nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00619-3

Pan-Peptide Meta Learning for T-cell receptor–antigen binding recognition

Received: 8 October 2022

Accepted: 25 January 2023

Published online: 06 March 2023

Check for updates

Yicheng Gao[1,2,5], Yuli Gao[1,2,5], Yuxiao Fan[1,2], Chengyu Zhu[1,2], Zhiting Wei[1,2], Chi Zhou[1,2], Guohui Chuai[1,2], Qinchang Chen[3], He Zhang[2] & Qi Liu[1,2,3,4]

The identification of the mechanisms by which T-cell receptors (TCRs) interact with human antigens provides a crucial opportunity to develop new vaccines, diagnostics and immunotherapies. However, the accurate prediction and recognition of TCR–antigen pairing represents a substantial computational challenge in immunology. Existing tools only learn the binding patterns of antigens from many known TCR binding repertoires and fail to recognize antigens that have never been presented to the immune system or for which only a few TCR binding repertoires are known. However, the binding specificity for neoantigens or exogenous peptides is crucial for immune studies and immunotherapy. Therefore, we developed Pan-Peptide Meta Learning (PanPep), a general and robust framework to recognize TCR–antigen binding, by combining the concepts of meta-learning and the neural Turing machine. The neural Turing machine adds external memory to avoid forgetting previously learned tasks, which is used here to accurately predict TCR binding specificity with any peptide, particularly unseen ones. We applied PanPep to various challenging clinical tasks, including (1) qualitatively measuring the clonal expansion of T cells; (2) efficiently sorting responsive T cells in tumour neoantigen therapy; and (3) accurately identifying immune-responsive TCRs in a large cohort from a COVID-19 study. Our comprehensive tests show that PanPep outperforms existing tools. PanPep also offers interpretability, revealing the nature of peptide and TCR interactions in 3D crystal structures. We believe PanPep can be a useful tool to decipher TCR–antigen interactions and that it has broad clinical applications.

Neoantigens and virus peptides are presented by the major histocompatibility complex I (MHC-I), which elicits an immune response by being recognized by the T-cell receptor on the surface of T cells that carry the CD8 antigen ($CD8^+$)[1]. The identification of immunogenic peptides or clonally expanded responsive T cells provides a great opportunity to develop new vaccines, diagnostics and immunotherapies[2–4]. Accordingly, many experimental approaches, such as tetramer analysis[5], tetramer-associated T-cell receptor sequencing[6] and T-scan[7], have been developed to detect interactions between TCRs and peptides presented by MHC molecules (pMHCs). However, these experimental methods are

[1]Key Laboratory of Spine and Spinal Cord Injury Repair and Regeneration (Tongji University), Ministry of Education, Orthopaedic Department, Tongji Hospital, Bioinformatics Department, School of Life Sciences and Technology, Tongji University, Shanghai, China. [2]Translational Medical Center for Stem Cell Therapy and Institute for Regenerative Medicine, Shanghai East Hospital, Bioinformatics Department, School of Life Sciences and Technology, Tongji University, Shanghai, China. [3]Research Institute of Intelligent Computing, Zhejiang Lab, Hangzhou, China. [4]Shanghai Research Institute for Intelligent Autonomous Systems, Shanghai, China. [5]These authors contributed equally: Yicheng Gao, Yuli Gao. e-mail: qiliu@tongji.edu.cn

time-consuming, technically challenging and costly[8]. Therefore, the accurate prediction and recognition of TCR–antigen pairing is in high demand and represents one of the most computationally challenging issues in modern immunology.

Several computational tools exist to analyse diverse TCR patterns and predict peptide–TCR binding specificity[9–21]. These tools are mainly categorized into three groups: (1) defining quantitative similarity measurements to cluster TCRs and decipher their antigen-specific binding patterns, including TCRdist[22], DeepTCR[23], GIANA[24], iSMART[25], GLIPH[9,10] and ELATE[11]; (2) developing peptide-specific TCR binding prediction models, including TCRGP[12], TCRex[13], and NetTCR-2[14]; and (3) developing peptide–TCR binding prediction models not restricted to specific peptides while requiring the available known binding TCRs for model training, including pMTnet[8], DLpTCR[15], ERGO2[16] and TITAN[17]. Clearly, the tools in the first category cannot be applied directly for peptide–TCR binding recognition and the tools in the second category are restricted to specific peptides with limited utility. Efforts have been made in the third category to develop general peptide–TCR binding prediction models in which both the peptide and TCRs are embedded. However, these methods only show a relatively robust TCR recognition accuracy to learn the binding patterns of peptides with a large number of known TCR binding repertoires. They failed to recognize peptides with few known binding TCRs or those that were not present in the training data due to the substantial diversity of the interaction space[8,15,18]. In other words, these models cannot be generalized to learn peptide–TCR binding patterns of peptides unavailable in the training dataset (unseen) or exogenous peptides to the immune system, while recognition of the binding specificity for neoantigens or exogenous peptides is crucial for immune studies and immunotherapy[19,20].

Accurate identification of TCRs binding any peptide in a pan-peptide manner is challenging[19], as this approach requires creative strategies that (1) fully exploit the information from peptides that are recorded to bind only a few TCRs, (2) adequately generalize to recognize the binding of TCRs to peptides that were previously unseen by the immune system, and (3) clearly reveal which part of the TCR plays a crucial role in binding and recognition. The development of such general and robust peptide–TCR binding recognition methods remains one of the most challenging problems in immunology studies, and it is expected to facilitate the investigation of the biological processes underlying the interaction between antigens and TCRs and the development of personalized immunotherapy[21,26,27].

Therefore, we introduce PanPep for TCR–antigen binding recognition, which is presented as a general and robust framework inspired by the integration meta-learning[28–31] with a neural Turing machine (NTM)[32], for accurately predicting TCR binding specificity with any peptide, particularly neoantigens or exogenous peptides. Specifically, PanPep formulates peptide–TCR recognition as a set of peptide-specific TCR binding recognition tasks in a meta-learning framework that can be generalized to any new peptide-specific tasks in a pan-peptide manner. We apply PanPep in three different settings: majority, few-shot[33] and zero-shot learning[34], which generalize well to any type of peptide–TCR pair recognition. These three settings correspond to the three recognition settings of the peptide-binding patterns: (1) with a large number of known binding TCRs, (2) with few known binding TCRs or (3) with peptides that were not present in the training data, mimicking those peptides that may be totally unseen by the immune system. Our comprehensive testing results show that PanPep significantly outperforms existing methods for TCR recognition of any type of peptide at all three settings, and most importantly, none of the existing tools except PanPep accurately identifies TCRs binding exogenous or new peptides to the immune system. We further demonstrate the utility of PanPep in several clinical applications under the zero-shot setting. Finally, we suggest that the interpretability of PanPep reveals the nature of the interaction between the peptide and TCR in a 3D crystal structure, helping to discover the effects of different TCR amino acids on the interaction between the peptide and its complementarity-determining region 3 (CDR3) loop.

Results

PanPep overview

PanPep is a universal framework constructed to recognize any type of peptide–TCR binding, which is inspired by an integration of of meta-learning[28–31] with an NTM[32] (Fig. 1). In this framework, meta-learning achieves fast adaptation from few samples in different tasks[28,33], and NTM uses external memory to avoid forgetting in the learning process[32]. Therefore, PanPep is able to unify the recognition of peptide–TCR binding for any type of peptide in a pan-peptide manner by formulating the recognition in a meta-learning framework with memory storing of the learned peptide-specific TCR recognition task information. In other words, PanPep is equipped with the intuitions of human learning, similar to the approach that people use to adapt quickly and handle never-seen-before tasks based on experiences from their past. PanPep consists of two main learning modules (Fig. 1): a meta-learning module and a disentanglement distillation module (Methods). It facilitates peptide–TCR binding recognition in three settings: few-shot, zero-shot and majority (Fig. 1, Methods). PanPep starts to predict the peptide-specific binding TCRs in the few-shot setting and then generalizes it to identify TCRs binding unseen peptides in the zero-shot setting while also maintaining the capability of identifying the binding TCR in the majority setting. Specifically, in the few-shot setting, PanPep applies a model-agnostic meta-learning (MAML)-based model[28] at a peptide-specific task level. Peptide–TCR binding recognition is formulated in a peptide-specific manner, where a meta-training and a meta-testing procedure are performed. In the meta-training procedure, the model is trained on a set of peptide-specific TCR binding tasks containing a support set and a query set to obtain peptide-specific learners and update a meta-learner. In the meta-testing procedure, the meta-learner is fine-tuned on a new peptide-specific binding recognition task with only a small number of known peptide–TCR samples available (few support set) and finally tested on the query set (Fig. 1, Methods). In the zero-shot setting, we developed an NTM-inspired disentanglement distillation method to better generalize PanPep for unseen peptide–TCR recognition (no support set) (Methods). The rationale is to map an unseen peptide to a peptide-specific learner generation space (PLGS), where new peptide-specific learners are generated by retrieving the memory based on the PLGS (Methods). This method helps to generalize PanPep to recognize TCR binding to unseen peptides. Notably, we clearly define 'unseen' peptides in this setting: for existing tools using traditional machine learning models, unseen peptides are those that are unavailable in their own training dataset, while for PanPep, unseen peptides are new peptide-specific TCR binding tasks and no support set is available to fine-tune the meta-learner for these peptides (Fig. 1, Supplementary Table 1). In the majority setting, the meta-learner alone serves as a prior, which is used as a model parameter for initialization, and it is retrained for a specific peptide with a large number of known binding TCRs (large support set) (Fig. 1). Collectively, by applying PanPep in these three settings, it is designed as a universal framework that can be adapted quickly and well to recognize any type of peptide–TCR binding. PanPep outputs a continuous binding score between 0 and 1, reflecting the binding probability between a peptide and a TCR. Higher binding scores will have higher binding probabilities for each peptide–TCR pair and the TCR will be more likely to clonally expand.

