor (TCR) activation is challenging due to the lack of both unbiased benchmarking datasets and computational methods that are sensitive to small mutations to a peptide. To address these challenges, we curated a comprehensive database encompassing complete single amino acid mutational assays of 10,750 TCR-peptide pairs, centered around 14 immunogenic peptides against 66 TCRs. We then present an interpretable Bayesian model, called BATMAN, that can predict the set of peptides that activates a TCR. When validated on our database, BATMAN outperforms existing methods by 20% and reveals important biochemical predictors of TCR-peptide interactions.

Introduction

A single TCR can recognize a variety of peptides, a property known as TCR cross-reactivity [1, 2]. Predicting which peptides a TCR cross-reacts to is critical for numerous applications, including predicting viral escape [3], cancer neoantigen immunogenicity [4], autoimmunity [2, 5], and off-target toxicity of T-cell-based therapies [6]. However, predicting interactions among TCRs, peptides, and major histocompatibility complexes (TCR-pMHCs) remains challenging [7, 8] due to: (a) limited TCR cross-reactivity assay data; and (b) few experimentally validated negative examples [9], which are important for model discrimination (Figure 1a). Existing computational methods impressively cluster different TCRs that bind the same peptide [7, 10]. But the opposite task — predicting peptides that bind a given TCR — remains outstanding [8, 11, 12]. This is largely due to the sensitivity required to discriminate among single amino acid (AA) mutants [13] of a TCR's known index peptide, i.e. the peptide to which the TCR was identified to strongly bind. To address this challenge, we offer both a comprehensive experimental mutational scan database of TCR-pMHC binding (Figure 1b), and a method that can predict how peptide mutations affect TCR activation (Figure 1c-d).

Comprehensive database on TCR-specific mutational scans

We curated a database of continuous-valued TCR-pHMC binding data measured as T cell activation in mutational scan assays (hereafter referred to as TCR activation data; Figure 1c). This database includes 66 fully-sequenced $CD8^+$ mouse and human TCR clones (Extended Data Fig 1a), together recognizing 5 class-I MHCs and 14 unique index peptides that are length $L \in [8, 11]$ AA long and involved in cancer, viral infection or autoimmunity. For each TCR, we recorded the activation levels of all possible $L \times 19$ single-AA mutant peptides (Figure 1b-c). This achieved a coverage of the antigenic space unprecedented among existing methods (Figure 1a), and generated high-confidence true positive (TCR activated) and true negative (TCR inactive) examples. Our database showed that single AA changes of the index peptide result in both loss and gain of TCR activation over orders of magnitude (Figure 1b, Extended Data Fig 1b). Furthermore, different TCRs sharing a common index peptide bind with different structures [14, 15], and recognize different mutants of the index peptide [16, 17], demonstrating the need for benchmarking TCR-pMHC prediction methods with diverse index peptides and mutants.

BATMAN: a Bayesian inference model to predict TCR activation by mutant peptides

We present BATMAN — "**B**ayesian Inference of **A**ctivation of **T**CR by **M**utant **An**tigens" — a hierarchical Bayesian model that can predict TCR activation by single-AA mutant peptides based on their distances to the TCR's index peptide (Figure 1d). The peptide-to-index distance is a product of (a) a learned positional weight profile, corresponding to effects of mutated residues at different positions in the sequence, and (b) a learned AA substitution distance from the index AA to the mutant AA (Figure 1d, Methods section). BATMAN does not require an input TCR sequence and can be used for classification and continuous regression tasks for both TCR-specific and cross-TCR activation datasets.

We benchmarked BATMAN over a diverse subset of TCRs from our database — consisting of 1,884 TCR-pMHC pairs, spanning 11 human TCRs specific for unique 9-AA-long index peptides. For ease of interpretation, we discretized continuous TCR activation values into three levels: strong, weak or no activation. For multiple TCRs with the same index peptide in our database, we chose the one with the least class imbalance to construct the benchmarking dataset (Extended Data Figures have results for both classification and continuous regression tasks using activation values for all TCRs).

We validated BATMAN in two modes:

1. *Within-TCR* mode, where the train and test peptides were associated with the same TCR, and positional weight profiles were TCR-specific. We first used conventional AA substitution distance matrices, and performed 5-fold cross validation separately for each TCR, using about 144 random peptides from the set of single-AA mutants for the TCR for training, and the remaining 36 for testing. Then, we combined activation data from all TCRs to learn a TCR-independent AA distance matrix.

2. *Leave-one-TCR-out* mode, where peptides were tested for activation of a TCR left out of the training data, and positional weight profile was common across all TCRs. Here, we combined TCR activation data to infer both the AA distance matrix and a single positional weight profile across TCRs.

We compared BATMAN's performance to a host of other machine learning-based methods designed to predict TCR-pMHC interactions, including *pTEAM* [13], which, to our knowledge, is the only existing method dedicated for predicting peptide mutation effects on TCR activation.

BATMAN outperforms existing TCR-pMHC methods and learns TCR-pMHC biochemistry

BATMAN outperformed all other methods in both within-TCR (mean AUC=0.80 over next best method *pTEAM* at AUC=0.71) and leave-one-TCR-out (mean AUC=0.69 over next best method *pTEAM* at AUC=0.59) classification (Figure 2a). Previously developed neural network models trained on large publicly available databases (e.g., VDJdb [18], McPAS-TCR [19], and single-cell immune repertoire profiling data [20]) that excel at predicting different TCRs that bind a given peptide [7, 10], predicted only marginally better than random. Predicted TCR-pMHC interaction likelihood scores from these models were uncorrelated with true TCR activation values for the mutant peptides (Extended Data Fig 5).

Critical to achieving BATMAN's performance was learning TCR-pMHC-specific AA distance matrices by pooling training data across TCRs (Methods). For example, applying BATMAN with the conventionally-used [21] Hamming distance dropped the within-TCR AUC to 0.74 (Figure 2b). Extended Data Figs 2 to 4 further highlight the superior performance of BATMAN over previous methods when tested on all 66 TCRs using 70 different AA matrices, of which BLOSUM100 performed the best (within-TCR AUC=0.785, Figure 2b, Extended Data Fig 2a,b).

BATMAN's learned positional weight profiles (Figure 2c, Extended Data Fig 6) and AA distance matrix (Figure 2d,Extended Data Fig 7b-e) recapitulated three known biochemical features of TCR-pMHC interactions: (1) Positional weights peak near the middle of the peptide chain, reflecting the fact that central AA residues more directly affect TCR binding compared to anchor residues [17, 21–25] (see also Extended Data Fig 1b), (2) large changes in TCR activation correspond to non-aromatic to aromatic AA substitutions (e.g., valine-phenylalanine) affecting side-chain interactions [22, 26, 27], and (3) swapping in hydrophobic isoleucine and leucine residues for non-hydrophobic residues overall increases TCR activation, in line with these residues considered to increase immunogenicity [26–28]. BATMAN positional weight profiles were also consistent across different AA matrices and between classification and continuous regression tasks (Extended Data Fig 6), indicating that they indeed correspond to learned TCR-intrinsic features.

Discussion

We curated the largest database to date of experimentally validated TCR-pMHC interactions containing all single AA peptide mutations with positive and negative examples (Figure 1a). BATMAN fills a hitherto unoccupied niche of TCR-pMHC prediction methods by discriminating between small differences in peptide sequences for TCR activation. While existing methods seem to learn large-scale TCR activation properties across the antigenic space, they fail to predict single AA mutational

effects, which are essential for predicting neoantigen immunogenicity and TCR targets. This demands more high-throughput, high-confidence experiments [29] generating positive and negative TCR-pMHC interactions by similar peptides, as well as training TCR-pMHC methods by datasets like ours.

BATMAN could be further improved by: (a) incorporating TCR sequence information, (b) training on datasets from other types of experimental TCR cross-reactivity assays [29] (e.g., yeast display library enrichment [5, 30], T-Scan [31], and SABR [32]), which sample outside the one AA-mutational scan space, and (c) extending its predictions to MHC class-II restricted peptides [21, 33, 34].

Author contributions

AB, DJP, SN and HVM conceptualized the work; AB developed the software; AB and DJP designed the model; AB and CW implemented the user interface with help from SRC; AB and PB curated the database; AB conducted all formal analyses; AB, SN, and HVM wrote the original draft; all authors reviewed and edited the final draft; SN and HVM supervised the work.

Competing interests

The authors declare no competing interests.

Acknowledgments

We thank our lab members for discussion and feedback on method development and figure design, Paul G. Thomas and Zachary Sethna for discussion on TCR datasets, Anastasia Troshina and Vasilisa A. Kovaleva for designing the BATMAN logo, and all authors who shared their datasets with us. TCR-pMHC schematics in Figure 1c,d were created with BioRender (https://www.biorender.com/).

Funding

The research was supported by the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory; US National Institutes of Health Grants S10OD028632-01. This work was discussed in part at the Aspen Center for Physics, which is supported by National Science Foundation grant PHY-2210452. The funders had no role in the template design or decision to publish.

References

[1] A. K. Sewell, "Why must t cells be cross-reactive?", Nature Reviews Immunology **12**, 669–677 (2012).

[2] L. Wooldridge, J. Ekeruche-Makinde, H. a. Van Den Berg, A. Skowera, J. J. Miles, M. P. Tan, G. Dolton, M. Clement, S. Llewellyn-Lacey, D. A. Price, et al., "A single autoimmune t cell receptor recognizes more than a million different peptides", Journal of Biological Chemistry **287**, 1168–1177 (2012).

