Multi-Task Training Increases Native Sequence Recovery of Antigen-Specific T-cell Receptor Sequences

Anonymous Authors[1]

Abstract

T-cells are a critical component of the adaptive immune system that use T-cell receptors (TCRs) to bind highly specific non-self peptide fragments presented by major histocompatibility complex (MHC) molecules on the surface of other cells. Given their importance, a foundation model of TCR specificity that is capable of reliably mapping between TCR sequences and their cognate peptide-MHC (pMHC) ligands remains an unmet need. This study presents a key step towards developing a comprehensive foundation model by exploring the bi-directional mapping of both pMHCs to their corresponding TCRs, and vice versa. While validation performance was significantly worse in the TCR to pMHC direction given the highly asymmetric distribution of pMHC data, we find that the bidirectionally trained model outperformed the model trained in a single pMHC to TCR direction. We present our findings as a potential direction towards a unified generative foundation model of TCR:pMHC cross-reactivity.

1. Introduction

The T-cell receptor (TCR) and peptide-MHC (pMHC) interaction is a fundamental immunological event that triggers our bodies' T-cell response against cancer, viruses, and even self-antigens. As such, there has been significant effort in developing a model of TCR specificity to map TCR sequences to their cognate epitopes, or design TCRs against antigens of interest (Hudson et al., 2023). Such a model would revolutionize cellular therapies (Cao et al., 2021; Tzannou et al., 2017; Ellebrecht et al., 2016; Poole et al., 2022) and our ability to contextualize the T-cell response against a specific pathogen. However, a foundation model that reliably goes between TCRs and their corresponding pMHC ligands remains blocked by issues of extreme data sparsity and noisy provenance coupled with a complicated cross-reactivity landscape (Wooldridge et al., 2011; Sewell, 2012).

In addition to efforts in the structural space (Ribeiro-Filho et al., 2024), recent work explored the explicit framing of the TCR reactivity problem as a sequence-to-sequence (seq2seq) task, demonstrating the capacity of sequence-based models to design TCR sequences against a specific pMHC (Karthikeyan et al., 2023; Fast et al., 2023). Sequence-to-sequence learning introduced the concept of an encoder:decoder model that was end-to-end trainable in the context of machine translation tasks (Sutskever et al., 2014; Kalchbrenner & Blunsom, 2013; Cho et al., 2014). However, this framework quickly became a paradigm-shifting method for training deep networks to maximize the conditional likelihood of target sequences given a source sequence for arbitrary source:target pairs such as question answering, text summarization, or even the TCR:pMHC.

A key consideration for encoder:decoder models is the amount of parallel data (source-target pairs) available, a requirement that scales with the complexity of the sequence mapping. In machine translation, a number of methods have been developed to specifically address the challenge maximizing the information usage of low-resource languages, when labeled data is limited and expensive to generate (Haddow et al., 2022). Techniques such as back-translation (Sennrich et al., 2016), self-training (He et al., 2020), transfer learning (Liu et al., 2020), and semi-supervised translation approaches that leverage monolingual data have all been proposed with varying degrees of success, dependent on the problem specifics. However these approaches have been shown to exacerbate overfitting to specific domain contexts or sequence distributions (Shen et al., 2020). To combat these effects, multi-task learning of bidirectional translation models and mulitilingual translation models have been shown to significantly improve translation quality by sharing representations and aligning latent spaces across multiple languages (Niu et al., 2018; Ding et al., 2021). In this work, we explore the use of multi-task training to train on both directions of the TCR:pMHC specificity problem and evaluate

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2024 Workshop on Accessible and Efficient Foundation Models for Biological Discovery. Do not distribute.

its effect on the conditional generation of TCR sequences as well as its potential to simultaneously sample accurate cognate pMHCs using the same weights.

2. Methodology

2.1. Setup

For our setup, we largely follow the same seq2seq problem formulation as laid out in (Karthikeyan et al., 2023). Briefly, we adopt the amino acid level tokenization scheme and restrict the sequence space to the high entropy portions of the TCR:pMHC interface, namely the CDR3b, peptide, and MHC pseudo-sequence as defined in (Hoof et al., 2009). In addition we use the BART and T5 encoder:decoder transformer architectures, with modified hyper-parameters to control for their total parameter count (Table 1). However, in order to introduce flexibility with the directionality, we leverage the concept of task prefixes introduced in (Raffel et al., 2020) as a special token for the T5 model architecture, using the BART model and its prefix-agnostic tokenization scheme for comparison. Additionally, we keep the use of the '[SEP]' token to separate the peptide, which typically spans 8-11 amino acids, and the fixed length psuedosequence.