Existing tools failed on recognition TCRs to unseen peptides

We started our study based on a comprehensively curated peptide–TCR binding dataset (base dataset), which exhibited a long-tail distribution. Directly constructing a model based on these data may bias to the majority setting and fail in the few-shot and zero-shot settings (Supplementary Note 1). Then we benchmarked existing tools to determine

https://doi.org/10.1038/s42256-023-00619-3

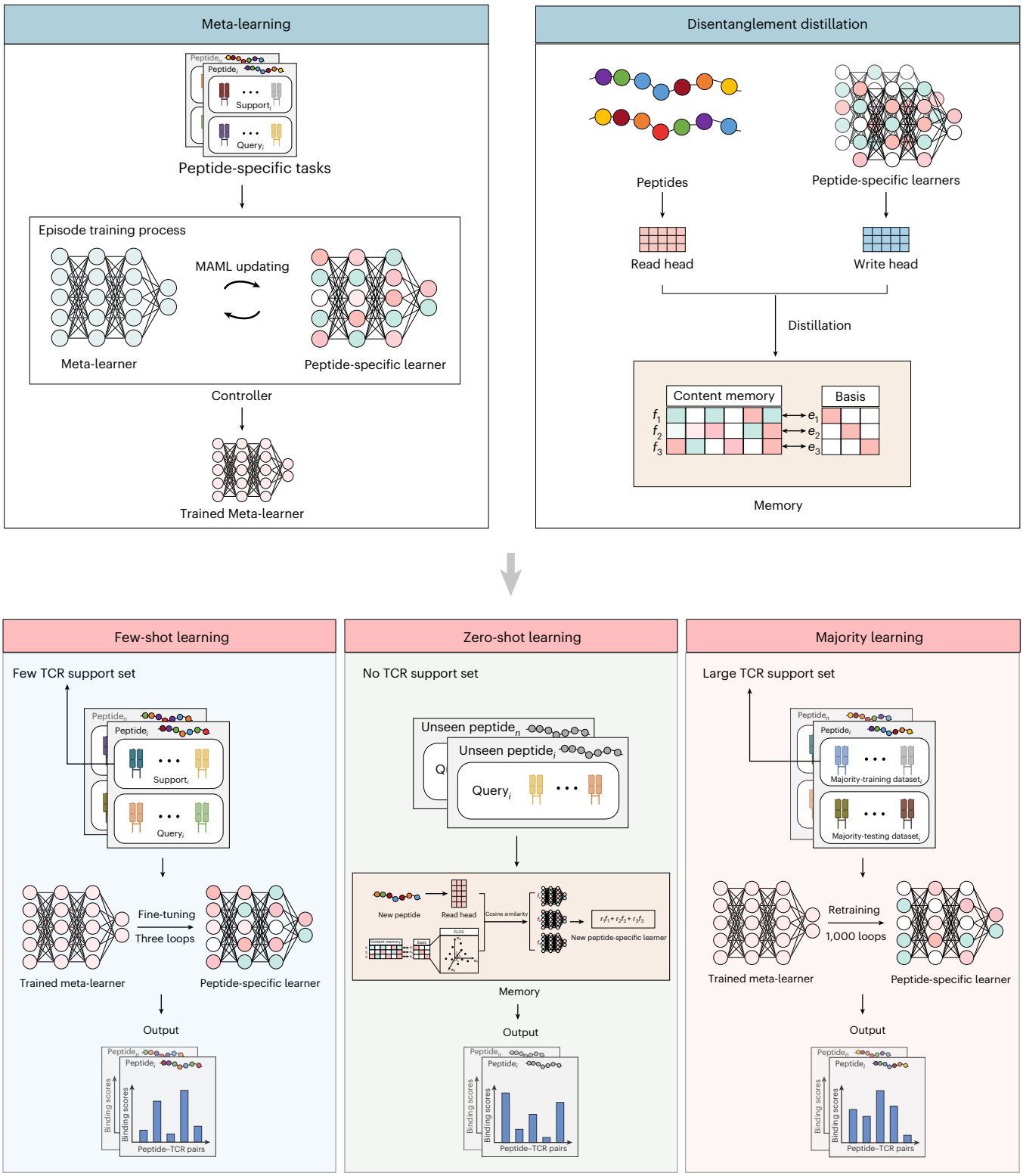

Fig. 1 | Illustration of the PanPep framework. PanPep comprises two modules: a meta-learning module and a disentanglement distillation module. The PanPep framework is tested in three settings. (1) Few-shot setting: PanPep uses the support sets to fine-tune the meta-learner with a few loops (usually approximately three), and the TCR binding recognition is evaluated with the query sets. (2) Zero-shot setting: PanPep extends TCR binding recognition from few-shot learning to zero-shot learning by disentanglement distillation. (3) Majority setting: PanPep uses the meta-learner as a prior, it is retrained with a large number of loops (usually ~1,000 loops), and the TCR binding recognition is evaluated with the query sets.

whether they failed on recognition of TCRs to unseen peptides. To this end, three mainstream tools for predicting peptide–TCR binding from the third category (pMTnet[8], ERGO2[18], and DlpTCR[15]), were tested at three levels with their own testing datasets (Supplementary Methods). Notably, TITAN[17] was excluded because this model removed peptides with few known binding TCRs and is trained on specific COVID-19 data

https://doi.org/10.1038/s42256-023-00619-3

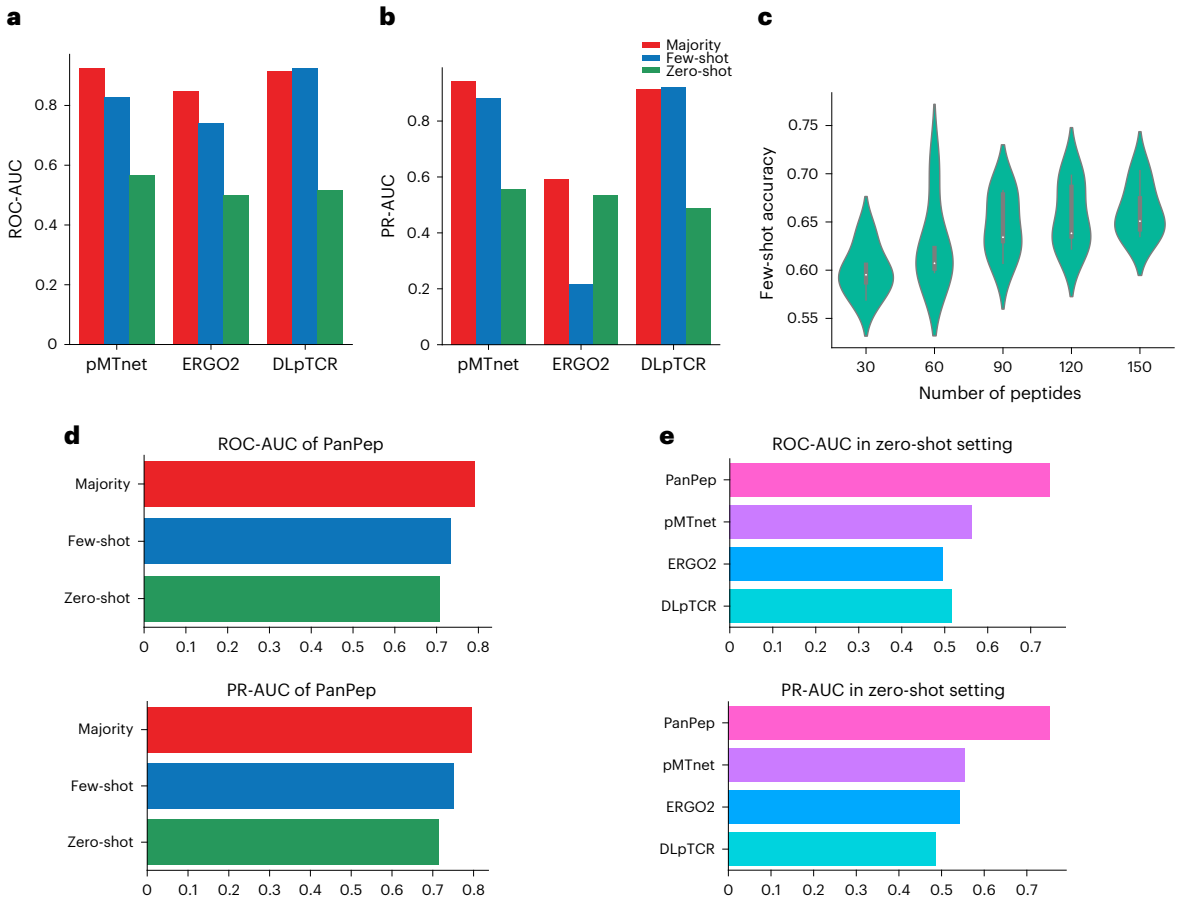

Fig. 2 | The performance of PanPep and other existing tools. a, The ROC-AUCs for pMTnet, ERGO2 and DLpTCR on their own datasets in the three settings. **b**, The PR-AUC for pMTnet, ERGO2 and DLpTCR on their datasets in the three settings. **c**, Effect of the number of peptide-specific tasks in meta-training on the performance of PanPep in few-shot learning. The performance is measured by calculating ACC based on fivefold cross-validation (average ACC of 0.599 for 30 peptides, 0.627 for 60 peptides, 0.647 for 90 peptides, 0.656 for 120 peptides and 0.661 for 150 peptides) and the ACC value of each fold was used as a sample point for generating the violin plot. For the boxplots within the violin plot, box boundaries represent interquartile range, whiskers extend to the most extreme data point no more than 1.5 times the interquartile range and the white dots represent the median. **d**, The ROC-AUC and PR-AUC for PanPep in the three different settings. **e**, The ROC-AUC and PR-AUC for PanPep and the other three tools in the zero-shot setting.

without data on cross-reactive T cells, and thus it has limited applications. In the few-shot setting, pMTnet and ERGO2 both showed poorer performance in terms of area under the receiver operating characteristic curve (ROC-AUC) and area under the precision-recall curve (PR-AUC), compared with in the majority setting (Fig. 2a,b, Supplementary Data 1–3). DlpTCR showed similar performance in the few-shot compared to the majority setting, perhaps because it ensembles 125 deep models and the overfits the data. In the zero-shot setting, all three models generated approximately 0.5 as a random guess in terms of ROC-AUC and PR-AUC, indicating that all these models failed to generalize to recognition of TCR binding to unseen peptides (Fig. 2a,b). In the majority setting, all these models achieved an ROC-AUC greater than 0.8. pMTnet and DlpTCR also achieved a PR-AUC of 0.8, but ERGO2 only achieved a PR-AUC of 0.6 owing to seriously imbalanced samples in the testing dataset (Fig. 2a,b). These failings in peptide–TCR binding prediction in few-shot and zero-shot settings are expected, as all these existing models are biased toward the majority setting, tending to learn the peptide–TCR binding patterns for peptides within a large TCR binding repertoire. This prevents their further application to exogenous or newborn antigen recognition in the clinic.

PanPep achieved good performance in different settings

In this study, we investigated the generalized ability of PanPep to predict peptide–TCR recognition in three different settings. We initially analysed PanPep in the few-shot setting. In this setting, 208 peptide-specific TCR binding repertoires were curated as the **meta-dataset**, and they were randomly split into a **meta-training dataset** and **meta-testing dataset** for model training and testing (Methods).

In the few-shot setting, the support set of each peptide-specific task in the meta-testing dataset is used to fine-tune the meta-learner, and the query set in the meta-testing dataset is used as the test dataset. To test the performance of PanPep in few-shot setting, we perform the fivefold cross-validation in meta-dataset (Methods), and all the binding TCRs in both the support set and query set were balanced using the **control** TCR set (see Methods for the construction of control TCR set). In this case, PanPep achieved an average of 0.734 ROC-AUC and 0.751 PR-AUC in the few-shot setting (Fig. 2d). Furthermore, PanPep achieved better performance because the number of peptide-specific tasks increased in the meta-training dataset (Fig. 2c). Specifically, we started with 30 randomly sampled peptide-specific tasks and gradually increased the number of tasks, showing that PanPep achieved considerably better performance with more available peptide-specific tasks during meta-training (Fig. 2c). This analysis indicated that PanPep effectively uses different peptide-specific TCR binding tasks and improves the meta-learner with a better ability to rapidly adapt to a new peptide-specific TCR recognition task, thus substantially improving performance in the few-shot setting.