[3] R. Keeton, M. B. Tincho, A. Ngomti, R. Baguma, N. Benede, A. Suzuki, K. Khan, S. Cele, M. Bernstein, F. Karim, et al., "T cell responses to sars-cov-2 spike cross-recognize omicron", Nature **603**, 488–492 (2022).

[4] M. Łuksza, Z. M. Sethna, L. A. Rojas, J. Lihm, B. Bravi, Y. Elhanati, K. Soares, M. Amisaki, A. Dobrin, D. Hoyos, et al., "Neoantigen quality predicts immunoediting in survivors of pancreatic cancer", Nature **606**, 389–395 (2022).

[5] X. Yang, L. I. Garner, I. V. Zvyagin, M. A. Paley, E. A. Komech, K. M. Jude, X. Zhao, R. A. Fernandes, L. M. Hassman, G. L. Paley, et al., "Autoimmunity-associated t cell receptors recognize hla-b* 27-bound peptides", Nature **612**, 771–777 (2022).

[6] K. Ishii, J. S. Davies, A. L. Sinkoe, K. A. Nguyen, S. M. Norberg, C. P. McIntosh, T. Kadakia, C. Serna, Z. Rae, M. C. Kelly, et al., "Multi-tiered approach to detect autoimmune cross-reactivity of therapeutic t cell receptors", Science Advances **9**, eadg9845 (2023).

[7] D. Hudson, R. A. Fernandes, M. Basham, G. Ogg, and H. Koohy, "Can we predict t cell specificity with digital biology and machine learning?", Nature Reviews Immunology, 1–11 (2023).

[8] P. Moris, J. De Pauw, A. Postovskaya, S. Gielis, N. De Neuter, W. Bittremieux, B. Ogunjimi, K. Laukens, and P. Meysman, "Current challenges for unseen-epitope tcr interaction prediction and a new perspective derived from image classification", Briefings in Bioinformatics **22**, bbaa318 (2021).

[9] C. Dens, K. Laukens, W. Bittremieux, and P. Meysman, "The pitfalls of negative data bias for the t-cell epitope specificity challenge", Nature Machine Intelligence **5**, 1060–1062 (2023).

[10]H. Koohy, D. Hudson, A. Lubbock, and M. Basham, "A comparison of clustering models for inference of t cell receptor antigen specificity", bioRxiv, 2023–08 (2023).

[11]F. Grazioli, A. Mösch, P. Machart, K. Li, I. Alqassem, T. J. O'Donnell, and M. R. Min, "On tcr binding predictors failing to generalize to unseen peptides", Frontiers in Immunology **13**, 1014256 (2022).

[12]L. Deng, C. Ly, S. Abdollahi, Y. Zhao, I. Prinz, and S. Bonn, "Performance comparison of tcr-pmhc prediction tools reveals a strong data dependency", Frontiers in Immunology **14**, 1128326 (2023).

[13]E. Dorigatti, F. Drost, A. Straub, P. Hilgendorf, K. I. Wagner, B. Bischl, D. Busch, K. Schober, and B. Schubert, "Predicting t cell receptor functionality against mutant epitopes", bioRxiv, 2023–05 (2023).

[14]C. H. Coles, R. M. Mulvaney, S. Malla, A. Walker, K. J. Smith, A. Lloyd, K. L. Lowe, M. L. McCully, R. Martinez Hague, M. Aleksic, et al., "Tcrs with distinct specificity profiles use different binding modes to engage an identical peptide–hla complex", The Journal of Immunology **204**, 1943–1953 (2020).

[15]I. Song, A. Gil, R. Mishra, D. Ghersi, L. K. Selin, and L. J. Stern, "Broad tcr repertoire and diverse structural solutions for recognition of an immunodominant cd8+ t cell epitope", Nature structural & molecular biology **24**, 395–406 (2017).

[16]A. Straub, S. Grassmann, S. Jarosch, L. Richter, P. Hilgendorf, M. Hammel, K. I. Wagner, V. R. Buchholz, K. Schober, and D. H. Busch, "Recruitment of epitope-specific t cell clones with a low-avidity threshold supports efficacy against mutational escape upon re-infection", Immunity (2023).

[17]A. K. Bentzen, L. Such, K. K. Jensen, A. M. Marquard, L. E. Jessen, N. J. Miller, C. D. Church, R. Lyngaa, D. M. Koelle, J. C. Becker, et al., "T cell receptor fingerprinting enables in-depth characterization of the interactions governing recognition of peptide–mhc complexes", Nature biotechnology **36**, 1191–1196 (2018).

[18]D. V. Bagaev, R. M. Vroomans, J. Samir, U. Stervbo, C. Rius, G. Dolton, A. Greenshields-Watson, M. Attaf, E. S. Egorov, I. V. Zvyagin, et al., "Vdjdb in 2019: database extension, new analysis infrastructure and a t-cell receptor motif compendium", Nucleic Acids Research **48**, D1057–D1062 (2020).

[19]N. Tickotsky, T. Sagiv, J. Prilusky, E. Shifrut, and N. Friedman, "Mcpas-tcr: a manually curated catalogue of pathology-associated t cell receptor sequences", Bioinformatics **33**, 2924–2929 (2017).

[20]10X Genomics, "LIT047_a New Way of Exploring Immunity - Linking Highly Multiplexed Antigen Recognition to Immune Repertoire and Phenotype", 10xGenomics, 1–13 (2019).

[21]M. E. Birnbaum, J. L. Mendoza, D. K. Sethi, S. Dong, J. Glanville, J. Dobbins, E. Özkan, M. M. Davis, K. W. Wucherpfennig, and K. C. Garcia, "Deconstructing the peptide-mhc specificity of t cell recognition", Cell **157**, 1073–1087 (2014).

[22]J. J. Calis, M. Maybeno, J. A. Greenbaum, D. Weiskopf, A. D. De Silva, A. Sette, C. Keşmir, and B. Peters, "Properties of mhc class i presented peptides that enhance immunogenicity", PLoS computational biology **9**, e1003266 (2013).

[23]S. Frankild, R. J. De Boer, O. Lund, M. Nielsen, and C. Kesmir, "Amino acid similarity accounts for t cell cross-reactivity and for "holes" in the t cell repertoire", PloS one **3**, e1831 (2008).

[24]H. Xia, J. McMichael, M. Becker-Hapak, O. C. Onyeador, R. Buchli, E. McClain, P. Pence, S. Supabphol, M. M. Richters, A. Basu, et al., "Computational prediction of mhc anchor locations guides neoantigen identification and prioritization", Science immunology **8**, eabg2200 (2023).

[25]J. J. Adams, S. Narayanan, M. E. Birnbaum, S. S. Sidhu, S. J. Blevins, M. H. Gee, L. V. Sibener, B. M. Baker, D. M. Kranz, and K. C. Garcia, "Structural interplay between germline interactions and adaptive recognition determines the bandwidth of tcr-peptide-mhc cross-reactivity", Nature immunology **17**, 87–94 (2016).

[26]F. Duan, J. Duitama, S. Al Seesi, C. M. Ayres, S. A. Corcelli, A. P. Pawashe, T. Blanchard, D. McMahon, J. Sidney, A. Sette, et al., "Genomic and bioinformatic profiling of mutational neoepitopes reveals new rules to predict anticancer immunogenicity", Journal of Experimental Medicine **211**, 2231–2248 (2014).

[27]J. Schmidt, A. R. Smith, M. Magnin, J. Racle, J. R. Devlin, S. Bobisse, J. Cesbron, V. Bonnet, S. J. Carmona, F. Huber, et al., "Prediction of neo-epitope immunogenicity reveals tcr recognition determinants and provides insight into immunoediting", Cell Reports Medicine **2** (2021).

[28]D. Chowell, S. Krishna, P. D. Becker, C. Cocita, J. Shu, X. Tan, P. D. Greenberg, L. S. Klavinskis, J. N. Blattman, and K. S. Anderson, "Tcr contact residue hydrophobicity is a hallmark of immunogenic cd8+ t cell epitopes", Proceedings of the National Academy of Sciences **112**, E1754–E1762 (2015).

[29]A. V. Joglekar and G. Li, "T cell antigen discovery", Nature methods **18**, 873–880 (2021).

[30]M. H. Gee, A. Han, S. M. Lofgren, J. F. Beausang, J. L. Mendoza, M. E. Birnbaum, M. T. Bethune, S. Fischer, X. Yang, R. Gomez-Eerland, et al., "Antigen identification for orphan t cell receptors expressed on tumor-infiltrating lymphocytes", Cell **172**, 549–563 (2018).

[31]T. Kula, M. H. Dezfulian, C. I. Wang, N. S. Abdelfattah, Z. C. Hartman, K. W. Wucherpfennig, H. K. Lyerly, and S. J. Elledge, "T-scan: a genome-wide method for the systematic discovery of t cell epitopes", Cell **178**, 1016–1028 (2019).

[32]A. V. Joglekar, M. T. Leonard, J. D. Jeppson, M. Swift, G. Li, S. Wong, S. Peng, J. M. Zaretsky, J. R. Heath, A. Ribas, et al., "T cell antigen discovery via signaling and antigen-presenting bifunctional receptors", Nature methods **16**, 191–198 (2019).

[33]P. Zdinak, S. Grebinoski, J. Torrey, E. Zarate-Martinez, L. Hicks, R. Ranjan, N. Trivedi, S. Arshad, M. Anderson, D. A. Vignali, et al., "De novo identification of cd4+ t cell epitopes", bioRxiv, 2022–11 (2022).