BART:

[SOS]EPITOPE[SEP]PSEUDOSEQUENCE[EOS]

$\rightleftharpoons$

[SOS]CDR3SEQUENCE[EOS]

T5:

[PMHC]EPITOPE[SEP]PSEUDOSEQUENCE[EOS]

$\rightleftharpoons$

[TCR]CDR3SEQUENCE[EOS]

2.2. Dataset

For the generation of a parallel corpus, we used experimentally validated immunogenic TCR:pMHC pairs taken from publicly available databases (McPAS (Tickotsky et al., 2017), VDJdb (Shugay et al., 2017), and IEDB (Vita et al., 2018). Additionally, we used a large injection of weakly-labeled data derived from the MIRA (Dines et al., 2020)) which contained CDR3b and peptide sequences along with the HLA-type of the individual, instead of the presenting MHC allele. MHC allele was inferred using MHCFlurry2.0 (O'Donnell et al., 2020)'s ranked presentation score metric among the HLA-type. Of importance, these pseudo-synthetic examples were not used in evaluation. More on the dataset standardization procedures can be found in the Supplementary Methods A.1.1. The resulting dataset contains over 670,000 paired sequences (N=7834 pMHCs). In order to assess the capacity of the models to recapitulate the diversity of antigen-specific TCRs, we curated a target-rich dataset by separating out the top-20 most represented pMHCs for validation and trained on the remaining data, removing the occurrences of the held-out epitopes bound alternate MHCs. This resulted in a final dataset split of 580k training sequences (N=7745 pMHCs) and 93k validation sequences (N=20 pMHCs). A key limitation of this dataset is its highly skewed HLA distribution towards well studied alleles (A*02:01, A*03:01, A*11:01, etc).

2.3. Model Training

For our experiments, we considered three model training regimes. The baseline model was trained on the pMHC $\rightarrow$ TCR direction, a bidirectional model was trained on both directions, and finally a multi-task model was trained on both directions as well a masked language modeling objective for both TCR and pMHC sequences. All models were trained using the standard categorical cross entropy loss function (Equation 1), favored in seq2seq tasks for its desired effect of maximizing the conditional likelihoods over target sequences (Sutskever et al., 2014; Cho et al., 2014).

$$\begin{aligned}\mathcal{L} = CE(\mathbf{y}, \hat{\mathbf{y}}) &= -\sum_{i=1}^{n} \mathbf{y}_i \log \hat{\mathbf{y}}_i \\ &= -\sum_{i=1}^{n}\sum_{j-1}^{k} y_{ij} \log p_\theta(y_{ij}|\mathbf{x})\end{aligned} \tag{1}$$

The baseline models were trained using the above cross entropy objective for three epochs as training for longer epochs resulted in worse validation loss and accuracy. The bidirectional and multi-task models were trained on a mutli-term objective, comprising of a linear combination of individual loss terms corresponding to each task/direction. This was achieved using a batch processing algorithm (Algorithm 1), where each batch was rearranged into one of four sequence-to-sequence mapping possibilities and the model was trained on target reconstruction. For the bidirectional model this was straightforward as we could swap the input and output tensors during training to get the individual loss contributions of the $\mathcal{L}_{pmhc \rightarrow tcr}and\mathcal{L}_{tcr \rightarrow pmhc}$ (Equation 2). For the multi-task model, the mapping possibilities are: 1) pMHC $\rightarrow$ TCR 2) pMHC $\rightarrow$ TCR 3) Corrupted pMHC* $\rightarrow$ pMHC 4) Corrupted TCR* $\rightarrow$ TCR, which combine to to form $\mathcal{L}_{multi}$ (Equation 3). For the purposes of comparison against the baseline models, the bidirectional and multitask models were trained for 3 epochs per task/direction. As such the bidirectional model was trained for 6 epochs, and the multi-task model was trained for a total of 12 epochs.

$$\mathcal{L}_{bidxn} = \mathcal{L}_{pmhc \to tcr} + \mathcal{L}_{tcr \to pmhc} \quad (2)$$

$$\mathcal{L}_{multi} = \mathcal{L}_{MLM} + \mathcal{L}_{pmhc \to tcr} + \mathcal{L}_{tcr \to pmhc} \quad (3)$$

Algorithm 1 Multi-Task Training Step

Batched Input: source pMHCs: **X**, target TCRs: **Y**
Sample $a \sim Bernoulli(0.5)$
if $a > 0.5$ **then**
  Swap **X** and **Y**
  Compute attention masks
end if
Sample $b \sim Bernoulli(0.5)$
if $b > 0.5$ **then**
  Set **X** = **X*** and **Y** = **X**
  Compute attention masks
end if
do Predict $\hat{\mathbf{Y}} = \phi(\mathbf{X})$ and gradient updates on CE(**y**, $\hat{\mathbf{y}}$)

2.4. Conditional Sampling

Given its marked performance gains over other sampling methods in the TCR:pMHC sequence space (Karthikeyan et al., 2023), we fix beam-search as our chosen method of sequence generation, and use greedy decoding for calculating the BLEU-score. These methods, as well as the broader class of mode-seeking decoding methods, aim to maximize for the highest probability conditional sequence. Recently, however, mode-seeking algorithms have come under scrutiny for sampling only a small portion of the true target distribution, as noted in (Eikema & Aziz, 2020). Instead of sampling tokens directly from the whole conditional distribution: $y_t \sim P(y_t|y_{<t}, x, \theta)$, given source sequence x and model parameters θ, they try to approximate the $y^{MAP} = \arg\max_{y \in Y} \log p(y|x, \theta)$ which explores the conditional distribution about the mode. However, combined with the ability to assess and maintain longer high probability subsequences, the use of beam search results in a powerful method for sampling native sequence-like predictions.