In the zero-shot setting, we designed a disentanglement distillation module (Methods) to extend few-shot learning to zero-shot learning, in which no support set is available to fine-tune the model for the new incoming peptide. In the disentanglement distillation module, we constructed mapping between peptide encoding and peptide-specific learners. A read head is designed to map a peptide encoding to a new PLGS (Methods), and a write head is designed to disentangle the peptide-specific learners based on the NTM concept. The memory consists of a basis and content memory, which stores the mapping between peptide embedding and the peptide-specific learner. Then, we can retrieve the memory to generate a new peptide-specific learner based on the embedding of the new peptide. Notably, we compared the different sizes of the basis from one to ten for the reconstruction error, and a basis of three dimensions was selected as the optimal parameter for PanPep (Supplementary Fig. 4a). We subsequently investigated the characteristics of the new peptide embedding generated by the **read head**. Compared with the original Atchley factor encoding of each peptide, each peptide was embedded into three features with PLGS. Based on this new peptide embedding, we found that similar TCR binding repertoires showed significantly smaller peptide distances with p values $< 1 \times 10^{-9}$ (Supplementary Fig. 1f), while the negative correlation between peptide distances and the similarity of TCR binding repertoires became more significant (−0.82) (Supplementary Fig. 1e), clearly indicating that PLGS has a great ability to embed peptides by extracting the nature of the peptide–TCR binding pattern; thus, it can be effectively applied for generalization in the zero-shot setting. We illustrated the effectiveness of our disentanglement distillation module by testing PanPep using the zero-shot dataset consisting of 491 unseen peptides with known binding TCR record used as the query set. In this case, no support set is available to fine-tune the model. Importantly, all the known binding TCRs in the query set were balanced using the control TCR set. PanPep achieved a 0.708 ROC-AUC and 0.715 PR-AUC (Fig. 2d), indicating that PanPep extends peptide–TCR binding recognition from the few-shot to zero-shot setting and generalizes well to unseen peptides. Furthermore, we compared PanPep with the transfer learning model (denoted as baseline model in this work), and PanPep shows a better performance compared with the transfer-learning-based strategy, which further illustrated the effectiveness of the meta-learning and disentanglement distillation strategy (Supplementary Methods and Supplementary Fig. 4b,c).

Finally, we illustrated that PanPep can be easily generalized to the majority setting, in which the three mainstream tools perform relatively well. A majority dataset containing 25 peptide-specific tasks was used to test PanPep (Methods). We evaluated PanPep by retraining (fine-tuning on 1,000 loops) the meta-learner on the support set and classifying the TCRs in the query set for each peptide-specific task. PanPep not only performed well in the few-shot setting and zero-shot setting but also exhibited relatively high performance in the majority setting (ROC-AUC of 0.792 and PR-AUC of 0.797) compared with existing tools (Fig. 2d and Supplementary Data 4).

PanPep outperforms existing methods in the zero-shot setting

In this test, we compared PanPep with existing tools for predicting peptide–TCR binding for unseen peptides, which is the main focus of our study. Thus, the curated zero-shot dataset was applied here as an independent test dataset, in which the peptides in this dataset are unavailable in the meta-dataset of PanPep and in any training sets of other existing tools. Additionally, no support set is available to fine-tune the model for these peptides in PanPep. As a result, PanPep significantly outperformed the other methods in the zero-shot setting, with an ROC-AUC of 0.744 and a PR-AUC of 0.755 (ROC-AUC of 0.563 and PR-AUC of 0.555 for pMTnet; ROC-AUC of 0.496 and PR-AUC of 0.542 for ERGO2; and ROC-AUC of 0.517 and PR-AUC of 0.488 for DLpTCR) (Fig. 2e and Supplementary Data 5). Thus, PanPep can predict peptide-specific TCR binding for unseen peptides, showing great potential for exogenous or newborn antigen recognition in various immunology studies and clinical applications.

PanPep exhibits potential in different clinical applications

We further demonstrated the utility of PanPep in several clinical applications, including (1) qualitatively measuring the clonal expansion of T cells, (2) sorting the responsive T cells efficiently in tumor neoantigen therapy and identifying neoantigen-reactive T-cell signatures, and (3) accurately identifying the immune-responsive TCR among millions of antigen-TCR pairs in large COVID-19 cohort, where none of the existing tools except PanPep can be scaled sufficiently to such a large-scale dataset.

PanPep qualitatively measures clonal expansion of T cells. As T cells with stronger binding to the peptide may undergo more clonal expansion[35], we further validated whether the predicted binding score of PanPep corroborated this impact qualitatively. This study is focused on the data generated from the 10x Genomics Chromium Single Cell Immune Profiling platform, which applies feature barcode technology to generate single-cell 5′ libraries and V(D)J-enriched libraries for TCR sequences and uses a highly multiplexed pMHC multimer reagent to identify binding specificity (Fig. 3a). We examined two single-cell datasets that contain profiles of $CD8^+$ T cells specific to 44 pMHC complexes from two healthy donors with no documented viral infection, and investigated the correlation between their clonal T-cell expansion ratios and the two binding indicators, that is PanPep binding score and original unique molecular identifier (UMI) count (Supplementary Methods). PanPep predicted the binding score for each T cell to the 44 pMHCs in the zero-shot setting and calculated the Spearman correlation coefficient with the clonal expansion ratio, where the same correlation analysis was performed using the UMI count in the previous study. In the comparison, the PanPep score of donors was positively correlated with the clonal ratio of T cells (correlation score of 0.280 for donor 1 and correlation score of 0.234 for donor 2), which is consistent with the expected conclusion that TCRs with higher binding scores are more likely to be clonally expanded (Fig. 3b). By contrast, the correlation between the UMI of donors and the clonal ratio of T cells was approximately 0 (correlation score of 0.045 for donor 1 and correlation score of 0.016 for donor 2), indicating that the UMI-based indicator is not appropriate to qualitatively reveal the clonal ratio of T cells (Fig. 3c). We also performed a digital scale of expansion (0 for unique clones and 1 for expanded clones) and tested the discrimination ability of PanPep score and UMI on it as a classification problem. In this case, PanPep score also exhibited higher performance compared with UMI in the classifications (Supplementary Fig. 5). Nevertheless, we found that the correlation between PanPep score and clone expansion is modest, and the performance improvements using PanPap score compared with UMI in the classification are also relatively small, indicating that the clonal expansion is highly context specific. Taken together, our results suggest that the PanPep score is more likely to be a clonally expanded indicator in a qualitative manner, and this predicted score potentially serves as a more accurate binding indicator for accurate clonal T-cell identification.

PanPep assists with T-cell sorting in neoantigen therapy. Adoptive cell transfer (ACT) is a promising method for cancer immunotherapy, but its efficiency essentially depends on the enrichment of tumour-specific T cells in the graft[22,36]. The effective sorting of T cells responding to specific neoantigens is the key step in ACT-based immunotherapy[9,37]. A previous study combined next-generation sequencing technology with high-throughput immunologic screening to identify tumour-infiltrating lymphocytes from nine patients with metastatic gastrointestinal cancers containing $CD8^+$ T cells that recognize somatic mutational neoantigens[38]. In the present study, we collected 10 different neoantigens and the specific experimentally validated TCRs to

https://doi.org/10.1038/s42256-023-00619-3

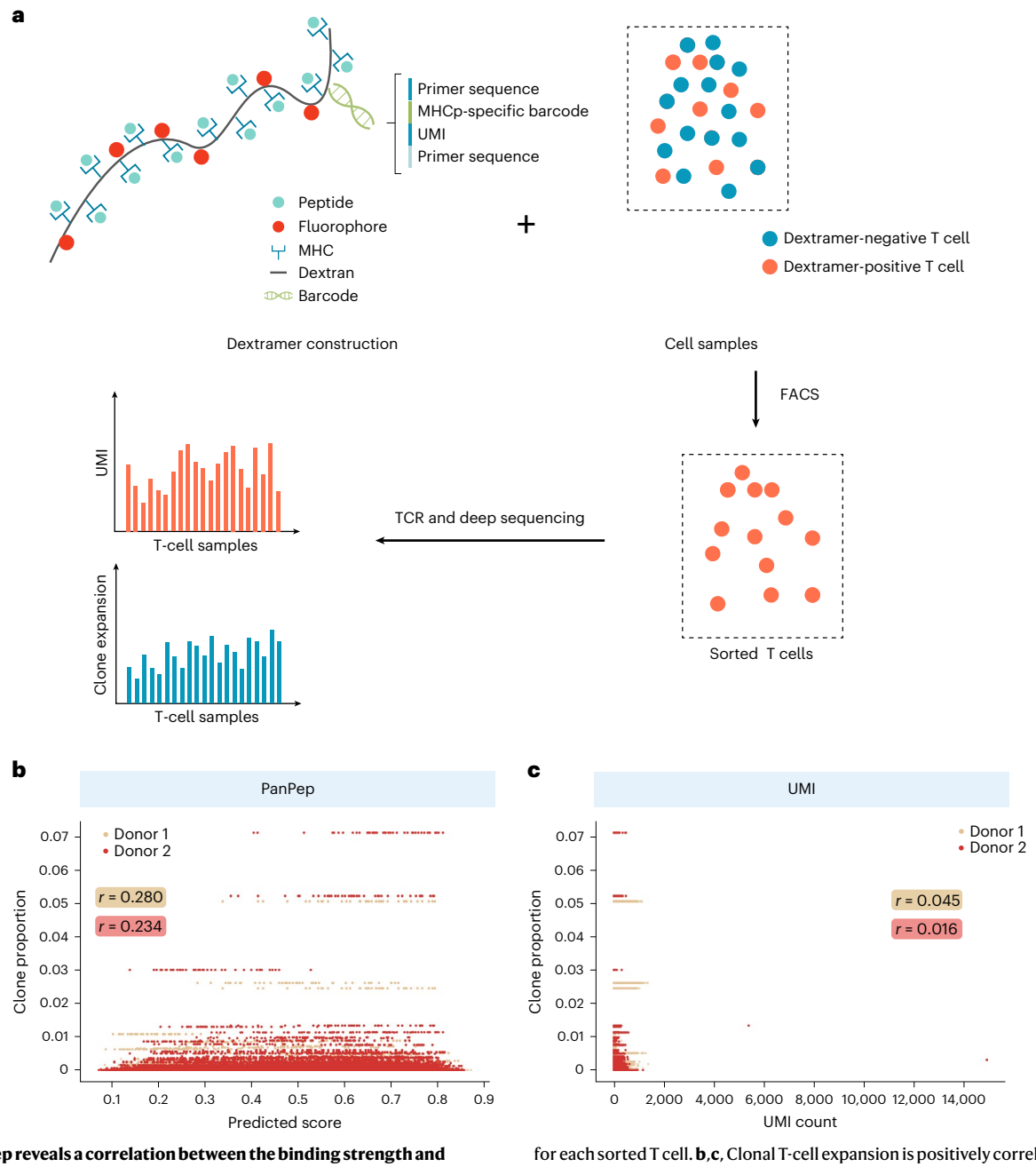

Fig. 3 | PanPep reveals a correlation between the binding strength and clonal T-cell expansion in a single-cell analysis. a, Schematic of the method used to sort and sequence the peptide-specific TCRs using the 10x Genomics platform. T cells were stained with a pool of 50 dCODE dextramers and then sorted by fluorescence-activated cell sorting (FACS). The sorted T cells were sent for sequencing and the UMI count and clonal expansion ratio were recorded for each sorted T cell. **b**,**c**, Clonal T-cell expansion is positively correlated with the predicted binding score using PanPep in the 10x Genomics Chromium Single Cell Immune Profiling datasets from patients with no documented viral infection, while it has no correlation with its own UMI count. *r* is the Spearman correlation score.

which they bind based on previous work[38]. All the validated binding TCRs were balanced by the control TCR set, and we tested whether the immune-responsive T cells were identified accurately by PanPep in the zero-shot setting (Supplementary Data 6 and Supplementary Methods). Compared with the failure of pMTnet, ERGO2 and DlpTCR to identify immune-responsive T cells, PanPep exhibited the best performance with an ROC-AUC of 0.7067 and a PR-AUC of 0.7821 for immune-responsive T-cell identification of neoantigens (Fig. 4a,b).