[34]M. H. Dezfulian, T. Kula, T. Pranzatelli, N. Kamitaki, Q. Meng, B. Khatri, P. Perez, Q. Xu, A. Chang, A. C. Kohlgruber, et al., "Tscan-ii: a genome-scale platform for the de novo identification of cd4+ t cell epitopes", Cell (2023).

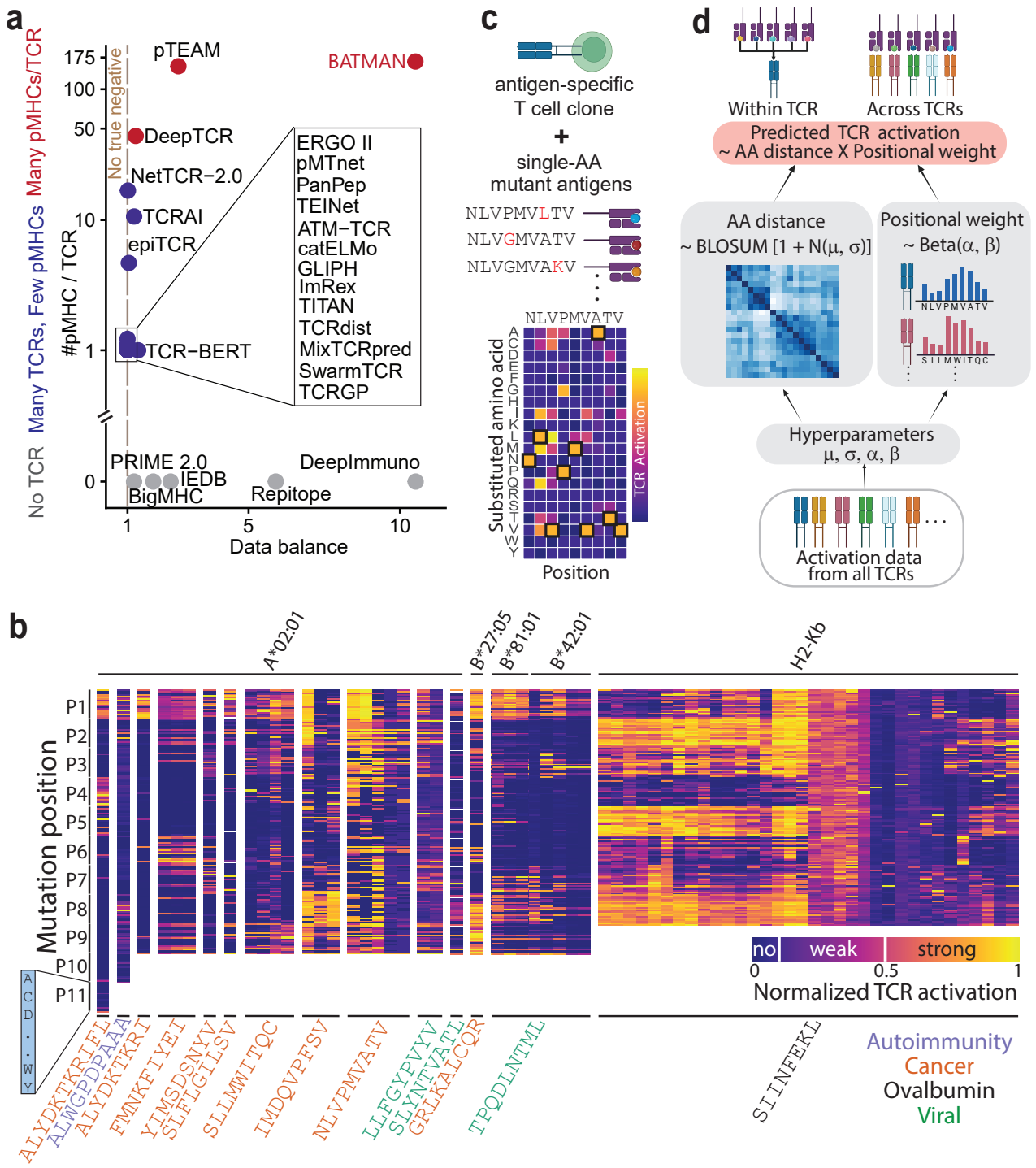


Figure 1. Database and method overview. a. Training dataset summary metrics for TCR-pMHC interaction prediction methods. Data balance is defined as the ratio of total number of TCR-pMHC pairs to the absolute difference in the number of positive and negative pairs. **b.** Curated mutational scan database for TCR activation, with each column corresponding to a TCR clone, grouped by their index peptide (indicated below each column) and recognized MHC (above), and each row corresponding to the substituted AA at a specific position, ordered alphabetically. **c.** Mutational scan assays report activation of a TCR clone against all single-AA mutants of its index (here, NLVPMVATV). **d.** BATMAN integrates mutational scan datasets across many TCRs to build a hierarchical Bayesian inference model. BATMAN infers hyperparameters from the training database and uses them to generate prior distributions for cross-TCR AA distance and TCR-specific positional weights, which are multiplied and used as a predictor of TCR activation by a given mutant.