2.5. Evaluation

To evaluate model performance in a holistic manner, we sought to assess both the fidelity of the conditional distributions learned by the model, as well as the decoding algorithm's ability to recapitulate the diversity of reference sequences using the following metrics:

- **Char-BLEU**: Character-level weighted n-gram precision calculated on the greedy decoding sequences against the $k = 20$ closest reference sequences. We use the NLTK's 'sentence_bleu' function to calculate a single translation's BLEU score and the 'corpus_bleu' function to compute the BLEU score over an entire dataset. Standard BLEU-4 metric was used for both (Papineni et al., 2002).

- **Perplexity**: Perplexity as a standard measure of language model performance, using the cross entropy loss calculated over the validation corpus.

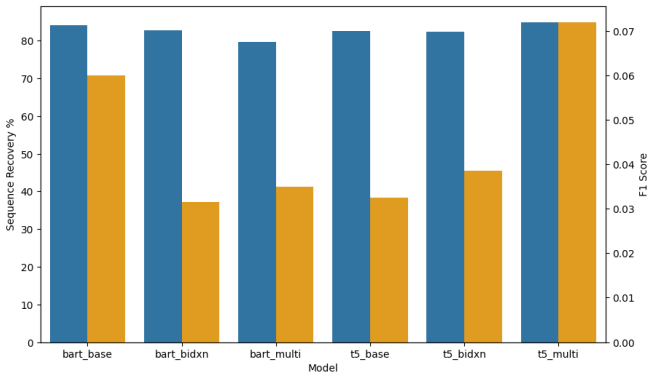
- **Precision, Recall, and F1@K**: Precision, recall, and F1 measures exact whole sequence recovery, computed after sampling K times (not K separate sequences).

- **Native Sequence Recovery**: In the *de novo* protein design space, native sequence recovery is a useful indicator of model performance. Here, for each model prediction, the index-matched exact amino acid recovery with the closest match of the same sequence length is calculated.

3. Results

As shown in Figure 1, we first evaluate the performance of the models trained in the various methods on TCR generations against the holdout dataset comprising the top-20 most well represented, *real* pMHCs. Of the models tested, the T5 multi-task model performed the best on all but one of the metrics with a character BLEU (CharBLEU) score of 99.1, a perplexity of 2.44, an F1 score of .072 and a sequence recovery rate of 85.1%. The second best model, which scored the highest on CharBLEU was the baseline BART model with a near perfect CharBLEU score of 99.8, a perplexity of 2.45, and F1 score of .06 and a sequence recovery rate of 84.2 %. Interestingly, we see opposite dynamics emerge between the BART and T5 models with the addition of auxiliary tasks. Whereas the T5 model sees an improvement in the native sequence recovery for the multi-task regime, the BART model's performance consistently decreases going from the baseline model to the bidirectional version, with the multi-task model performing the worst. This trend roughly holds true or for the CharBLEU and F1@100 as well. These results highlight the potential importance of the task prefix for boosting model performance in a relatively small number of epochs.

To understand and further characterize the performance on individual pMHCs and their contribution to the global performance of the models, we computed the same metrics and stratified them by pMHC (Table 3). For the sake of brevity, we report on just baseline and multi-task models. When independently evaluating the different pMHCs, we observe that the performances of the models are largely consistent across models within each pMHC. The pMHC EAAGIGILTV-A*02:01 was the worst performing pMHC



MODEL	CHARBLEU	PPL	F1@100	% REC.
BART	**99.8**	**2.45**	**.060**	**84.2**
BART (BD)	98.8	**2.45**	.032	82.8
BART (M)	98.9	2.49	.035	79.7
T5	96.2	2.58	.033	82.6
T5 (BD)	**99.2**	2.45	.039	82.3
T5 (M)	99.1	**2.44**	**.072**	**85.1**

Figure 1: *Average Model Performance across Top-20 Dataset* a) Multiple bar chart showing the native sequence recovery and F1@100 score of the benchmarked models. b) Table of values for percent sequence recovery and F1 as well as additional metrics including Char-BLEU and perplexity for the base, bidirectional, and multi-task BART and T5 models. Results computed using 100 sequences generated using the beam search algorithm with num beams set to 300.

of the 20, showing poor model performance on both sequence recovery and the F1 score across all models. In contrast, KLGGALQAK-A*03:01 and NLVPMVATV-A*02:01, among others, stand out as examples where all models performed well. Given the cross-model consistency in performance, we sought to better understand the 20 epitopes in the context of the metrics and the training data (Figure 2).

Since the metrics all implicitly or explicitly rely on held-out sequences, we first checked for a correlation between the number of TCRs for a held-out example and its performance. Unsurprisingly, we observed a weak positive trend between the number of reference CDR3b sequences and all performance metrics. Additionally, to better understand the generalization capacity of the models, we computed the edit distance to the closest training epitope and the CDR3b overlap between pairs, looking for correlations with the metrics. Here, the trends were far less pronounced ($R^2 \approx 0$).

Using a qualitative lens, we can begin to contextualize performance as marked by sparks or failures of both memorization and generalization. Take for example, the SARS-Cov-2 epitope YLQPRTFLL had the best performance by the T5 multi-task model and baseline BART model, with a sequence recovery of over 96% and an F1 score nearly 10x greater than the mean. This is likely explained by the presence of an epitope in the training set that shares 841 CDR3b sequences and has $d_edit = 1$. However, a low edit distance and high overlap do not guarantee outstanding performance, as evidenced by another SARS-Cov-2 epitope LLLDRLNQL. Remarkably, the second-best performing epitope was KLGGALQAK, with a >90% sequence recovery rate, despite the closest training sequence being six edits away and having an overlap of zero. While these results highlight the models' capability to generate antigen-specific sequences, they also underscore the importance of the training data's composition and its relationship to the test data. The mixed results suggest that both memorization and generalization play roles in model performance.