Furthermore, PanPep identifies neoantigen-reactive T-cell signatures. A recent study performed immunologic screening and single-cell RNA sequencing on five patients with gastrointestinal cancer to characterize the transcriptomic landscape of CD8$^+$ immune-responsive T cells[39]. In the present study, we collected 68 neoantigens and 1,448 CD8$^+$ T cells with CDR3 beta sequences available from these five patients[39] (Supplementary Methods). PanPep was then used for T-cell sorting and identified the neoantigen-reactive T cells in the zero-shot setting (Supplementary Data 7). We found that T cells in the responsive group exhibited significantly higher exhaustion scores and cytotoxicity scores (Fig. 4c). Responsive CD8$^+$ T cells tended to coexpress the *CXCL13* and *GZMA* genes at high levels, as shown in a previous study[39] (Fig. 4d). In addition, we also calculated the distribution of *GZMA*/*CXCL13* among patients (Supplementary Fig. 6). We subsequently analysed

https://doi.org/10.1038/s42256-023-00619-3

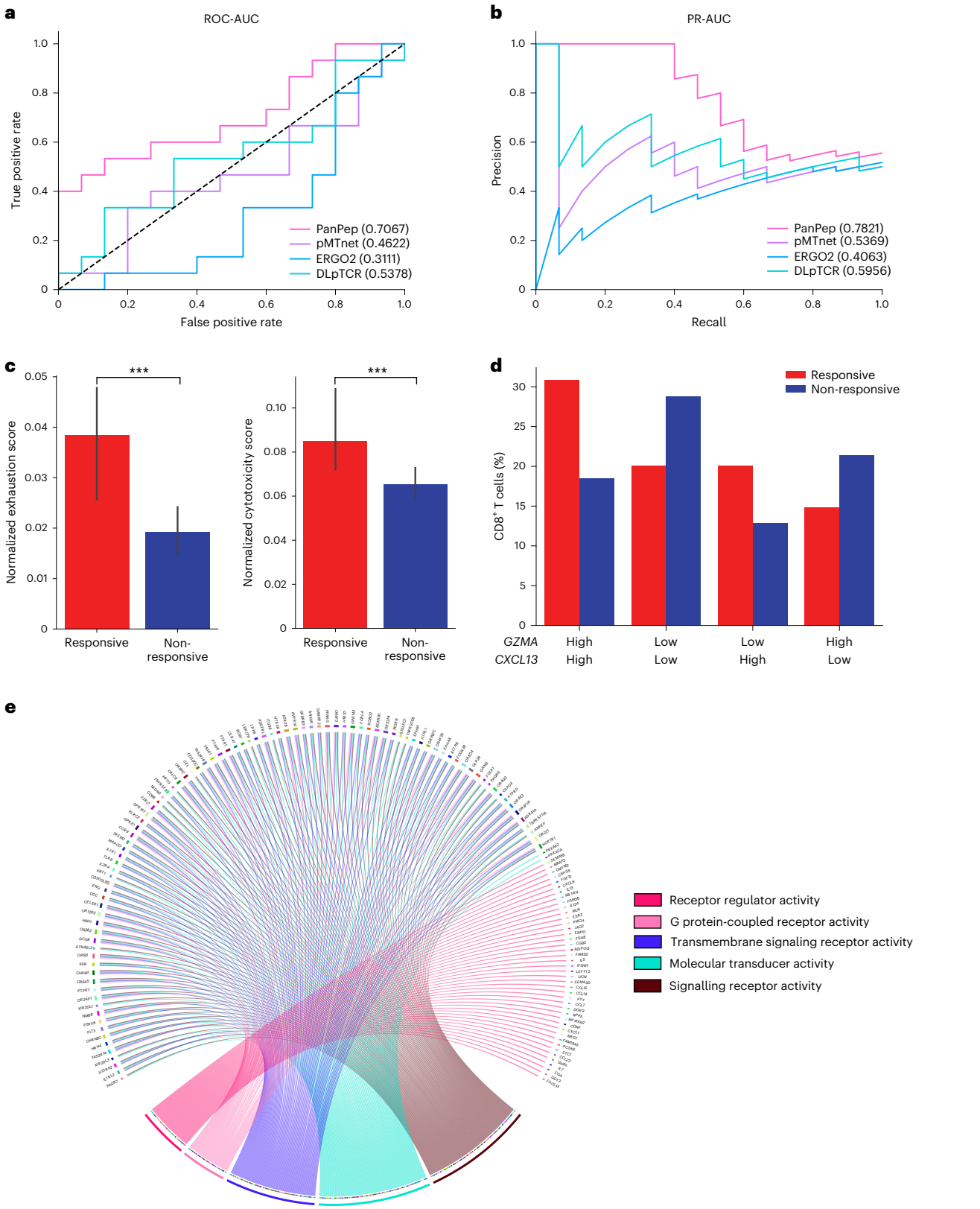

Fig. 4 | Validation of PanPep in sorting immune-responsive T cells and identifying the neoantigen-reactive T-cell signature. a, ROC-AUC for PanPep, pMTnet, ERGO2 and DLpTCR using gastrointestinal cancer dataset[38] reported previously. **b**, PR-AUC for PanPep, pMTnet, ERGO2 and DLpTCR in immune-responsive T-cell sorting. **c**, Exhaustion scores and cytotoxicity scores of T cells in the responsive group (a total of 324 independent T cells) and nonresponsive group (a total of 1,122 independent T cells). Both scores were normalized using max–min normalization. The data are represented as median values and the error bars show the 95% confidence interval of the median estimate (1,000 bootstrap iterations). The exhaustion scores and cytotoxicity scores are significantly higher in the responsive group compared to the nonresponsive group, with *P* values of 0.0005 and 0.0001, respectively. The *P* value was calculated by the two-sided *t* test. ***$P < 0.001$. **d**, The ratio of $CD8^+$ T cells with different levels of *GZMA* and *CXCL13* coexpression between the responsive group and the nonresponsive group. **e**, Upregulated genes that were differentially expressed in responsive T cells and nonresponsive T cells are enriched for T-cell functions.

https://doi.org/10.1038/s42256-023-00619-3

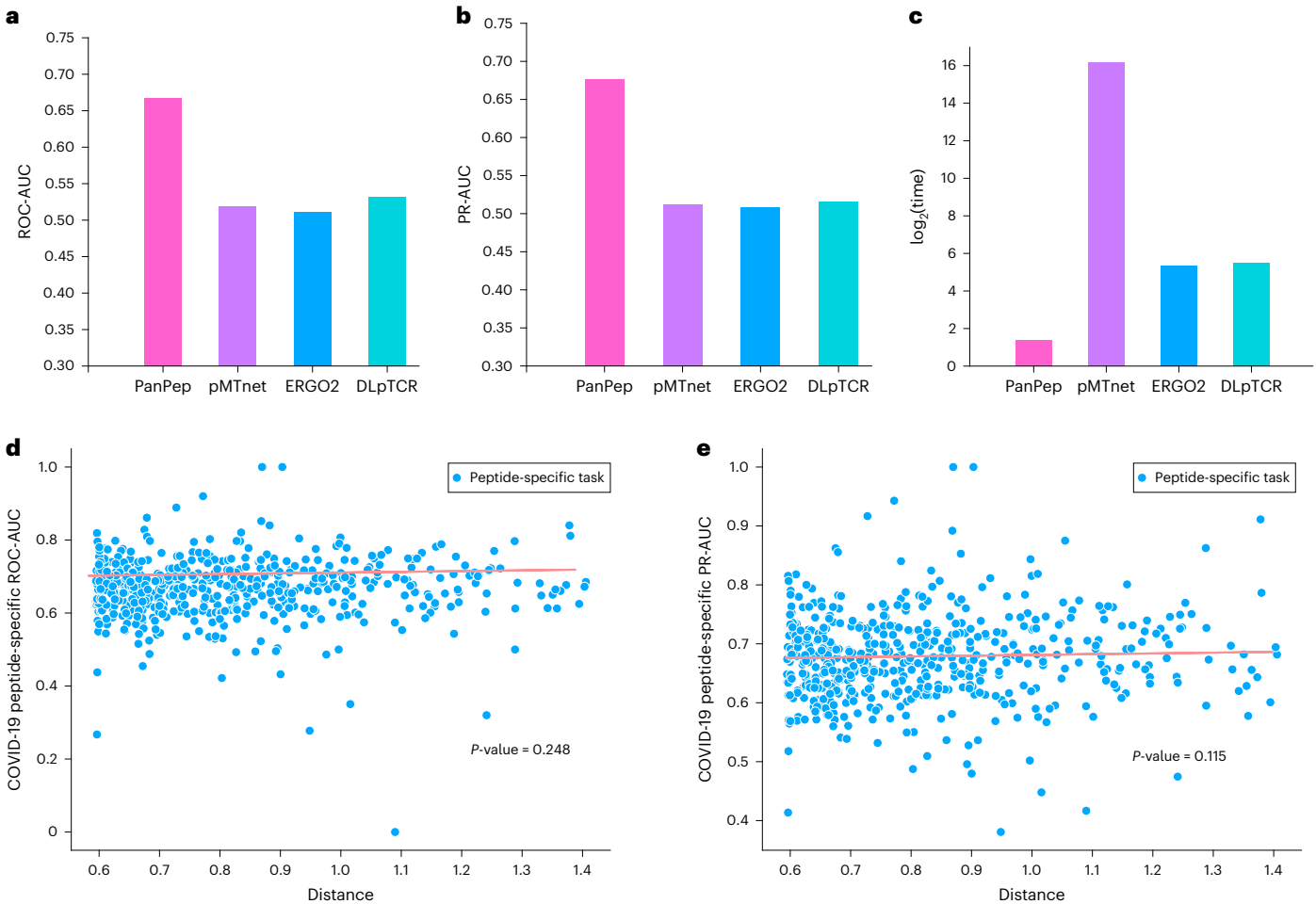

Fig. 5 | Validation of PanPep using the COVID-19 dataset. a, ROC-AUC for PanPep, pMTnet, ERGO2 and DLpTCR using a large independent COVID-19 dataset. **b**, PR-AUC for PanPep, pMTnet, ERGO2 and DLpTCR using a large independent COVID-19 dataset. **c**, Comparison of the computational efficiency of different tools. The running times are $\log_2$ transformed. **d**, The ROC-AUC of PanPep shows no correlation between the distance of the peptide from the COVID-19 dataset and the peptides from the meta-training dataset. The *P* value was calculated by a Spearman correlation test. **e**, The PR-AUC of PanPep shows no correlation between the distance of the peptide from the COVID-19 dataset and the peptides from the meta-training dataset. The *P* value was calculated by a Spearman correlation test. The *x* axis in **d** and **e** represents the minimum distances between peptides from the COVID-19 dataset and the peptides in the meta-training dataset. The *y* axis in **d** and **e** represents the performance of identifying TCRs binding to the peptide.

differentially expressed genes with a fold change >2 and FDR < 0.01 between the responsive T-cell group and the nonresponsive T-cell group. Gene ontology analysis showed that these upregulated genes were enriched in functions playing important roles in many immune processes[40–42], such as T-cell migration and activation (Fig. 4e). Overall, our results further suggested that PanPep can be applied to sort immune-responsive T cells in neoantigen screening, helping to identify neoantigen-reactive T-cell signatures that showed great potential in the development of effective ACT-based tumour immunotherapy.