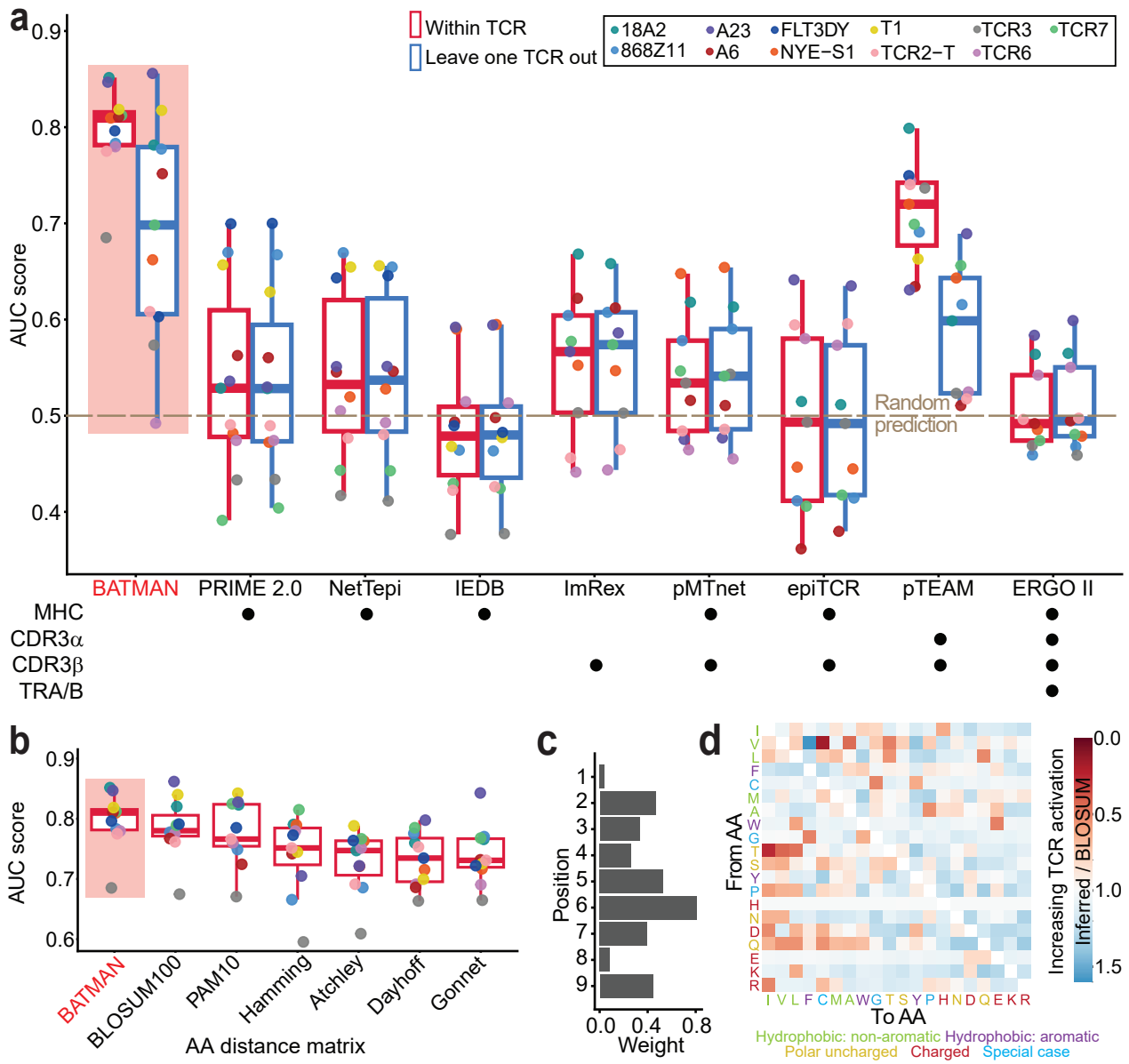
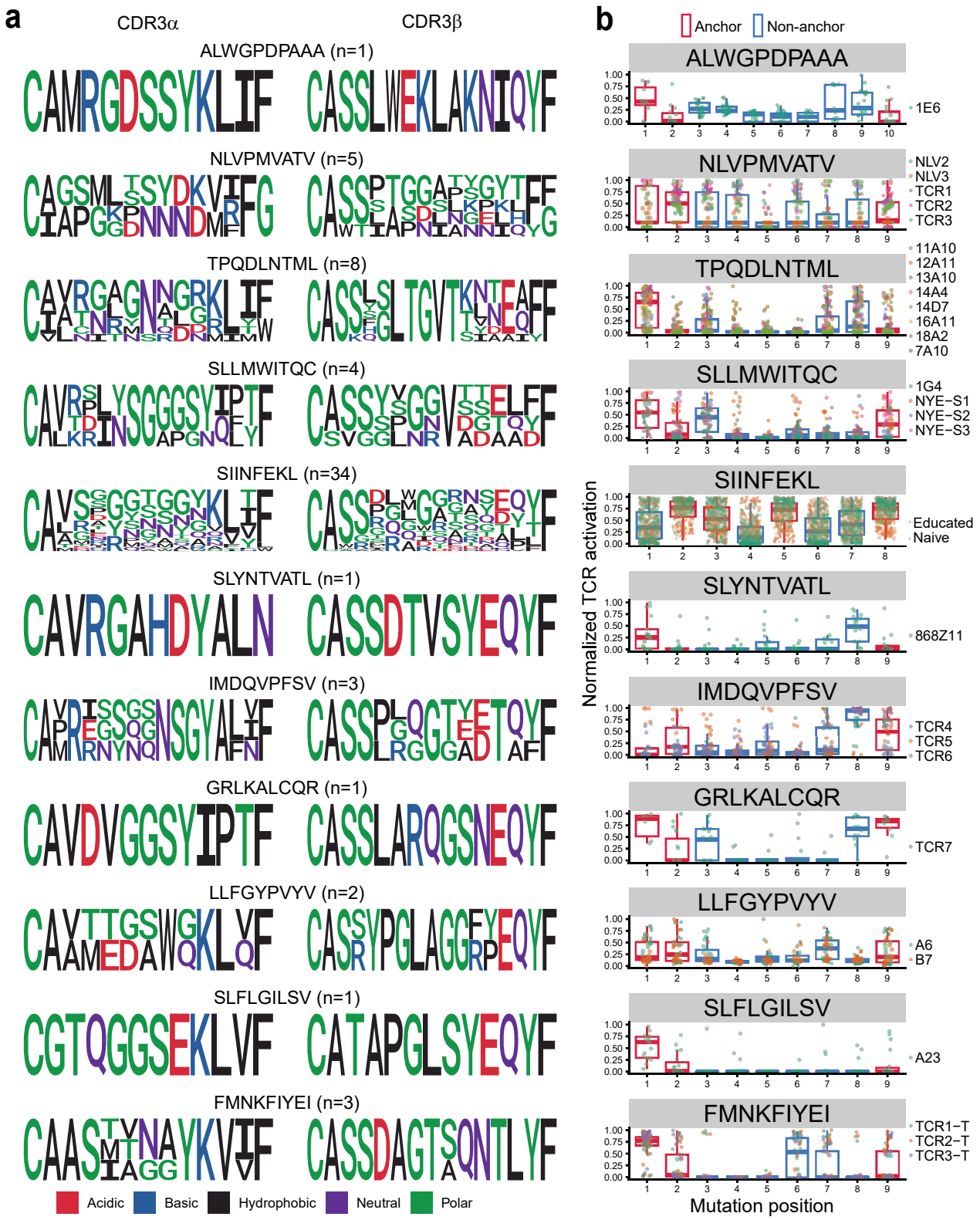
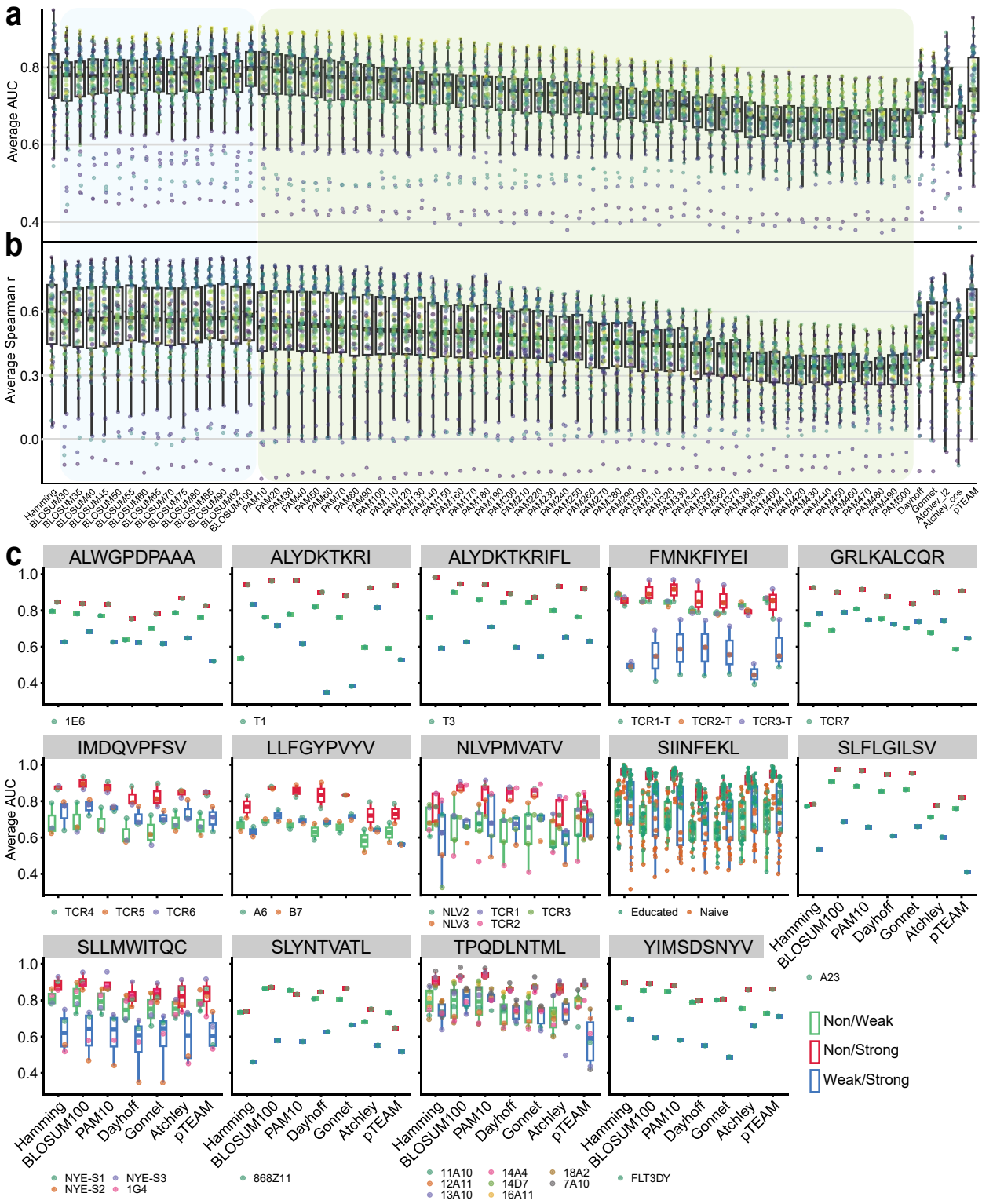


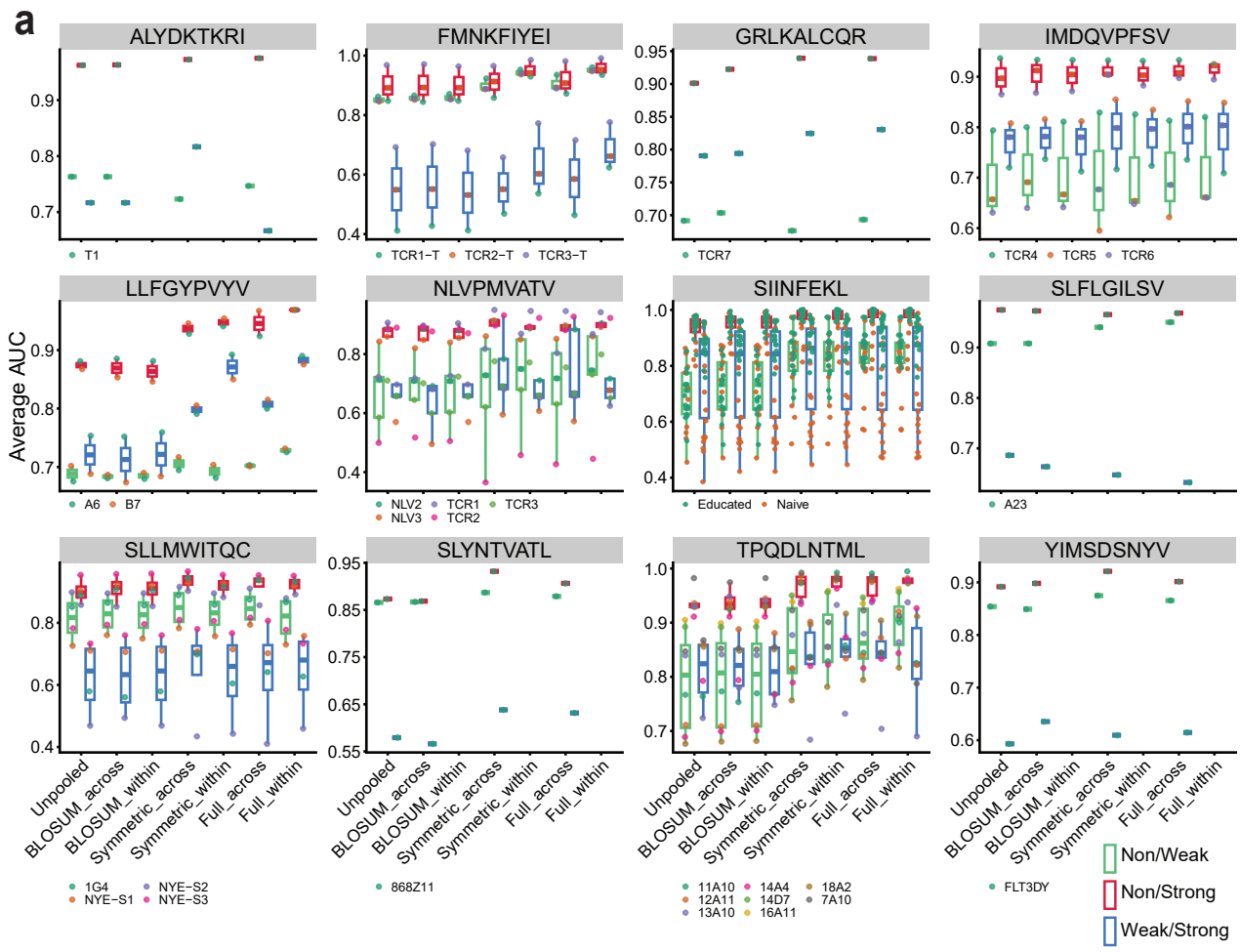
Figure 2. BATMAN outperforms existing TCR-pMHC interaction prediction methods and learns TCR-pMHC biochemistry. a. Average classification area under the curve (AUC) scores for within-TCR and leave-one-TCR-out classification of BATMAN compared with AUC scores from different methods, with their respective requirements indicated (dot matrix). **b.** Within-TCR AUCs when conventional AA distance matrices (indicated) are used. **c.** Example cross-TCR positional weight profile, and **d.** example ratio of inferred matrix elements to BLOSUM100, the best performing conventional AA distance matrix, for within-TCR classification. AAs are ordered by their hydropathy.