Finally, with regards to the TCR de-orphanization problem, we found that both bidirectional and multi-task models fared substantially worse. Interestingly, while the peptide-only CharBLEU scores was near 0, the models generated plausible-looking MHC pseudosequences. We believe that the disparity in performance between tasks is due to combination of lack of tuning for generating longer sequences as well as the asymmetric number of pMHCs than TCRs in the training data. As such we hold off on reporting the standard metrics for this iteration of our study. Instead we calculated the exact pseudo-sequence recovery (F1@1) score using beam search decoding as a litmus test for reverse direction performance. The average F1@1 scores were .02, .04, .13, and .29 for the BART (M), BART (BD), T5(M), and T5 (BD) models, respectively.

4. Discussion

In this study we set out to investigate the effect of multi-task training in improving the mapping quality between TCR and their corresponding pMHCs. We showed opposite dynamics between the BART and T5 models, rediscovering the importance of T5's prefix tokens. While the baseline BART model and multi-task T5 model showed similar performance in the TCR generation task, the T5 model proved to be superior when considering the TCR to pMHC direction. The performance of the T5 multi-task model in generating accurate TCR sequences suggests that jointly learning both directions of the TCR-pMHC sequence mapping, along with the unconditional distributions of both modalities, enhances model generalization and captures intricate sequence patterns. However, the observed variability in performance across different pMHC epitopes highlights the critical role of training data composition and its relationship to the test set. Future work will explore leveraging monolingual data for iterative back-translation and other techniques, such as label smoothing, to further improve model robustness and

applicability across diverse TCR-pMHC interactions.

References

Cao, X., Liu, G., Zhang, J., Zhao, Y., Chen, H., Zheng, H., Rui, W., Jia, L., Zhao, X., Lin, X., and Lu, P. A Novel CMV-Specific TCR-T Cell Therapy Is Effective and Safe for Refractory CMV Infection after Allogeneic Hematopoietic Stem Cell Transplantation. *Blood*, 138(Supplement 1):3848–3848, 11 2021. ISSN 0006-4971. doi: 10.1182/blood-2021-146446. URL https://doi.org/10.1182/blood-2021-146446.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

Dines, J. N., Manley, T. J., Svejnoha, E., Simmons, H. M., Taniguchi, R., Klinger, M., Baldo, L., and Robins, H. The immunerace study: A prospective multicohort study of immune response action to covid-19 events with the immunecode™ open access database. *medRxiv*, 2020. doi: 10.1101/2020.08.17.20175158. URL https://www.medrxiv.org/content/early/2020/08/21/2020.08.17.20175158.1.

Ding, L., Wu, D., and Tao, D. Improving neural machine translation by bidirectional training. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3278–3284, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.263. URL https://aclanthology.org/2021.emnlp-main.263.

Eikema, B. and Aziz, W. Is map decoding all you need? the inadequacy of the mode in neural machine translation, 2020.

Ellebrecht, C. T., Bhoj, V. G., Nace, A., Choi, E. J., Mao, X., Cho, M. J., Zenzo, G. D., Lanzavecchia, A., Seykora, J. T., Cotsarelis, G., Milone, M. C., and Payne, A. S. Reengineering chimeric antigen receptor t cells for targeted therapy of autoimmune disease. *Science*, 353(6295):179–184, 2016. doi: 10.1126/science.aaf6756. URL https://www.science.org/doi/abs/10.1126/science.aaf6756.

Fast, E., Dhar, M., and Chen, B. Tapir: a t-cell receptor language model for predicting rare and novel targets. *bioRxiv*, 2023. doi: 10.1101/2023.09.12.557285. URL https://www.biorxiv.org/content/early/2023/09/15/2023.09.12.557285.

Haddow, B., Bawden, R., Barone, A. V. M., Helcl, J., and Birch, A. Survey of Low-Resource Machine Translation. *Computational Linguistics*, 48(3):673–732, 09 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00446. URL https://doi.org/10.1162/coli_a_00446.

He, J., Gu, J., Shen, J., and Ranzato, M. Revisiting self-training for neural sequence generation, 2020.

Hoof, I., Peters, B., Sidney, J., Pedersen, L. E., Sette, A., Lund, O., Buus, S., and Nielsen, M. Netmhcpan, a method for mhc class i binding prediction beyond humans. *Immunogenetics*, 61:1–13, 2009.

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

Kalchbrenner, N. and Blunsom, P. Recurrent continuous translation models. In *Conference on Empirical Methods in Natural Language Processing*, 2013. URL https://api.semanticscholar.org/CorpusID:12639289.

Karthikeyan, D., Raffel, C., Vincent, B., and Rubinsteyn, A. Conditional generation of antigen specific t-cell receptor sequences. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. URL https://openreview.net/forum?id=SckdgVW3Kq.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. Multilingual denoising pre-training for neural machine translation, 2020.