PanPep exhibits high performance in virus immune recognition. We further investigated the applications of PanPep in virus immune recognition for the COVID-19 study. We collected a recently published dataset from a COVID-19 cohort provided by the ImmuneCODE project[43,44], which collected millions of TCR sequences from more than 1,400 subjects exposed to or diagnosed with COVID-19, and high-confidence, virus-specific TCRs were recorded[44]. We curated this large COVID-19 dataset, resulting in a total of 1,168,776 peptide–TCR pairs associated with 541 different peptides (Supplementary Methods).

As a result, PanPep achieved an ROC-AUC of 0.668 and a PR-AUC of 0.677 in this zero-shot setting (Fig. 5a,b). Compared with the other three tools with an ROC-AUC of approximately 0.5 and a PR-AUC of 0.5, which is similar to a random guess, PanPep exhibited a substantial improvement in peptide-specific TCR recognition (Fig. 5a,b). In addition, we investigated whether PanPep performed well when the distance of COVID-19 peptides is far from those peptides in the meta-training dataset. For this case, all peptide embeddings were generated by the read head of PanPep. The Euclidean distance between one COVID-19 peptide and each peptide in the meta-training dataset of PanPep was calculated, and the minimum distance was selected. The performance of PanPep showed no correlation between the minimum distance and the performance of specific COVID-19 peptides in terms of ROC-AUC (Spearman's correlation analysis, $P = 0.248$) and PR-AUC (Spearman's correlation analysis, $P = 0.115$), indicating that the performance of PanPep is robust even for the most distant peptides in the COVID-19 dataset (Fig. 5d,e). Overall, our results from this large independent COVID-19 cohort study further suggest that PanPep was generalized well across species since the training peptides for PanPep are all from *Homo sapiens* while it is applied for virus peptide recognition. In addition, we compared the computational efficiency of PanPep to the other three tools. Because of the data input size limitation of ERGO and DLpTCR, we must randomly select 100,000 examples from the COVID-19 dataset to fairly benchmark the computational efficiency of these tools. As a result, PanPep required the significantly shortest time to complete the

https://doi.org/10.1038/s42256-023-00619-3

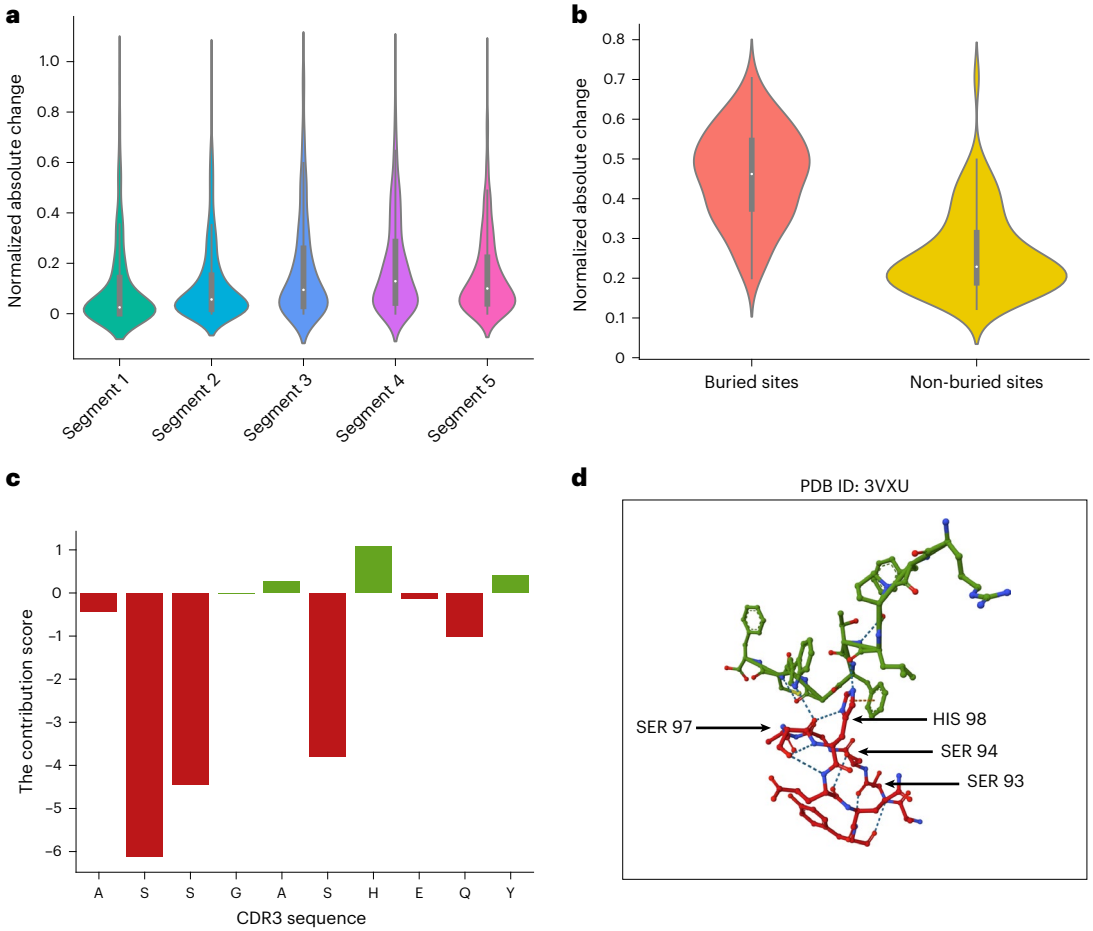

Fig. 6 | Validation of PanPep in the identification of crucial sites in the 3D crystal structure. a, Residues in the middle of the TCR CDR3 sequence show significant changes in the score predicted by PanPep through alanine scanning mutagenesis. A total of 561 TCR CDR3 sequences that can be divided into five segments of equal length from the zero-shot dataset was used to perform the alanine scanning. The absolute change in the predicted scores of mutated TCR to the original TCR was normalized by max–min normalization in each segment for the peptide. The difference in values between Part 2 and Part 3 and the other parts are significant, calculated by the two-sided *t* test, with $P < 0.05$. **b**, Residues in buried sites of the TCR CDR3 sequence show significant changes in the predicted score using PanPep by alanine scanning mutagenesis compared with the nonburied sites, calculated by the two-sided *t* test, with $P < 1 \times 10^{-10}$. A total of 77 peptide–TCR 3D crystal complexes from the IEDB database were analysed here. A similar absolute change value normalization method was performed. For the boxplots in the violin plot (**a**,**b**), the box boundaries represent interquartile range, whiskers extend to the most extreme data point no more than 1.5 times the interquartile range, and the white solid dot in the middle of the box represents the median. **c**, The contribution score of each residue in the TCR CDR3 sequence. The contribution score was calculated as follows: (1) calculating the averaged attention scores of that residue in CDR3 to all amino acids in the peptide, which is denoted as pep-score; (2) calculating the averaged attention scores of that residue in CDR3 to all amino acids in CDR3 itself, which is denoted as TCR-score; and (3) the contribution score is defined as $\log_2$(pep-score/TCR-score) for each residue. A higher positive contribution score indicates that the residue is more important to the peptide, and a lower negative contribution score indicates that the residue is more important to the TCR. **d**, The 3D crystal structure of 3VXU. Red, CDR3 sequence of TCR; green, peptide epitope.

prediction at 4.27 seconds due to the fast generation of peptide-specific learners in the zero-shot setting, while pMTnet required approximately 20 h 30 min (Fig. 5c). This result further indicated the significantly high computational efficiency of PanPep in immune recognition, and none of the existing tools except PanPep can be scaled sufficiently to the large-scale dataset containing millions of antigen–TCR pairs.

PanPep discovers crucial sites in peptide–TCR 3D structure

Although PanPep is designed at the 1D sequence level, we showed that it can perform a simulated mutation analysis to obtain structural evidence of CDR3 residues whose mutations lead to significant changes in the predicted binding between the TCR and peptide in the 3D structure. The alanine scanning technique[45] is a site-directed mutagenesis method used to discover the contributions of specific residues to the function of a particular protein in structural biology studies. Therefore, we performed residue-wise alanine substitution for the TCRs in the zero-shot dataset, and PanPep predicted the binding scores of these mutated TCRs to their specific peptide through zero-shot learning. We then divided CDR3 into five equal segments and, as expected, residues in the middle segments of CDR3 are more likely to be in close contact with peptides, resulting in a larger change in predicted binding compared to those in the other segments (Fig. 6a). Additionally, we collected 77 peptide–TCR 3D crystal complexes from the IEDB database[46] and manually curated the buried fraction of each residue in CDR3 (Supplementary Data 8 and Supplementary Methods). We next performed the same simulated alanine scanning mutation of these TCR CDR3s, and each TCR was divided into groups of buried and nonburied sites based on whether the buried fraction value was larger than 0. We clearly showed that the residues in buried sites indeed induced a significant change compared with the nonburied sites (*t* test, $P < 1 \times 10^{-10}$; Fig. 6b). These finer 3D structural data support the hypothesis that PanPep captures the residues involved in contact between peptides and TCRs in the 3D

structure. Furthermore, the interpretable mechanism of the PanPep model was shown with an example of a peptide–TCR crystal structure[47] (Protein Data Bank (PDB) ID: 3VXU; Fig. 6d). PanPep identified the H residue with highest contribution score, which involves the pi–cation interaction with peptide (Fig. 6c; Supplementary Note 2). Overall, our results suggest that although PanPep was constructed at the 1D sequence level by taking advantage of the attention mechanism, it can be applied to capture the nature of the interactions between peptides and TCRs and discover crucial sites in 3D structures.

Discussion

Owing to the broad clinical applications of recognizing TCRs binding to unseen peptides and the vast generation space of unseen peptides, such as neoantigens and exogenous virus peptides, an effective and robust peptide–TCR binding recognition model for any type of peptide, especially neoantigens and exogenous peptides, is needed. Therefore, PanPep is designed as a universal framework for robust peptide–TCR binding recognition in various settings, including few-shot learning, zero-shot learning and majority learning.

Notably, in the current study, we did not consider the alpha chain of TCR in our model, as most of the previous studies did not consider this point either[8,17]. The main reason lies in the limited availability of TCR data with both alpha and beta chain information in existing databases (see Supplementary Methods for detailed statistics of the alpha chain information in existed databases)[14,48]. We also did not consider the HLA typing for peptide–TCR binding recognition at present, while PanPep can be easily extended by incorporating the HLA type information and achieved acceptable prediction ability (Supplementary Note 3). The limited availability of HLA type information at present limits the downstream applications. However, future updating is expected when more information on alpha chains and HLA typing are available.

Although PanPep outperformed other methods significantly, the performance can be improved especially for the few-shot and zero-shot settings when more training data are available. Future updating of PanPep is expected. (1) The performance of PanPep was limited by the number of available peptides that could be used to construct the peptide episodes. Increased availability of experimentally validated data will further improve the performance of PanPep, and a carefully designed model can also be developed for output evaluation with confidence score. (2) Currently, PanPep only considers the interaction of the CDR3 beta chain of TCR with the peptide. Recent work[49] also suggested that catch bonds[50] in other chains may play an important role in the TCR recognition of neoantigens, and thus future modelling considering more information, such as the HLA type and alpha/beta chains, is expected. (3) As more 3D crystal structure data become available in the future, building models with 3D data directly will further improve peptide–TCR binding recognition. (4) More direct applications are expected to illustrate the superiority and utility of PanPep in the future.