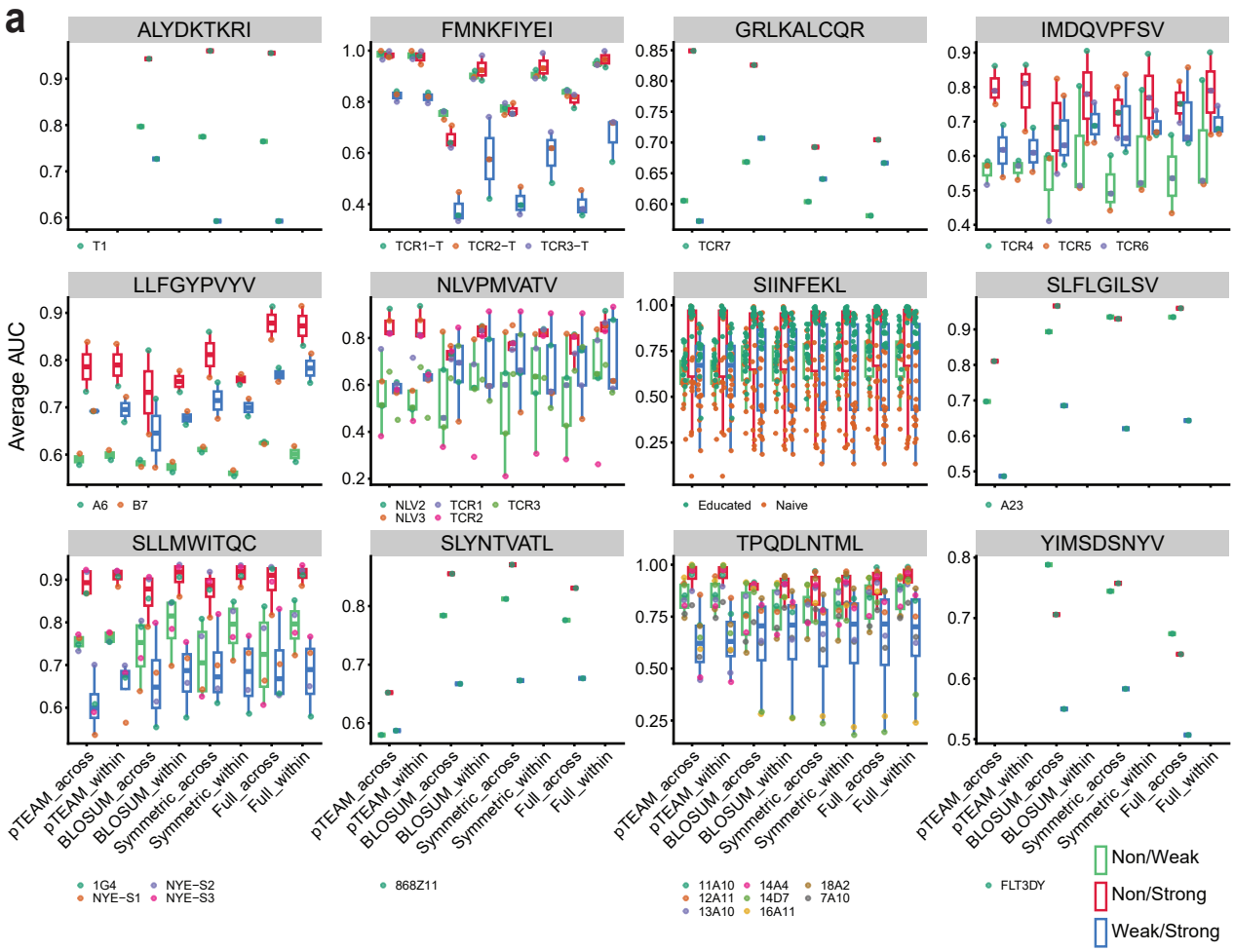
Extended Data Figure 1. **(a)** CDR3α and CDR3β sequence diversity of antigen-specific TCRs present in our database. **(b)** Normalized TCR activation by mutant peptides, grouped by mutation position.



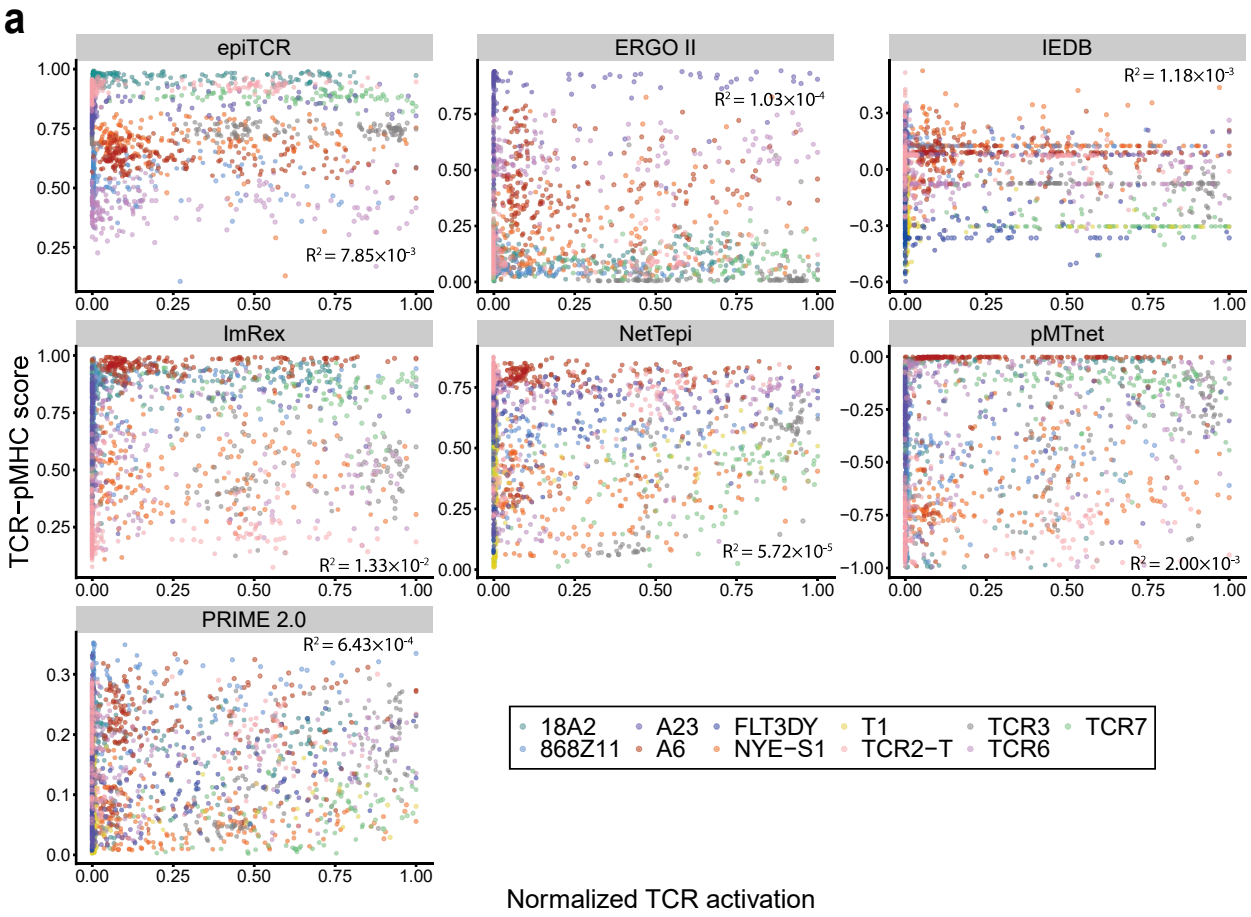
Extended Data Figure 2. Extended, unpooled performance analyses for BATMAN. (a) Classification and **(b)** regression performances in within-TCR tests without cross-TCR pooling using different amino acid distance matrices, with *pTEAM* results shown for comparison (points colored by TCRs). **(c)** Pairwise classification AUC for selected amino acid distances for results plotted in (a).