Nagano, Y. and Chain, B. tidytcells: standardizer for tr/mh nomenclature. *Frontiers in Immunology*, 14, 2023. ISSN 1664-3224. doi: 10.3389/fimmu.2023.1276106. URL https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2023.1276106.

Niu, X., Denkowski, M., and Carpuat, M. Bi-directional neural machine translation with synthetic parallel data. In Birch, A., Finch, A., Luong, T., Neubig, G., and Oda, Y. (eds.), *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 84–91, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2710. URL https://aclanthology.org/W18-2710.

O'Donnell, T. J., Rubinsteyn, A., and Laserson, U. Mhcflurry 2.0: Improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing. *Cell Systems*, 11(1):42–48.e7, 2020. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2020.06.010. URL https://www.sciencedirect.com/science/article/pii/S2405471220302398.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

Poole, A., Karuppiah, V., Hartt, A., Haidar, J. N., Moureau, S., Dobrzycki, T., Hayes, C., Rowley, C., Dias, J., Harper, S., Barnbrook, K., Hock, M., Coles, C., Yang, W., Aleksic, M., Lin, A. B., Robinson, R., Dukes, J. D., Liddy, N., Van der Kamp, M., Plowman, G. D., Vuidepot, A., Cole, D. K., Whale, A. D., and Chillakuri, C. Therapeutic high affinity t cell receptor targeting a krasg12d cancer neoantigen. *Nature Communications*, 13(1):5333, Sep 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-32811-1. URL https://doi.org/10.1038/s41467-022-32811-1.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

Ribeiro-Filho, H. V., Jara, G. E., Guerra, J. V. S., Cheung, M., Felbinger, N. R., Pereira, J. G. C., Pierce, B. G., and de Oliveira, P. S. L. Exploring the potential of structure-based deep learning approaches for t cell receptor design. *bioRxiv*, 2024. doi: 10.1101/2024.04.19.590222. URL https://www.biorxiv.org/content/early/2024/04/24/2024.04.19.590222.

Sennrich, R., Haddow, B., and Birch, A. Improving neural machine translation models with monolingual data, 2016.

Sewell, A. Why must t cells be cross-reactive? *Nature reviews. Immunology*, 12:669–77, 08 2012. doi: 10.1038/nri3279.

Shen, J., Chen, P.-J., Le, M., He, J., Gu, J., Ott, M., Auli, M., and Ranzato, M. The source-target domain mismatch problem in machine translation, 2020.

Shugay, M., Bagaev, D. V., Zvyagin, I. V., Vroomans, R. M., Crawford, J. C., Dolton, G., Komech, E. A., Sycheva, A. L., Koneva, A. E., Egorov, E. S., Eliseev, A. V., Van Dyk, E., Dash, P., Attaf, M., Rius, C., Ladell, K., McLaren, J. E., Matthews, K. K., Clemens, E., Douek, D. C., Luciani, F., van Baarle, D., Kedzierska, K., Kesmir, C., Thomas, P. G., Price, D. A., Sewell, A. K., and Chudakov, D. M. VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Research*, 46(D1):D419–D427, 09 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx760. URL https://doi.org/10.1093/nar/gkx760.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks, 2014.

Tickotsky, N., Sagiv, T., Prilusky, J., Shifrut, E., and Friedman, N. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics*, 33(18):2924–2929, 05 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx286. URL https://doi.org/10.1093/bioinformatics/btx286.

Tzannou, I., Papadopoulou, A., Naik, S., Leung, K., Martinez, C. A., Ramos, C. A., Carrum, G., Sasa, G., Lulla, P., Watanabe, A., Kuvalekar, M., Gee, A. P., Wu, M.-F., Liu, H., Grilley, B. J., Krance, R. A., Gottschalk, S., Brenner, M. K., Rooney, C. M., Heslop, H. E., Leen, A. M., and Omer, B. Off-the-shelf virus-specific t cells to treat bk virus, human herpesvirus 6, cytomegalovirus, epstein-barr virus, and adenovirus infections after allogeneic hematopoietic stem-cell transplantation. *Journal of Clinical Oncology*, 35(31):3547–3557, 2017. doi: 10.1200/JCO.2017.73.0655. URL https://doi.org/10.1200/JCO.2017.73.0655. PMID: 28783452.

Vita, R., Mahajan, S., Overton, J. A., Dhanda, S. K., Martini, S., Cantrell, J. R., Wheeler, D. K., Sette, A., and Peters, B. The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Research*, 47(D1):D339–D343, 10 2018. ISSN 0305-1048. doi: 10.1093/nar/gky1006. URL https://doi.org/10.1093/nar/gky1006.

Wooldridge, E.-M., van den Berg, S., Miles, T., Dolton, C., Llewellyn-Lacey, P., and Peakman, S. A single autoimmune t cell receptor recognizes more than a million different peptides. *Journal of Biological Chemistry*, 287 (92):1168–1177, 2011. doi: 10.1074/jbc.M111.289488. URL https://www.sciencedirect.com/science/article/pii/S0167569998012997.