Methods

Design of PanPep

We constructed PanPep and applied it to three settings, that is, the few-shot setting, zero-shot setting and majority setting, to address the challenge of the long-tail data distribution of the TCR binding repertoire and generalize the recognition of TCR binding to unseen peptides. The framework consists of two main learning modules (Fig. 1), that is, the meta-learning module and disentanglement distillation module. In the meta-learning module, the meta-learner is responsible for capturing the meta information about the interactions between TCRs and peptides by operating across different peptide-specific tasks. The peptide-specific learner is responsible for capturing the peptide-specific TCR recognition information by operating on each peptide-specific task. In the disentanglement distillation module, we constructed the map between peptide embedding and peptide-specific learner through model distillation.

We trained PanPep by generating a set of peptide-specific tasks, where each peptide-specific task consists of a support set .. and a query set $\{\mathrm{tcr}_i, y_i\}_{i=1}^{Q}$. Each pair of tcr and y represents a TCR sequence and a label indicating whether this TCR recognizes this peptide, respectively. In this study, a peptide-specific task consisting of a support set and a query set is called a peptide episode (Supplementary Fig. 2a). Overall, the PanPep training process consists of three main procedures: acquisition of high-order meta information, storage of the peptide-specific learner parameters and construction of a mapping between peptide and peptide-specific learners. The first two steps are performed collectively by the meta-learner and peptide-specific learner in the meta-learning module, while the last step is performed by NTM through model distillation in the disentanglement distillation module.

In the testing procedure, we sampled another set of peptide-specific tasks with these peptides unseen to the training tasks to perform a pan-peptide test of PanPep that would indicate its generalizability. Then, the trained model was fine-tuned on its support set and applied to classify the TCRs in its query set. Specifically, PanPep was tested in three settings. In the few-shot setting, we sampled another set of peptide-specific tasks with peptides that were unseen in the training tasks, and the number of TCRs in the support set was very small. Then, the model was fine-tuned for a few epochs on its support set and applied to classify the TCRs in its query set. In the zero-shot setting, we collected the peptide-specific tasks with these peptides that were unseen to the training tasks, and no support sets were available for their tasks. Then, the peptide-specific model was generated based on peptide embedding using the disentanglement distillation module, thereby classifying the TCRs in its query set. In the majority learning setting, we also sampled another set of peptide-specific tasks with a large number of support set samples. Then, the model was fine-tuned for a large number of epochs on its support set and applied to classify the TCRs in its query set. The difference between the few-shot setting and the majority setting is that the number of TCRs in the support set was large in the majority setting compared to that of the few-shot setting; therefore, the fine-tuning applied in the majority setting tended to prevail over the prior rather than that of a fast adaptation from the prior in the few-shot setting.

Notably, a clear comparison between the meta-learning-based model formulation of PanPep and existing peptide–TCR prediction tools in terms of model training and testing in three different settings is summarized in Supplementary Table 1.

Meta-learning module. In the meta-learning module, we adopted the MAML framework[28] that can be adapted to the peptide-specific task distribution $p(T)$. This framework does not make an assumption on the form of the model, other than to assume that the model is parameterized by a parameter vector θ. Specifically, a model can be considered a function f_θ with parameter θ. A meta-learner aims to learn a parameter θ_{meta} with high-order meta information that can be rapidly adapted to a new peptide-specific task from $p(T)$. When the meta-learner is adapted to a new task T_i the meta-learner parameter θ_{meta} will be updated into peptide-specific learner parameter θ_i by a few gradient descent updates on task T_i. The parameter updates based on the support set for each task T_i will be called **inner loops**. In the present study, we set the inner loop as 3 and use the cross-entropy $\mathcal{L}$ as the loss function. Therefore, the updates of parameter θ_i of the peptide-specific learner on T_i are formulated as follows:

$$\theta_i^{(1)} = \theta_{\mathrm{meta}} - \ell \cdot \nabla_{\theta_{\mathrm{meta}}} E_{\mathrm{tcr}_j', y_j' \sim S_i}\left[\mathcal{L}\left[f_{\theta_{meta}}\left(\mathrm{TE}_i, \mathrm{tcr}_j'\right), y_j'\right]\right] \quad (1)$$

$$\theta_i^{(2)} = \theta_i^{(1)} - \ell \cdot \nabla_{\theta_i^{(1)}} E_{\mathrm{tcr}_j', y_j' \sim S_i}\left[\mathcal{L}\left[f_{\theta_i^{(1)}}\left(\mathrm{TE}_i, \mathrm{tcr}_j'\right), y_j'\right]\right] \quad (2)$$

$$\theta_i^{(3)} = \theta_i^{(2)} - \ell \cdot \nabla_{\theta_i^{(2)}} E_{\mathrm{tcr}_j', y_j' \sim S_i}\left[\mathcal{L}\left[f_{\theta_i^{(2)}}\left(\mathrm{TE}_i, \mathrm{tcr}_j'\right), y_j'\right]\right] \quad (3)$$

where $\theta_i^{(3)}$ represents the parameter of the peptide-specific learner after three inner loops on task T_i, ℓ is the learning rate of the inner loop and S_i denotes the support set of task T_i. TE_i is the peptide-specific task embedding, which also represents the encoding of peptide i. The peptide-specific learner uses $(\mathrm{TE}_i, \mathrm{tcr}_j')$ as input, which represents the pair of peptide encoding of task T_i and encoding of TCR j. y_j' is the label ndicating whether tcr_j' binds to peptide i.

The meta-learner parameter θ_{meta} is trained by optimizing the performance of f_{θ_i} on query set Q_i with respect to θ_{meta} across tasks from $p(T)$. Specifically, the objective of the meta-learner is to optimize the parameter θ_{meta} such that a few gradient steps (inner loops) in a new task will produce the most efficient behaviour for that task. The meta-learner parameter updates across tasks are called **outer loops**. In this study, we adopted the Adam optimizer to perform the meta-learner optimization, and the peptide-specific task batch size is 1; therefore, the meta-learner parameter θ_{meta} at one step is updated as follows:

$$\theta_{\mathrm{meta}} \leftarrow \theta_{\mathrm{meta}} - \ell' \cdot \nabla_{\theta_{\mathrm{meta}}} E_{\mathrm{tcr}_j, y_j \sim Q_i}\left[\mathcal{L}\left[f_{\theta_i^{(3)}}(\mathrm{TE}_i, \mathrm{tcr}_j), y_j\right]\right] \tag{4}$$

where ℓ' is the learning rate of the outer loop. tcr_j, y_j are the TCR j and label j indicating whether tcr_j in the query set binds to the peptide i in task T_i, respectively. Therefore, the updates of the meta-learner parameter involve the high-order derivative of the gradient. We also store the peptide-specific learners $f_{\theta_i^*}$ and their query sets Q_i^* for all training tasks after the convergence of the meta-learner for the next disentanglement distillation module.

Disentanglement distillation module. Although the model can rapidly adapt to the new tasks from $p(T)$ in the meta-learning module, the peptide-specific learner must be fine-tuned on the support set of task T_i. The peptide–TCR binding recognition of those exogenous or newborn peptides, which are defined as the unseen peptides in our study, represents the new tasks for which the support set is lacking for its task. We cannot obtain the peptide-specific learners directly for these tasks by fine-tuning the meta-learner since the support set is unavailable. This type of problem is very challenging and termed zero-shot learning in our study. NTM has been considered a perfect candidate for meta-learning and few-shot prediction, which extends an external memory for the model to rapidly encode and retrieve results. Although several studies have combined meta-learning with NTM[13,51,52], no proposed method generalizes NTM to the zero-shot prediction. We extend the basic idea of NTM and propose a disentanglement distillation module in PanPep that allows us to rapidly generate the peptide-specific learner for the unseen peptides. Similar to the architecture of the NTM, the disentanglement distillation module consists of a controller, memory, write head and read head, and detailed descriptions of these components used in the present study are provided below.

(1) The controller stores all peptide-specific learners $f_{\theta_i^*}$ and query sets Q_i^* on training tasks in the meta-learning module.
(2) Memory consists of basis I and content memory C, where basis is an identity matrix $I^{L\times L}$ and content memory is a matrix $C^{L\times V}$. V is the length of the parameter vector θ_i^* of the peptide-specific learner, and L is the dimension of the new peptide embedding space, termed the PLGS.
(3) Write head is the disentanglement operation, which is a learnable matrix $W^{N\times L}$, where N is the number of peptide-specific learners on training tasks.
(4) The read head is a matrix $A^{E\times L}$ that maps the peptide-specific task embedding TE_i to the new peptide embedding space (PLGS) as TE_i', where E is the length of the TE_i.

We constructed the mapping between peptide embedding and peptide-specific learners by first writing all peptide-specific learners into the content memory, which is formulated as follows:

$$M \leftarrow (F^T W)^T \tag{5}$$

where F is the matrix consisting of peptide-specific learner parameters in the controller, that is, $F^{N\times V} = \left[f_{\theta_1^*}, f_{\theta_2^*}, ..., f_{\theta_N^*}\right]^T$.

Then, a new peptide-specific learner $f_{\theta_i'}'$ is generated by retrieving the memory. Considering the order-irrelevant peptide-specific tasks, we adopt the content-based memory address mechanism in NTM. Specifically, the read head uses each peptide-specific task embedding TE_i as input and outputs a new peptide embedding TE_i' in the PLGS, which is used to calculate the similarity with each vector in the basis I. In the present study, the similarity function $K(u, v)$ is set as cosine similarity:

$$K(u, v) = \frac{u \cdot v}{||u||||v||} \tag{6}$$

which is used to produce a read-attention weight $w(m)$ on each vector $I(m)$ in the basis I, with elements computed based on softmax:

$$w(m) = \frac{\exp\left(K\left(\mathrm{TE}_i', I(m)\right)\right)}{\sum_n \exp\left(K\left(\mathrm{TE}_i', I(n)\right)\right)} \tag{7}$$

such that $\sum_m w(m) = 1$, and a new peptide-specific learner is retrieved by read-attention weight vector from content memory M:

$$f_{\theta_i'}' \leftarrow \sum_m w(m) M(m) \tag{8}$$

where $M(m)$ represents the mth row parameter vector in M.

Finally, inspired by the idea of **learning without forgetting** (LwF)[53], which uses the distil loss as the external term for the loss of the next new task, we extended this basic idea to our model disentanglement process. In the meta-learning module, we store the peptide-specific learners $f_{\theta_i^*}$ and their query sets Q_i for all the peptide-specific training tasks. After conducting all training tasks, all the peptide-specific learners are written into the memory using the write head W. Read head A takes the peptide embedding TE_i and maps it into the PLGS, subsequently generating a new peptide-specific learner $f_{\theta_i'}'$ by retrieving the content memory. Then, we propose using the distil loss $\mathcal{L}_d$ to update the write head W and read head Q as described below.

$$\begin{aligned}\mathcal{L}_d = -E_{\mathrm{TE}_i\sim p(T)}\Bigg[\sum_{\mathrm{tcr}_j\sim Q_i}\Big[f_{\theta_i^*}^{+}(\mathrm{TE}_i, \mathrm{tcr}_j) \times \log\left(f_{\theta_i'}'^{+}(\mathrm{TE}_i, \mathrm{tcr}_j)\right) \\ + f_{\theta_i^*}^{-}(\mathrm{TE}_i, \mathrm{tcr}_j) \times \log\left(f_{\theta_i'}'^{-}(\mathrm{TE}_i, \mathrm{tcr}_j)\right)\Big]\Bigg]\end{aligned} \tag{9}$$

where TE_i is the embedding of peptide-specific task T_i from $p(T)$. tcr_j belongs to the query set Q_i for peptide-specific task T_i. f^+ and f^- represent the probability output of model $f_{\theta_i^*}$ or $f_{\theta_i'}'$ for tcr_j binding and not binding to peptide i, respectively.