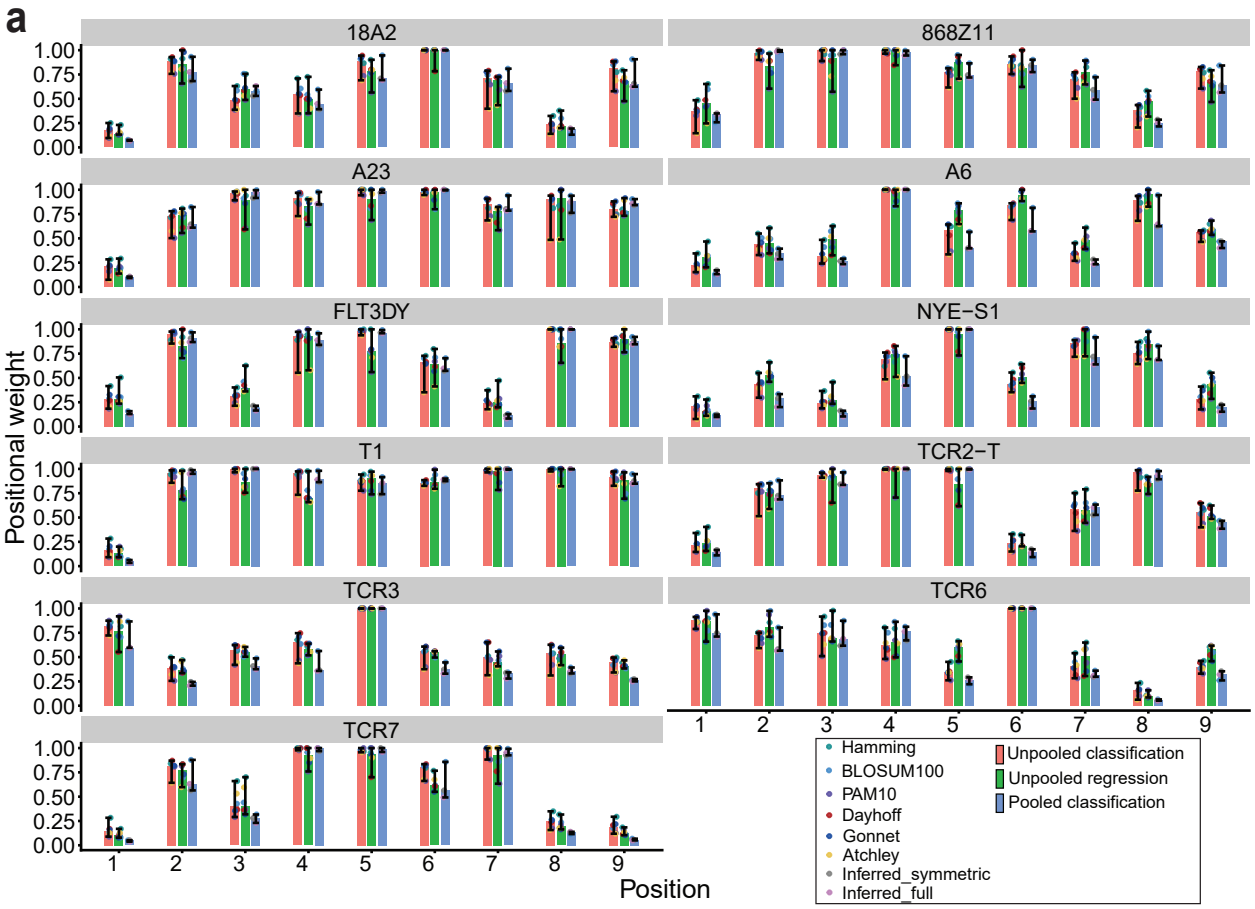
Extended Data Figure 3. Pooling across TCRs improves within-TCR classification performance. (a) Pairwise classification area under the curve (AUC) with different amino acid matrices (*BLOSUM_**, inferred *Symmetric_**, and inferred *Full_** matrices) and pooling modes (**_within* TCRs specific for a index peptide and **_across* TCRs specific for all index peptides of same length). Unpooled results shown for comparison. All BLOSUM results are shown for BLOSUM100, as the best performer as per Extended Data Fig 2a.



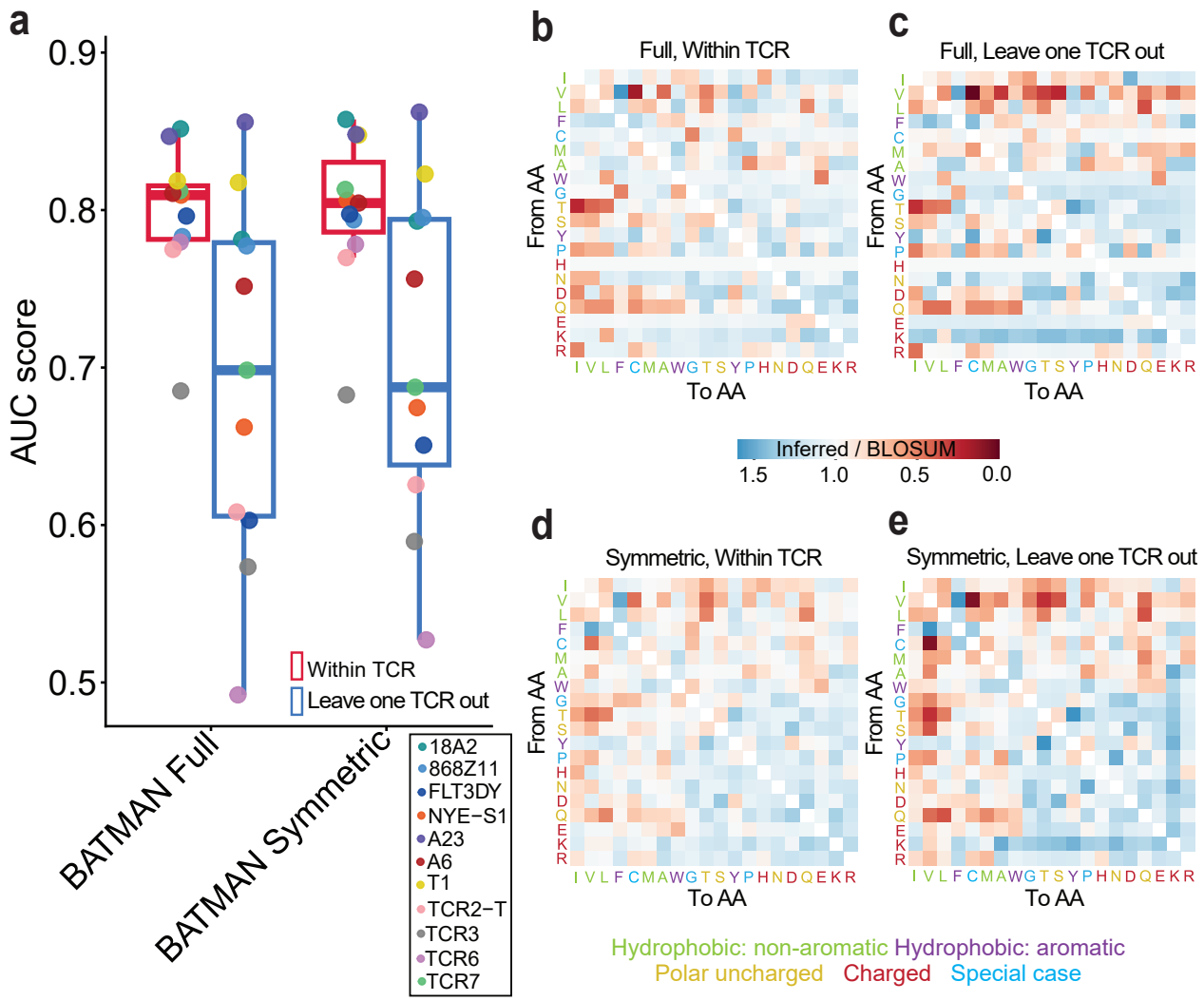
Extended Data Figure 4. Leave-one-TCR-out classification performance of BATMAN compared to pTEAM.(a) Pairwise classification area under the curve (AUC) with different amino acid matrices (*BLOSUM_**, inferred *Symmetric_**, and inferred *Full_** matrices) and pooling modes (**_within* TCRs specific for a index peptide and **_across* TCRs specific for all index peptides of same length). All BLOSUM results are shown for BLOSUM100, as the best performer as per Extended Data Fig 2a.



Extended Data Figure 5. TCR-pMHC scores from different methods do not correlate with TCR activation by mutant peptides (a) TCR-pMHC interaction scores and normalized TCR activation of mutant peptides for the TCRs selected in Figure 2. Note that we used the negative of pMTnet peptide rank as the corresponding TCR-pMHC score, so that higher scores imply stronger TCR-pMHC interactions for all methods.