A. Appendix

A.1. Supplementary Methods

A.1.1. Data Processing

First, to aggregate the data spanning various sources, formats, and nomenclature, we mapped the columns from each individual dataset to a common consensus schema and concatenated the data along the consensus columns. In the interest of data retention, missing values were reasonably imputed according to other information for that data instance. To keep only the cytotoxic (CD8+) T-cells, we filtered the instances wherever the cell-type was provided or where the HLA-Allele was of MHC-class I. In cases where an HLA-haplotype was provided instead of the specific HLA-allele, as was the case for the MIRA data, we used MHCFlurry2.0 (O'Donnell et al., 2020) to predict the best presenting allele for the given epitope among the potential options from the haplotype information. This data augmentation step resulted in a 5-fold expansion of our training data. An key caveat to note is that the additional training examples are all derived from a single disease context (SARS-Cov-2), skewing the training data's distribution. Additionally there is room for slight error given the peptide-MHC assignment in-silico and not validated experimentally. However, given the merits of including examples with thousands of TCRs against an epitope, we argue for its inclusion. Where the granularity of the HLA-information or TR genes was at the serotype level, we inferred the canoncial gene/allele by starting off with the subgroup '*01' and incremented it until a matching IMGT gene was found. This step has the potential of introducing minor differences between the unknown ground truth and the imputed pseudo-sequence, as the pseudosequence is well conserved within serotype. Once the data was aggregated and values were imputed, we applied the following column-level standardization for each source of information:

- **Complementarity Determining Region (CDR3b), Epitope, and MHC Pseudo-Sequence**: All amino acid representations were normalized used the tidytcells.aa.standardise function found in the TidyTcells python package (Nagano & Chain, 2023).
- **TR Genes**: The TidyTcells package (Nagano & Chain, 2023) was once again used to standardize the nomenclature surrounding the T-Cell Receptor genes (e.g. TRB-V and TRB-J).
- **HLA-Allele**: HLA alleles were imputed where allele level information when necessary and then normalized using the MHCgnomes package to the standard HLA-[A,B,C]*XX:YY format.

A.2. Model/Training Hyperparameters

Table 1: Model Architecture Hyperparameters

	BART	T5
Parameters	46M	42M
d_{model}	768	256
Vocab Size	28	128
Encoder Layers	6	10
Decoder Layers	6	10
Max Position Embedding	512	512
Attention Heads	16	16
Feed Forward Dim	128	1024
Cross Attention	✓	✓

Table 2: Model Training Parameters

		BART	T5
Baseline	Epochs	3	3
	Batch Size	128	128
	Learning Rate	5e-05	3e-04
	Weight Decay	0.01	0.001
	Optimizer	AdamW	AdamW
Bidirectional Training	Epochs	6	6
	Batch Size	128	128
	Learning Rate	5e-05	3e-04
	Weight Decay	0.01	0.001
	Optimizer	AdamW	AdamW
Multitask Training	Epochs	12	12
	Batch Size	128	128
	Learning Rate	5e-05	3e-04
	Weight Decay	0.01	0.001
	p_{MLM}	0.15	0.15
	Optimizer	AdamW	AdamW

A.3. Extended Data Tables

Table 3: Individual Top-20 pMHC Performance Metrics

		Evaluation Metrics				
	Model	Char-BLEU	P@100	R@100	F1@100	% Recovery
AVFDRKSDAK-A*11:01 (n=1798, $d_{train} = 6$)	BART	0.96	0.00	0.00	0.00	84.3
	BART (M)	0.93	0.00	0.00	0.00	84.4
	T5	0.88	0.03	0.03	0.03	87.2
	T5 (M)	0.98	0.01	0.01	0.01	86.0
CRVLCCYVL-C*07:02 (n=497, $d_{train} = 6$)	BART	0.89	0.00	0.00	0.00	77.8
	BART (M)	0.82	0.00	0.00	0.00	73.0
	T5	0.77	0.00	0.00	0.00	76.4
	T5 (M)	0.83	0.00	0.00	0.00	77.0
EAAGIGILTV-A*02:01 (n=506, $d_{train} = 1$)	BART	0.85	0.00	0.00	0.00	67.2
	BART (M)	0.73	0.00	0.00	0.00	64.7
	T5	0.75	0.00	0.00	0.00	59.8
	T5 (M)	0.81	0.00	0.00	0.00	65.3
ELAGIGILTV-A*02:01 (n=2035, $d_{train} = 1$)	BART	0.92	0.00	0.00	0.00	84.3
	BART (M)	1.00	0.01	0.1	0.01	84.4
	T5	0.93	0.00	0.00	0.00	77.6
	T5 (M)	1.00	0.02	0.02	0.02	84.2
GILGFVFTL-A*02:01 (n=11619, $d_{train} = 2$)	BART	0.91	0.02	0.02	0.02	84.9
	BART (M)	1.00	0.03	0.03	0.03	86.3
	T5	0.93	0.01	0.01	0.01	83.2
	T5 (M)	0.93	0.02	0.02	0.02	83.9
GLCTLVAML-A*02:01 (n=12254, $d_{train} = 6$)	BART	1.00	0.02	0.02	0.02	86.3
	BART (M)	1.00	0.04	0.04	0.04	86.9
	T5	0.93	0.01	0.01	0.01	81.8
	T5 (M)	1.00	0.01	0.01	0.01	85.3
IVTDFSVIK-A*11:01 (n=792, $d_{train} = 6$)	BART	0.90	0.0	0.0	0.0	81.4
	BART (M)	0.78	0.01	0.01	0.01	79.1
	T5	0.92	0.01	0.01	0.01	82.5
	T5 (M)	0.73	0.0	0.0	0.0	81.6
KLGGALQAK-A*03:01 (n=13937, $d_{train} = 6$)	BART	0.97	0.24	0.24	0.24	93.5
	BART (M)	1.00	0.10	0.10	0.10	90.6
	T5	0.92	0.14	0.14	0.14	92.7
	T5 (M)	1.00	0.14	0.14	0.14	92.7
LLLDRLNQL-A*02:01 (n=2151, $d_{train} = 1$)	BART	1.00	0.01	0.01	0.01	78.9
	BART (M)	0.87	0.0	0.0	0.0	79.3
	T5	0.78	0.01	0.01	0.01	78.0
	T5 (M)	0.87	0.01	0.01	0.01	79.8
LLWNGPMAV-A*02:01 (n=2870, $d_{train} = 4$)	BART	0.81	0.04	0.04	0.04	86.5
	BART (M)	0.89	0.06	0.06	0.06	87.4
	T5	0.92	0.08	0.08	0.08	87.4
	T5 (M)	0.95	0.05	0.05	0.05	87.3
LPRRSGAAGA-B*07:02	BART	0.97	0.03	0.03	0.03	86.2
	BART (M)	0.91	0.08	0.08	0.08	85.7