Detailed meta-learner architecture of PanPep. PanPep uses the peptide–TCR pair as input and outputs the binding probability of this pair (Supplementary Fig. 2b). In the current study, PanPep does not consider the HLA typing information and the alpha chain of TCR due to limited available data (see Discussion and Supplementary Methods). Each peptide–TCR pair is first encoded by the Atchley factor into a 40 × 5 matrix denoted as the peptide–TCR matrix, of which the peptide is padded into a 15 × 5 matrix by a zero vector and the TCR CDR3 sequence is padded into 25 × 5 matrix. Each amino acid was featured as five numerical

values, which represented its biochemical characteristics. We captured the interaction of the peptide and TCR by adopting the self-attention mechanism to the peptide–TCR matrix, and we used the sinusoidal position encoding method to add position information to each amino acid for the peptide and TCR. The peptide–TCR matrix containing position information is then used as input to the self-attention layer, which is a powerful architecture that consists of *Q*, *K* and *V* matrices and has been widely used in natural language processing to capture the relationship between words. We used a one-head self-attention mechanism[54], and the *Q*, *K* and *V* matrices with a size of 5 × 5 were applied here. The output of the self-attention layer was then fed to a 5 × 5 linear transformation layer and activated with the ReLU function. Next, the output was the same size as the original peptide–TCR matrix, while the embedding of each amino acid was added to the information from its context. We further used the convolutional layer[55] with sixteen 2 × 1 kernels and the ReLU activation function[56] to extract the latent information, followed by a batch-normalization layer and a max-pooling layer with a 2 × 2 kernel. After the pooling layer, the output was flattened and fed into a two-neuron dense layer. Finally, the probability of binding was calculated using softmax.

Dynamic sampling. In the training process of the meta-learning module, we must construct a peptide episode for each peptide-specific task in each outer loop. Since the number of known binding TCRs is different for those peptides and the data distribution obeys the long-tail distribution, constructing the peptide episode for each peptide once will lose considerable information for the peptides with a large TCR binding repertoire. Therefore, we proposed a dynamic sampling method during the meta-learner training process; namely, the support set and query set for each peptide will be reselected randomly in each training epoch. This method helps to mitigate the effect of an unbalanced data distribution on model training[57], and it can reduce the randomness when evaluating the performance in the few-shot setting as well.

Data curation

Several datasets are curated in our study, including the base dataset, c**ontrol** TCR set, meta-dataset and majority dataset. We described the detailed data curation procedures for these datasets.

We first collected comprehensive peptide–TCR binding records from four databases, including IEDB[46], VDJdb[58], PIRD[59] and McPas-TCR[60], for the exploratory study, and this dataset is termed the base dataset. We used the filtering criteria described below to curate the data. We first retained the records for TCRs from *Homo sapiens*. Then, only the records where the description of HLA alleles belong to HLA-I were retained. In addition, we excluded those records in VDJdb where the confidence scores are equal to 0 indicating that there were no insufficient method details or evidence to support the binding conclusions. Also only the high-confidence binding TCRs in the PRID dataset were retained. Furthermore, we retained only the records containing the CDR3 beta chain due to the importance of the CDR3 beta chain in peptide–TCR binding. We did not consider CDR3 alpha chain in the current study due to the limited availability of the data (Supplementary Methods). As a result, the base dataset contains 699 unique peptides and 29,467 unique TCRs with 32,080 related peptide–TCR binding pairs considering the cross-reactivity of peptide–TCR binding.

We then designed the **control TCR set** to contain nonbinding TCRs for each peptide-specific task. Since one TCR sequence might bind to different peptides with cross-reactivity and the meta-learning module of PanPep is highly sensitive to the data quality, we must reduce the bias from the mislabels among the nonbinding TCRs as much as possible. Therefore, although the negative mismatched TCRs were used as the negative controls in several previous studies, we avoided using them as the control because this control set is biased and may contain TCRs binding to the given peptide through cross-reactivity. In our study, the nonbinding TCR repertoire of the control TCR set was collected from 587 healthy volunteers' peripheral blood using a multiplexed polymerase chain reaction assay that targets the variable region of the rearranged TCRβ locus, which contains 60,333,379 TCRs as the repertoire that was not activated, as reported and adopted in previous studies[61,62]. Considering the large number of TCRs in this healthy repertoire, randomly sampling a part of TCRs from this large pool as a control set has a very low probability of encountering TCRs binding to the given peptide.

We curated a meta-dataset based on the base dataset where peptides with at least five binding TCR records were selected to train the meta-learner and obtain peptide-specific learner in PanPep. In total, the meta-dataset consists of 208 different peptides with 31,223 peptide–TCR binding pairs. We randomly split the meta-dataset into a meta-training dataset (4/5 of the peptide-specific tasks) and a meta-testing dataset (1/5 of the peptide-specific tasks) to perform the fivefold cross-validation. We must construct a peptide episode for each peptide in the meta-training dataset, which consists of a support set of two known binding TCRs and a query set of three known binding TCRs, to train the meta-learner in a peptide-specific manner. All the binding TCRs were balanced by the control TCR set. Similarly, we also constructed the peptide episode for each peptide in the meta-testing dataset. For each episode, we used the support set to fine-tune the meta-learner, obtaining the peptide-specific learner, and it was tested in the query set.

In the zero-shot setting, we want to evaluate the generalizability of PanPep to unseen peptides. Therefore, we chose the remaining peptides in the base dataset, which excluded the peptides from meta-dataset, resulting in a zero-shot dataset consisting of 491 unique peptides with 857 known binding TCR pairs. These peptides were not used in the PanPep training process. For each peptide in zero-shot dataset, all the known binding TCRs are included in the query set, and no TCRs are available in the support set for the fine-tuning of the model. Then, all the binding TCRs were balanced by the control TCR set.

We merged the datasets from the existing peptide–TCR binding prediction tools, including pMTnet, ERGO2 and DLpTCR, in the majority setting to illustrate that PanPep can also be easily generalized to the majority setting. For testing the ability of PanPep in generalization to peptides with a large number of TCR, we constructed the **majority dataset** based on the peptides whose known number of binding TCRs is more than 100 in the training datasets of these tools. The majority dataset consisting of 25 peptides with 23,232 known peptide–TCR binding pairs used for training and 5,230 peptide–TCR pairs used for testing. For each peptide in the majority setting, we constructed a peptide episode that consists of a support set and a query set. The support set contains all its corresponding TCRs in the training datasets for these tools, and the query set contains all corresponding TCRs in its own testing dataset. The support set of each peptide episode was used to retrain the meta-learner in PanPep and obtain the peptide-specific learner. Then, the query set of each peptide episode was used to evaluate the performance of PanPep in this majority setting.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All datasets used for training and testing PanPep in three settings and predicted clone expansion result are shared on the github repository (https://github.com/bm2-lab/PanPep) and its Zenodo (https://doi.org/10.5281/zenodo.7544387)[63]. The control TCR set where TCRs were sequenced from 587 healthy individuals can be obtained from original study[61] (https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-015-0238-z) and Zenodo (https://doi.org/10.5281/zenodo.7544387)[63]. The datasets used for training and testing were integrated from IEDB[46] (http://www.iedb.org/), VDJdb[58] (https://vdjdb.

cdr3.net/), PIRD[59] (https://db.cngb.org/pird/) and McPas-TCR[60] (http://friedmanlab.weizmann.ac.il/McPAS-TCR/). The detailed information of the 10x Genomics cohort is available at: https://www.10xgenomics.com/resources/datasets. The data used to test the ability of PanPep for immune-responsive T-cell sorting in neoantigen screening were downloaded from original study[38] (https://doi.org/10.1126/science.aad1253). The single-cell gene expression data used to identify neoantigen-reactive T-cell signatures for PanPep is available at: https://doi.org/10.17632/93cvs2z3mz.2. The published large cohort COVID-19 dataset is available at https://clients.adaptivebiotech.com/pub/covid-2020. The collected 3D crystal complexes are available at PDB (https://www.rcsb.org) and their accession numbers were provided in Supplementary Data 8.

Code availability

PanPep is available on github (https://github.com/bm2-lab/PanPep) and its Zenodo (https://doi.org/10.5281/zenodo.7544387)[63], together with a usage documentation and several example testing datasets.