Extended Data Figure 6. Inferred positional weights for activating TCRs. (a) Weights are consistent across different amino acid distance matrices (points within the error bar), pooling schemes, and classification and regression tasks (indicated by the color of the bars). To make different weight profiles comparable, we normalized positional weights by their maximum over all positions for each weight profile.



Extended Data Figure 7. BATMAN performs similarly when inferring symmetric AA matrix. (a) BATMAN within-TCR and leave-one-TCR-out AUCs with examples of corresponding (b,c) inferred *Full_** and **(d,e)** inferred *Symmetric_** matrices, for the TCRs selected in Figure 2. AAs are ordered by their hydropathy. For generating AA matrices in **(b-e)**, we used the full data from the selected TCRs.

Methods

TCR activation dataset collection and processing

We collected continuous TCR-pMHC datasets for complete single-AA mutational scans from all publications containing raw datasets (n=12). To normalize datasets across publications, we scaled TCR activation values by the maximum activation value over all recorded peptides tested against that TCR. The only exceptions to this normalization scheme were for experiments where the TCR activation measurements were on a logarithmic scale (e.g., EC50 values), in which case we used the logarithm of the TCR activation values and linearly transformed them to map to the [0,1] interval. Following previous works [13], we discretized the normalized TCR activation values to 3 ordered levels for downstream classification tasks: no activation ($a_{no} \in [0, 0.1)$), weak activation ($a_{weak} \in [0.1, 0.5)$), and strong activation ($a_{strong} \in [0.5, 1]$). For regression tasks, we directly used the normalized TCR activation values. More technical TCR-specific notes on data collection and processing, as well as links to source publications, can be found in the Supplementary Notes. A number of publications (see Supplementary Materials for citations) contained further mutational scan experiments relevant for our database, but the associated raw datasets were not readily available to us.

Web application for visualizing TCR-pMHC interactions from our database

TCR-pMHC interactions from our database (Figure 1b) are visualized via the web application at https://batman.cshl.edu/. All interactive plots are deployed as a RShiny application, using *ShinyDashboard* (v. 0.7.2). The scatter plot displaying peptide clustering based on index-to-mutant distance was generated via *ggplot2* (v. 2_3.4.4) and rendered using *plotly* (v. 4.10.3). The heatmap presenting normalized peptide activation per index peptide was generated with *InteractiveComplexHeatmap* (v. 1.8.0). The Alluvium plot visualizing the binding of index and mutated peptides to TCRs was generated with *ggplot2* and *ggalluvial* (v. 0.12.5). The code for the application will be available upon publication.

Training and validation of BATMAN

Bayesian hierarchical classifier for TCR activation

We first describe how BATMAN works for a given TCR in within-TCR validation. For classification tasks, BATMAN (Figure 1d) performs Bayesian logistic regression to predict the ordered categorical activation level for the given TCR and peptide, $a\,(\text{peptide}) \in \{a_{no}, a_{weak}, a_{strong}\}$, using the peptide-to-index distance $d\,(\text{peptide}, \text{index})$ corresponding to the index peptide of the TCR, using this link function:

$$\text{Prob}\left[a\,(\text{peptide})\,|d\,(\text{peptide}, \text{index})\right] = \begin{cases} 1 - \text{logit}^{-1}\left(d_0 - d\,(\text{peptide}, \text{index}) - c_1\right), & \text{if } a\,(\text{peptide}) = a_{\text{non}} \\ \text{logit}^{-1}\left(d_0 - d\,(\text{peptide}, \text{index}) - c_1\right) - \text{logit}^{-1}\left(d_0 - d\,(\text{peptide}, \text{index}) - c_2\right), & \text{if } a\,(\text{peptide}) = a_{\text{weak}} \\ \text{logit}^{-1}\left(d_0 - d\,(\text{peptide}, \text{index}) - c_2\right), & \text{if } a\,(\text{peptide}) = a_{\text{strong}} \end{cases} \quad (1)$$

where the inverse logit function is defined as $\text{logit}^{-1}(x) = \frac{1}{1+e^{-x}}$, d_0 is a constant intercept and c_1 and c_2 are two constant cutpoints with the constraint $c_1 < c_2$, with the following hyperprior distributions:

$$d_0 \sim \text{Normal}\left(\mu_0, \sigma_0\right), \quad (2)$$

$$c_1, c_2 \overset{iid}{\sim} \text{Normal}\left(0, 2\right), \quad (3)$$

$$\mu_0 \sim \text{Normal}\left(0, 2\right), \quad (4)$$

$$\sigma_0 \sim \text{HalfNormal}\left(2\right). \quad (5)$$

For any peptide-index sequence pair, the peptide-to-index distance $d\,(\text{peptide}, \text{index})$ is computed based on position-dependent weights $w\,(\text{position})$ and a 20x20 AA substitution distance matrix M:

$$d\,(\text{peptide}, \text{index}) = \sum_{\substack{\text{position} \\ \in\{1,2,..,L\}}} w\,(\text{position})\, M\left[aa\,(\text{index, position}), aa\,(\text{peptide, position})\right], \quad (6)$$

where each element in $M\left[aa\,(\text{index, position}), aa\,(\text{peptide, position})\right]$ corresponds to the substitution of amino acid residue $aa\,(\text{index, position})$ to $aa\,(\text{peptide, position})$ at a given position in the index and peptide sequences, respectively. The diagonal elements of M are all zero, such that the distance from the index peptide to itself is zero. BATMAN infers the weights $w\,(\text{position})$ and AA distance matrix elements of $M\left[aa_1, aa_2\right]$ with $aa_1, aa_2 \in \{\text{A,C,D,...,W,Y}\}$.

Position-dependent weights $w\,(\text{position}) \in [0,1]$ with position $\in \{1,2,..,L\}$ where L is the length of the TCR's index peptide have the prior:

$$w\,(\text{position}) \overset{iid}{\sim} \text{Beta}\,(\alpha,\beta)\,. \tag{7}$$

Elements of M follow:

$$M\,[aa_1,aa_2] = D\,[aa_1,aa_2]\,(1+\delta\,[aa_1,aa_2])\,, \tag{8}$$

$$\delta\,[aa_1,aa_2] \overset{iid}{\sim} \text{Normal}\,(\mu,\sigma) \tag{9}$$

where D is a pre-defined AA distance matrix (e.g., BLOSUM100) used for constructing the prior for the inferred AA matrix M.

The hyperparameters of $d\,(\text{peptide},\text{index})$ have the following, weakly informative hyperprior distributions,

$$\alpha \sim \text{Gamma}\,(4,4)\,, \tag{10}$$

$$\beta \sim \text{Gamma}\,(25,5)\,, \tag{11}$$

$$\mu \sim \text{Normal}\,(0,0.5)\,, \tag{12}$$

$$\sigma \sim \text{Exponential}\,(1)\,. \tag{13}$$

We verified via prior predictive sampling that these assumptions can yield all anticipated outcomes i.e. activation levels.

Regression tasks with BATMAN

To use BATMAN for regression tasks of predicting continuous-valued normalized TCR activation $a\,(\text{peptide}) \in [0,1]$, we modified Equation (1) to

$$\text{Prob}\,[a\,(\text{peptide})\,|d\,(\text{peptide},\text{index})] = \text{Normal}\,(d_0 - d\,(\text{peptide},\text{index})\,,\sigma)\,,\sigma \sim \text{Exponential}(1), \tag{14}$$

with all other steps being identical as described above for classification tasks. An example of such an application is shown in Extended Data Fig 2b.

Pooling across TCRs for training BATMAN

The hierarchical Bayesian inference set-up allows BATMAN to integrate datasets from multiple TCRs having the same index peptide length ('pooling across TCRs'). In such cases, the positional weight profiles $w\,(\text{position, TCR})$ and the intercepts $d_0\,(\text{TCR})$ are TCR-specific, but have the same prior distributions as specified above, i.e.,

$$w\,(\text{position, TCR}) \overset{iid}{\sim} \text{Beta}\,(\alpha,\beta)\,, \tag{15}$$

and

$$d_0\,(\text{TCR}) \overset{iid}{\sim} \text{Normal}\,(\mu_0,\sigma_0)\,, \tag{16}$$

with the hyperparameters α,β,μ_0 and σ_0 having hyperpriors as above. These TCR-specific weight profiles are used to calculate TCR-specific peptide-to-index distances $d\,(\text{peptide,index,TCR})$ similarly as above,

$$d\,(\text{peptide,index,TCR}) = \sum_{\substack{\text{position}\\ \in\{1,2,..,L\}}} w\,(\text{position,TCR})\,M\,[aa\,(\text{index, position})\,,aa\,(\text{peptide, position})]\,. \tag{17}$$

TCR-specific peptide-to-index distances are consequently used, similar to Equation (1), to construct TCR-specific activation probabilities $a\,(\text{peptide,TCR})$,

$$\text{Prob}\,[a\,(\text{peptide},\text{TCR})\,|d\,(\text{peptide,index,TCR})] = \begin{cases} 1-\text{logit}^{-1}\,(d_0\,(\text{TCR}) - d\,(\text{peptide},\text{index,TCR}) - c_1)\,, & \text{if } a_1 \\ \text{logit}^{-1}\,(d_0\,(\text{TCR}) - d\,(\text{peptide},\text{index,TCR}) - c_1) - \text{logit}^{-1}\,(d_0\,(\text{TCR}) - d\,(\text{peptide},\text{index,TCR}) - c_2)\,, & \text{if } a_2 \\ \text{logit}^{-1}\,(d_0\,(\text{TCR}) - d\,(\text{peptide},\text{index,TCR}) - c_2)\,. & \text{if } a_3 \end{cases} \tag{18}$$

where

$$a_1 : a\,(\text{peptide,TCR}) = a_{\text{non}}$$
$$a_2 : a\,(\text{peptide,TCR}) = a_{\text{weak}}$$
$$a_3 : a\,(\text{peptide,TCR}) = a_{\text{strong}}$$

In both within-TCR and cross-TCR cases, pooling was performed over different positions in the peptide sequence, and different elements of the matrix M, corresponding to different AA substitutions. Pooling across AA substitutions allowed us to assign $M\,[aa_1, aa_2] = D\,[aa_1, aa_2]\,(1+\mu)$ for AA substitutions absent in the training set but present in the test set.

Unpooled BATMAN is implemented in both python (v. 3.11.5) and R (v. 4.2.3), using *pymc* (v. 5.6.1) and *brms* (v. 2.20.4) packages respectively. For all unpooled results shown in this paper, we sampled from the exact posterior using the default settings of the 'No U-Turn Sampler" of *brms*. Hyperprior selection options are less flexible in *brms* than *pymc*, so we used only *pymc* for applications involving pooling. In all such cases, we sampled inferred parameters from approximated posteriors using the "Automatic Differentiation Variational Inference" (ADVI) method, with the convergence criterion being that the loss function did not change by more than 0.1% if the number of iterations was doubled.

Pooling schemes for BATMAN

We tested different parameter inference and pooling schemes for BATMAN. In Figure 2b (except for the results highlighted as 'BATMAN'), Extended Data Fig 2a-c, and 'unpooled' results in Extended Data Fig 3, we did not pool across TCRs, i.e., BATMAN was trained individually for each TCR separately. In these cases, the unpooled inferred weights had a Beta$(2,2)$ distribution as the prior.

In Figure 2b (except for the results highlighted as 'BATMAN'), Extended Data Fig 2a-c, *unpooled* and for all BLOSUM matrices in Extended Data Figs 3 and 4, we did not infer the AA matrix, i.e., M was set to the indicated AA distance matrix (and BLOSUM100 for the *unpooled* results). In other cases where we inferred the matrix M, the pre-defined matrix D was always chosen to be BLOSUM100, since it performed the best among all the conventional AA distance functions in unpooled training for both classification and regression tasks (Extended Data Fig 2). For a subset of cases where we inferred the AA matrix (Extended Data Figs 3, 4 and 7 *Symmetric_** results), we constrained M to be symmetric. For the TCRs in our database, we did not find a significant change in performance if we constrained the inferred AA matrix to be symmetric Extended Data Figs 3, 4 and 7, even though the asymmetric part of the inferred full AA matrix was prominent for hydrophobic AA residues (Figure 2d,Extended Data Fig 7b). For plotting the inferred AA matrices in Figure 2d and Extended Data Fig 7b-e, we divided all matrices by the corresponding values of $1+\mu$ in each case to make their ratios to BLOSUM100 more interpretable.

When pooling across TCRs, for Figure 2a, we pooled across the selected 11 TCRs, and in Extended Data Figs 3, 4 and 7, we pooled within TCRs specific for an index peptide (**_within*) or across TCRs specific for all index peptides of same length (**_within*). BATMAN performance improved by pooling the training data across TCRs, even when inferring TCR-specific weights and using BLOSUM100 (Extended Data Fig 3).

Finally, while in most cases we inferred TCR-specific positional weight profiles, for leave-one-TCR-out tasks (Figure 2a, Extended Data Figs 4 and 7) we inferred a common weight profile for all TCRs in the training set.

Training schemes for BATMAN

For within-TCR validation tasks, we performed 5-fold cross-validation of BATMAN. The folds were stratified by TCR activation levels for classification tasks and TCR activation deciles for regression tasks, and kept identical among all methods (averaged over folds) for comparison.

For TCRs with a sufficient number of peptide examples ($\geq$5) of all 3 activation levels to perform 5-fold cross validation, BATMAN classification performance was quantified in terms of 3 pairwise AUCs based on the peptide-to-index distance $d\,(\text{peptide}, \text{index})$ of each mutant peptide, calculated using TCR-specific or cross-TCR positional weight profile and AA distance matrix inferred by BATMAN. In Figure 2a,b, Extended Data Fig 2a,b, and Extended Data Fig 7a an average of the 3 AUCs are plotted, whereas the rest of the result figures show individual AUCs. For the rest of the TCRs, we discarded examples belonging to the least-represented activation level, and used BATMAN as a two-class classifier. All AUCs were calculated using the *multiclass.roc* function from the *pROC* (v. 1.18.4) package in *R*.

AA distance matrices in prior distribution

To convert conventional AA substitution matrices (D' set to BLOSUM_*, PAM_*, Dayhoff, or Gonnet) into distance matrices D suitable to be used in priors for BATMAN, we performed the transformation

$$D\,[aa_1, aa_2] = \left(1 - \frac{D'\,[aa_1, aa_2]}{D'\,[aa_1, aa_1]}\right)\left(1 - \frac{D'\,[aa_2, aa_1]}{D'\,[aa_2, aa_2]}\right), \tag{19}$$

so that the AA distance matrix D was always symmetric, with diagonal elements equal to zero. The Hamming matrix had all off-diagonal elements equal to 1. To construct *Atchley_** matrices (for Figure 2b and Extended Data Figs 2 and 6), we calculated pairwise L_2 (for *Atchley* or *Atchley_l2*) and cosine (for *Atchley_cos*) distances between 5-dimensional Atchley embedding vectors for respective AAs to construct the matrix D_{ij}.

Other TCR-pMHC interaction prediction methods

Training dataset summary of different TCR-pMHC methods

We compared our benchmarking dataset with the training datasets of existing TCR-pMHC interaction prediction methods (Figure 1a). We estimated (1) the total number of TCRs and pMHCs considered by each, and (2) the statistics of all experimentally validated examples of TCR-pMHC interactions spanning their respective full training datasets (Figure 1a). We discarded any subsampling and artificial generation of training dataset (e.g., by random pairing of pMHCs and TCRs, commonly used to generate artificial negative examples). Further method-specific notes on acquisition of training dataset statistics can be found in the Supplementary Notes.

Implementation of different TCR-pMHC methods

We tested a subset of pre-trained TCR-pMHC methods on our database. The selection was based on availability of webservers, pretrained models, and ease of installing and running models locally. We trained *pTEAM* in both within-TCR and leave-one-TCR-out modes. For the rest of the methods, we used available pre-trained models on our dataset. Each tested method yielded a continuous-values TCR-pMHC interaction score for each mutant-TCR pair, which was used to calculate 3 AUCs for classification tasks that were subsequently averaged in the final results. The Supplementary Notes section contains links and summaries of different methods tested, and more technical details on their applications on our database.

Implementing pTEAM

A recent method, *pTEAM*, was specifically developed to predict TCR activation by mutants. We implemented *pTEAM* following the description in its source preprint [13]. Briefly, we used Atchley embeddings for index and mutant peptides, and, for leave-one-TCR-out tasks, aligned TCR sequences. These embeddings were used as inputs to random forests with 250 trees for classification and regression tasks, with same folds as BATMAN. Each Pairwise AUC was calculated by averaging over two AUCs corresponding to 3 activation level probabilities output from the random forests. We used *R* to align TCR sequences with the *muscle* (v 3.40.0) package and implement the random forests with the *randomForest* (v 4.7-1.1) package. All AUCs were calculated using the *multiclass.roc* function from the *pROC* (v. 1.18.4) package in *R*.

While BATMAN classifiers outperformed *pTEAM* over the diverse set of 11 selected TCRs (Figure 2a) and for most TCRs in within-TCR tasks (Extended Data Fig 2a), the performance difference was not as pronounced in leave-one-TCR-out tasks (Extended Data Fig 4) when training and test sets both contained TCRs specific for the same index peptide. This demonstrated the importance of validating mutant effect prediction methods on diverse, unbiased collections of TCRs, covering as many unique index peptides and mutants as possible, which is absent in the original work introducing *pTEAM* [13]. Note that except for BATMAN and *pTEAM*, all the methods score similarly in within-TCR and leave-one-TCR-out tasks in Figure 2a, since they are pre-trained, and so the difference in AUC is caused solely by the difference in the test sets in these two tasks.

Data availability

The publicly available subset of the fully curated database of TCR-pMHC interactions can be downloaded as an excel sheet from https://github.com/meyer-lab-cshl/BATMAN/tree/main/TCR_epitope_database. The full database will be available upon publication.

Code availability

Custom analysis code was written in python (version $\geq$ 3.10.11) or R (version $\geq$ 3.4.0). The python implementation of BATMAN ('pyBATMAN') can be installed from https://pypi.org/project/pybatman/ and run locally. pyBATMAN installation instructions and input file specifications can be found at https://github.com/meyer-lab-cshl/BATMAN/. Example TCR-pMHC input dataset and python script for running pyBATMAN can be found at https://github.com/meyer-lab-cshl/BATMAN/tree/main/run_batman. An interactive Jupyter notebook tutorial on pyBATMAN usage can be downloaded from https://github.com/meyer-lab-cshl/BATMAN/blob/main/run_batman/pyBATMAN_Tutorial.ipynb.