Table 3: (continued)

		Evaluation Metrics				
	Model	Char-BLEU	P@100	R@100	F1@100	% Recovery
(n=2299, $d_{train} = 5$)	T5	0.95	0.13	0.13	0.13	88.4
	T5 (M)	0.77	0.01	0.01	0.01	84.4
LVVDFSQFSR-A*11:01 (n=1906, $d_{train} = 7$)	BART	0.97	0.03	0.03	0.03	86.2
	BART (M)	1.00	0.02	0.02	0.02	85.1
	T5	1.00	0.01	0.01	0.01	84.2
	T5 (M)	0.97	0.01	0.01	0.01	86.2
NLVPMVATV-A*02:01 (n=9911, $d_{train} = 1$)	BART	1.00	0.08	0.08	0.08	88.9
	BART (M)	0.89	0.1	0.1	0.1	89.5
	T5	0.90	0.11	0.11	0.11	89.3
	T5 (M)	1.00	0.09	0.09	0.09	90.0
RAKFKQLL-B*08:01 (n=1837, $d_{train} = 4$)	BART	1.00	0.01	0.01	0.01	79.1
	BART (M)	0.98	0.01	0.01	0.01	73.6
	T5	0.84	0.01	0.01	0.01	82.6
	T5 (M)	0.98	0.0	0.0	0.0	81.2
SPRWYFYYL-B*07:02 (n=4017, $d_{train} = 2$)	BART	0.98	0.03	0.03	0.03	85.1
	BART (M)	0.74	0.04	0.04	0.04	87.7
	T5	0.89	0.04	0.04	0.04	86.0
	T5 (M)	1.00	0.08	0.08	0.08	88.2
STLPETAVVRR-A*11:01 (n=943, $d_{train} = 9$)	BART	0.94	0.0	0.0	0.0	83.2
	BART (M)	0.91	0.01	0.01	0.01	82.9
	T5	0.77	0.0	0.0	0.0	82.4
	T5 (M)	0.83	0.0	0.0	0.0	84.1
TPRVTGGGAM-B*07:02 (n=2898, $d_{train} = 6$)	BART	1.00	0.09	0.09	0.09	87.7
	BART (M)	1.00	0.07	0.07	0.07	87.5
	T5	0.87	0.08	0.08	0.08	88.0
	T5 (M)	0.98	0.06	0.06	0.06	87.3
TTDPSFLGRY-A*01:01 (n=717, $d_{train} = 1$)	BART	1.00	0.01	0.01	0.01	78.9
	BART (M)	0.85	0.01	0.01	0.01	78.7
	T5	0.87	0.01	0.01	0.01	81.2
	T5 (M)	0.98	0.08	0.08	0.08	88.6
YLQPRTFLL-A*02:01 (n=2771, $d_{train} = 1$)	BART	0.84	0.55	0.55	0.55	96.2
	BART (M)	0.95	0.09	0.09	0.09	22.9
	T5	0.75	0.0	0.0	0.0	78.7
	T5 (M)	1.00	0.71	0.71	0.71	97.1
YVLDHLIVV-A*02:01 (n=16916, $d_{train} = 6$)	BART	0.86	0.02	0.02	0.02	86.9
	BART (M)	0.92	0.05	0.05	0.05	87.1
	T5	0.94	0.03	0.03	0.03	86.4
	T5 (M)	0.92	0.04	0.04	0.04	87.3

A.4. Dataset-Based Performance Characterization

Table 4: Characterization of Train/Test Target Overlap

Test Peptide	Closest Train Peptide	Edit Distance	CDR3b Overlap
AVFDRKSDAK	TPAFDKSAF	6	0/1655
CRVLCCYVL	LSEFCRVLCCYVLEE	6	0/435
EAAGIGILTV	AAGIGILTV	1	3/489
ELAGIGILTV	ELAGIGLTV	1	5/1919
GILGFVFTL	ILGFVFTLT	2	0/8089
GLCTLVAML	YVFCTVNAL	6	0/7388
IVTDFSVIK	ITDQVPFSV	6	0/563
KLGGALQAK	TRLALIAPK	6	0/12660
LLLDRLNQL	**LLLLDRLNQL**	**1**	**601/2103**
LLWNGPMAV	LLFGYPVAV	4	0/2458
LPRRSGAAGA	LPSYAALAT	5	0/2140
LVVDFSQFSR	VYADSFVIR	7	0/1871
NLVPMVATV	NLVPVVATV	1	1/8456
RAKFKQLL	RLSFKELLV	4	0/916
SPRWYFYYL	**LSPRWYFYY**	**2**	**2815/3390**
STLPETAVVRR	RSLAPEVRGYW	9	0/925
TPRVTGGGAM	TPRDLGACI	6	0/2606
TTDPSFLGRY	HTTDPSFLGRY	1	46/451
YLQPRTFLL	**YLQPRTFL**	**1**	**841/1650**
YVLDHLIVV	YLNDHLEPWI	6	0/8318

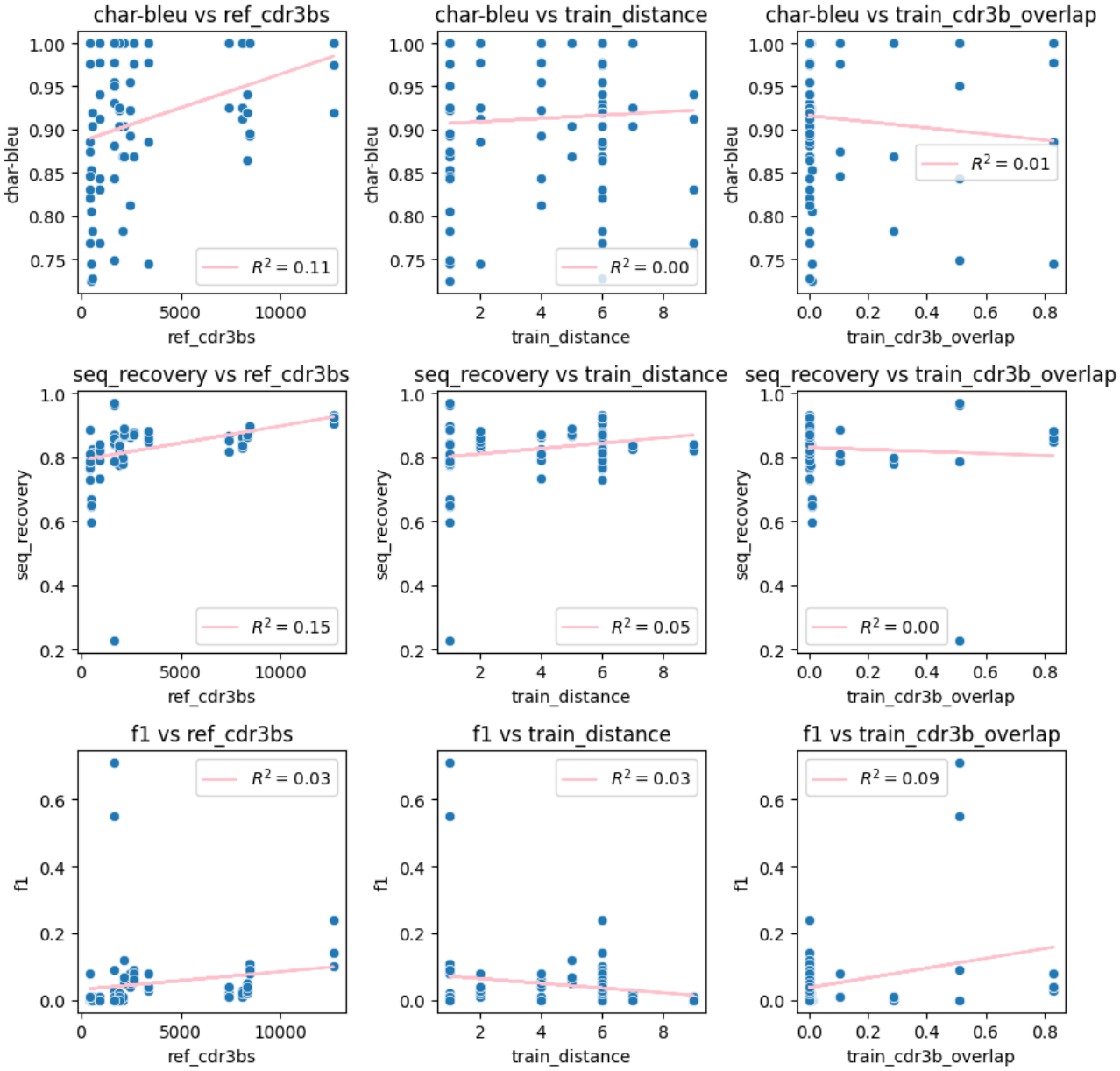


Figure 2: *Dataset Characteristic Correlation Plot*. Pairwise correlation between performance metrics (CharBLEU, sequence recovery, F1@100) and potential explanatory variables (number of reference CDR3bs, edit distance to closest training set epitope, CDR3b overlap between pMHC and closest epitope) shown for all 20 pMHCs with the baseline and multi-task models (n=80 per plot).