References

1. Waldman, A. D., Fritz, J. M. & Lenardo, M. J. A guide to cancer immunotherapy: from T cell basic science to clinical practice. *Nat. Rev. Immunol.* **20**, 651–668 (2020).
2. Schumacher, T. N. & Schreiber, R. D. Neoantigens in cancer immunotherapy. *Science* **348**, 69–74 (2015).
3. Linette, G. P. & Carreno, B. M. Neoantigen vaccines pass the immunogenicity test. *Trends Mol. Med.* **23**, 869–871 (2017).
4. Ott, P. A. et al. An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature* **547**, 217–221 (2017).
5. Altman, J. D. et al. Phenotypic analysis of antigen-specific T lymphocytes. *Science* **274**, 94–96 (1996).
6. Zhang, S.-Q. et al. High-throughput determination of the antigen specificities of T cell receptors in single cells. *Nat. Biotechnol.* **36**, 1156–1159 (2018).
7. Kula, T. et al. T-Scan: a genome-wide method for the systematic discovery of T cell epitopes. *Cell* **178**, 1016–1028.e13 (2019).
8. Lu, T. et al. Deep learning-based prediction of the T cell receptor–antigen binding specificity. *Nat. Mach. Intell.* **3**, 864–875 (2021).
9. Glanville, J. et al. Identifying specificity groups in the T cell receptor repertoire. *Nature* **547**, 94–98 (2017).
10. Huang, H., Wang, C., Rubelt, F., Scriba, T. J. & Davis, M. M. Analyzing the *Mycobacterium tuberculosis* immune response by T-cell receptor clustering with GLIPH2 and genome-wide antigen screening. *Nat. Biotechnol.* **38**, 1194–1202 (2020).
11. Dvorkin, S., Levi, R. & Louzoun, Y. Autoencoder based local T cell repertoire density can be used to classify samples and T cell receptors. *PLoS Comput. Biol.* **17**, e1009225 (2021).
12. Jokinen, E., Huuhtanen, J., Mustjoki, S., Heinonen, M. & Lähdesmäki, H. Predicting recognition between T cell receptors and epitopes with TCRGP. *PLoS Comput. Biol.* **17**, e1008814 (2021).
13. Gielis, S. et al. Detection of enriched T cell epitope specificity in full T cell receptor sequence repertoires. *Front. Immunol.* **10**, 2820 (2019).
14. Montemurro, A. et al. NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. *Commun. Biol.* **4**, 1060 (2021).
15. Xu, Z. et al. DLpTCR: an ensemble deep learning framework for predicting immunogenic peptide recognized by T cell receptor. *Brief. Bioinform.* **22**, bbab335 (2021).
16. 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.* **11**, 1803 (2020).
17. Weber, A., Born, J. & Rodriguez Martínez, M. TITAN: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics* **37**, i237–i244 (2021).
18. Springer, I., Tickotsky, N. & Louzoun, Y. Contribution of T cell receptor alpha and beta CDR3, MHC typing, V and J genes to peptide binding prediction. *Front. Immunol.* **12**, 664514 (2021).
19. Reddy, S. T. The patterns of T-cell target recognition. *Nature* **547**, 36–38 (2017).
20. Moris, P. et al. Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. *Brief. Bioinform.* **22**, bbaa318 (2021).
21. Rosenberg, S. A. & Restifo, N. P. Adoptive cell transfer as personalized immunotherapy for human cancer. *Science* **348**, 62–68 (2015).
22. Dash, P. et al. Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* **547**, 89–93 (2017).
23. Sidhom, J.-W., Larman, H. B., Pardoll, D. M. & Baras, A. S. DeepTCR is a deep learning framework for revealing sequence concepts within T-cell repertoires. *Nat. Commun.* **12**, 1605 (2021).
24. Zhang, H., Zhan, X. & Li, B. GIANA allows computationally-efficient TCR clustering and multi-disease repertoire classification by isometric transformation. *Nat. Commun.* **12**, 4699 (2021).
25. Zhang, H. et al. Investigation of antigen-specific T-cell receptor clusters in human cancers. *Clin. Cancer Res.* **26**, 1359–1371 (2020).
26. Donovan, L. K. & Taylor, M. D. Amplifying natural antitumor immunity for personalized immunotherapy. *Cell Res.* **32**, 505–506 (2022).
27. Kiyotani, K., Toyoshima, Y. & Nakamura, Y. Immunogenomics in personalized cancer treatments. *J. Hum. Genet.* **66**, 901–907 (2021).
28. Finn, C., Abbeel, P. & Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. 34th International Conference on Machine Learning* (Eds Precup, D. & Teh, Y. W.) 1126–1135 (JMLR.org, 2017).
29. Brbić, M. et al. MARS: discovering novel cell types across heterogeneous single-cell experiments. *Nat. Methods* **17**, 1200–1206 (2020).
30. Rusu, A. A. et al. *7th International Conference on Learning Representations* (OpenReview.net, 2019).
31. Antoniou, A., Edwards, H. & Storkey, A. *7th International Conference on Learning Representations* (OpenReview.net, 2019).
32. Graves, A., Wayne, G. & Danihelka, I. Neural turing machines. Preprint at https://arxiv.org/abs/1410.5401 (2014).
33. Wang, Y., Yao, Q., Kwok, J. T. & Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* **53**, 1–34 (2020).
34. Wang, W., Zheng, V. W., Yu, H. & Miao, C. A survey of zero-shot learning: Settings, methods, and applications. *ACM Trans. Intell. Syst. Technol.* **10**, 1–37 (2019).
35. Huang, H. et al. Select sequencing of clonally expanded CD8 + T cells reveals limits to clonal expansion. *Proc. Natl Acad. Sci. USA* **116**, 8995–9001 (2019).
36. Klebanoff, C. A., Khong, H. T., Antony, P. A., Palmer, D. C. & Restifo, N. P. Sinks, suppressors and antigen presenters: how lymphodepletion enhances T cell-mediated tumor immunotherapy. *Trends Immunol.* **26**, 111–117 (2005).
37. Pogorelyy, M. V. et al. Precise tracking of vaccine-responding T cell clones reveals convergent and personalized response in identical twins. *Proc. Natl Acad. Sci. USA* **115**, 12704–12709 (2018).
38. Tran, E. et al. Immunogenicity of somatic mutations in human gastrointestinal cancers. *Science* **350**, 1387–1390 (2015).
39. Zheng, C. et al. Transcriptomic profiles of neoantigen-reactive T cells in human gastrointestinal cancers. *Cancer Cell* **40**, 410–423.e417 (2022).
40. Wang, D. The essential role of G protein-coupled receptor (GPCR) signaling in regulating T cell immunity. *Immunopharmacol. Immunotoxicol.* **40**, 187–192 (2018).

41. Lämmermann, T. & Kastenmüller, W. Concepts of GPCR-controlled navigation in the immune system. *Immunol. Rev.* **289**, 205–231 (2019).
42. Cantrell, D. T cell antigen receptor signal transduction pathways. *Annu. Rev. Immunol.* **14**, 259–274 (1996).
43. May, D. H. et al. Immunosequencing and epitope mapping reveal substantial preservation of the T cell immune response to Omicron generated by SARS-CoV-2 vaccines. Preprint at *medRxiv* https://doi.org/10.1101/2021.12.20.21267877 (2021).
44. Nolan, S. et al. A large-scale database of T-cell receptor beta (TCRβ) sequences and binding associations from natural and synthetic exposure to SARS-CoV-2. Preprint at *Research Square* https://doi.org/10.21203/rs.3.rs-51964/v1 (2020).
45. Weiss, G. A., Watanabe, C. K., Zhong, A., Goddard, A. & Sidhu, S. S. Rapid mapping of protein functional epitopes by combinatorial alanine scanning. *Proc. Natl Acad. Sci. USA* **97**, 8950–8954 (2000).
46. Vita, R. et al. The immune epitope database (IEDB): 2018 update. *Nucleic Acids Res.* **47**, D339–D343 (2019).
47. Shimizu, A. et al. Structure of TCR and antigen complexes at an immunodominant CTL epitope in HIV-1 infection. *Sci. Rep.* **3**, 3097 (2013).
48. Tirosh, I. et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).
49. Zhao, X. et al. Tuning T cell receptor sensitivity through catch bond engineering. *Science* **376**, eabl5282 (2022).
50. Huang, D. L., Bax, N. A., Buckley, C. D., Weis, W. I. & Dunn, A. R. Vinculin forms a directionally asymmetric catch bond with F-actin. *Science* **357**, 703–706 (2017).
51. Munkhdalai, T. & Yu, H. Meta networks. In *Proc. 34th International Conference on Machine Learning* (Eds Precup, D. & Teh, Y. W.) 2554–2563 (JMLR.org, 2017).
52. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D. & Lillicrap, T. Meta-learning with memory-augmented neural networks. In *Proc. 33rd International Conference on International Conference on Machine Learning* (Eds Balcan, M. F. & Weinberger, K. Q.) 1842–1850 (JMLR.org, 2016).
53. Li, Z. & Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**, 2935–2947 (2017).
54. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).
55. Shin, H.-C. et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **35**, 1285–1298 (2016).
56. Agarap, A. F. Deep learning using rectified linear units (relu). Preprint at https://arxiv.org/abs/1803.08375 (2019).
57. Menon, A. K. et al. *9th International Conference on Learning Representations* (OpenReview.net, 2021).
58. Bagaev, D. V. et al. VDJdb in 2019: database extension, new analysis infrastructure and a T-cell receptor motif compendium. *Nucleic Acids Res.* **48**, D1057–D1062 (2020).
59. Zhang, W. et al. PIRD: Pan immune repertoire database. *Bioinformatics* **36**, 897–903 (2020).
60. Tickotsky, N., Sagiv, T., Prilusky, J., Shifrut, E. & Friedman, N. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics* **33**, 2924–2929 (2017).
61. Dean, J. et al. Annotation of pseudogenic gene segments by massively parallel sequencing of rearranged lymphocyte receptor loci. *Genome Med.* **7**, 123 (2015).
62. Luu, A. M., Leistico, J. R., Miller, T., Kim, S. & Song, J. S. Predicting TCR-epitope binding specificity using deep metric learning and multimodal learning. *Genes* **12**, 572 (2021).
63. Gao, Y., Gao, Y. & Liu, Q. Pan-Peptide Meta Learning for T-Cell Receptor-Antigen Binding Recognition (v1.0.0). *Zenodo* https://doi.org/10.5281/zenodo.7544387 (2023).

Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No. 2021YFF1201200, No. 2021YFF1200900, received by Q.L.), National Natural Science Foundation of China (Grant No. 31970638, 61572361, received by Q.L.), Shanghai Natural Science Foundation Program (Grant No. 17ZR1449400, received by Q.L.), Shanghai Artificial Intelligence Technology Standard Project (Grant No. 19DZ2200900, received by Q.L.), Shanghai Shuguang scholars project (received by Q.L.), Shanghai excellent academic leader project (received by Q.L.), WeBank scholars project (received by Q.L.) and Fundamental Research Funds for the Central Universities (received by Q.L.).

Author contributions

Q.L., Yicheng Gao and Yuli Gao designed the framework of this work. Yicheng Gao, Yuli Gao, Y.F., C. Zhu, Z.W., C. Zhou, G.C., Q.C. and H.Z. performed the analyses. Yicheng Gao, Yuli Gao and Q.L. wrote the paper with the help of other authors. All authors read and approved the final paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00619-3.

Correspondence and requests for materials should be addressed to Qi Liu.

Peer review information *Nature Machine Intelligence* thanks Dong Xu and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature research

Corresponding author(s): Qi Liu

Last updated by author(s): Jan 19, 2023

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☐	☒	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	No software was used for data collection.
Data analysis	We built and used the model 'PanPep' for the downstream data analysis. The source code developed for this study is available on Github: https://github.com/bm2-lab/PanPep (DOI: https://doi.org/10.5281/zenodo.7544387). The peptide embedding clustering was computed by the DBSCAN algorithm from the scikit-learn package (v1.0.2), implemented by Python (v3.9.7).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

All datasets used for training and testing PanPep in three settings and predicted clone expansion result are shared on the github repository (https://github.com/bm2-lab/PanPep) and its Zenodo (https://doi.org/10.5281/zenodo.7544387) .
The control TCR set where TCRs were sequenced from 587 healthy individuals can be obtained from original study (https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-015-0238-z) and Zenodo (https://doi.org/10.5281/zenodo.7544387) .
The datasets used for training and testing were integrated from IEDB (http://www.iedb.org/), VDJdb (https://vdjdb.cdr3.net/), PIRD (https://db.cngb.org/pird/) and McPas-TCR (http://friedmanlab.weizmann.ac.il/McPAS-TCR/).

The detailed information of the 10X Genomic cohort is available at: https://www.10xgenomics.com/resources/datasets.
The data used to test the ability of PanPep for immune-responsive T cell sorting in neoantigen screening were downloaded from original study (https://www.science.org/doi/10.1126/science.aad1253).
The single cell gene expression data used to identify neoantigen-reactive T-cell signatures for PanPep is available at: https://doi.org/10.17632/93cvs2z3mz.2.
The published large cohort COVID-19 dataset is available at https://clients.adaptivebiotech.com/pub/covid-2020.
The collected 3D-crystal complexes are available at PDB (https://www.rcsb.org) and their accession numbers were provided in Supplementary Data 8.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	In this work, we merged a base dataset from IEDB, VDJdb, PRID, and McPas-TCR, resulting in a total of unique peptides and 29467 unique TCRs with 32080 related peptide-TCR binding pairs. Furthermore, meta dataset consists of 208 different peptides with 31223 peptide-TCR binding pairs. Zero-shot dataset consisting of 491 unique peptides and 857 known binding TCR pairs. Majority dataset consisting of 25 peptides with 23232 known peptide-TCR binding pairs used for training and 5230 peptides used for testing. Considering the limited available data at present, our model was built by combining the concepts of the meta learning and Neural Turing Machine (NTM), for addressing the few-shot learning and zero-shot learning problem in immunology, so these datasets are sufficient for the analysis.
Data exclusions	In the data preprocessing step of COVID-19 dataset, we filtered the TCR sequences with unproductive and "*". There is no data exclusion in other dataset in this study.
Replication	We ran the example python code to confirm our results are reproducible.
Randomization	Randomly drawn disjoint training and testing folds were used to evaluate our model. In this study, we use 5-fold cross-validation to evaluate PanPep.
Blinding	In this experiment, the investigators understand group allocation and the investigators were blinded to group allocation.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging