Applications of Artificial Intelligence &

Machine Learning in Cancer Immunology

By John-William Sidhom

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

December, 2018

Abstract

The treatment of cancer has long relied upon the use of non-specific and toxic chemotherapies and radiation that target quickly dividing cells. As a result, many patients experience the severe side effects associated with these therapies including vomiting, nausea, fatigue, and alopecia. Additionally, these therapies fail to provide durable and lasting responses in most cases of metastatic disease.

The immune system has long been though to play an important role in preventing cancer through immune surveillance; the idea that the immune system is poised with the means to detect cancer early on and eliminate malignant cells. However, as evident by aggressive disease, cancer is able to evade immune recognition and ultimately become very advanced. In recent years, immunotherapy has changed the treatment paradigm for several types of cancer. Of note, checkpoint blockade inhibitors have provided durable and lasting responses for a minority with metastatic disease. While these advances in therapy have provided hope where there was none in the cases of aggressive disease, there is still much work to be done to expand the benefits of immunotherapy for a small subset of patients to the whole.

In an effort to understand why certain patients respond to immunotherapy while other do not, there has been an effort to collect as much data through a variety of high-throughput 'big data' assays including whole exome sequencing, single-cell assays, and T-cell receptor sequencing. In this doctoral work, we develop a variety of machine learning and artificial intelligence methods to parse the nature of this data to unveil concepts that have helped us understand the prerequisites for a successful immune response to eliminate cancer. Of note, we develop a collection of deep learning algorithms to understand the interaction of peptide-MHC and T-cell receptor that is ultimately responsible for successful recognition of tumor by the immune system.

Committee: Dr. Drew M. Pardoll (advisor), Dr. Alexander S. Baras, Dr. Steven Salzberg

Dedication

Tried to put you on the team; you rather be the mascot.

Acknowledgements

I first would like to acknowledge my family including:

- Mom & Dad for always supporting me and giving opportunities very few people in this world have to pursue their passions and dreams for how they want to spend their lives. For their investment in me as a person, I owe everything I have accomplished and everything I will accomplish to them.
- Eriene & Mary-Joy for constantly challenging me and showing me how we all can pursue excellence in very different ways. While they are my younger sisters, I look up to them in many ways.
- Leo (my dog) who has very much become a core part of our family and brings so much joy to me and our family each and every day.

I would like to acknowledge Alexander S. Baras who has served as a mentor, a colleague, and friend throughout my doctoral experience. He has by far invested the most time in me, and has constantly looked out for my best interests.

I would like to acknowledge Drew M. Pardoll who has served as my primary mentor and advisor throughout my doctoral experience. It truly has been a dream to train under a giant and legend in the field of immunotherapy. The opportunities he has afforded me cannot be more appreciated. Mostly, I cannot thank him enough for allowing me to be the person I am in all of my dimensions and providing me the platform to grow as a scientist and individual.

I would also like to thank all my friends who have supported me during this time including John Azer, Bishoy Hanna, Meena Hanna, Mena Morgan, John Rophael, Mina Attia, Michelle Anis, Marianne Fahmy, Debebe Theodros, and Michael Foote.

Finally, I would like to thank all the diverse communities I have been privileged to be part of during my time here in Baltimore that each has served as a kind of home and family away from home.

Table of Contents

I. Introduction – pg. 1

II. ImmunoMap: A Bioinformatics Tool for T-Cell Repertoire Analysis – pg. 4

III. AI-MHC: an-allele integrated deep learning framework for improving Class I & Class II HLA-binding – pg. 29

IV. DeepTCR: a deep learning framework for revealing structural concepts within TCR Repertoire – pg. 46

V. ExCYT: A Graphical User Interface for Streamlining Analysis of High-Dimensional Cytometry Data – pg. 63

VI. Convolving Pre-Trained Convolutional Neural Networks at Various Magnifications to Extract Diagnostic Features for Digital Pathology – pg. 87

VII. Conclusion – pg. 96

VIII. References – pg. 97

IX. Appendices – pg. 105

List of Tables

Table III.1. AUC Values for Class I Alleles – pg. 41

Table III.2. Class I – IEDB Benchmark Performance – pg. 41

Table III.3. AUC Values for Class II Alleles – pg. 43

Table III.4. Class II – IEDB Benchmark Performance – pg. 43

Table V.1. Overview of All Functions Present in the ExCYT GUI – pg. 72

Table V.2. Overview of Software-assisted Flow-Cytometry Analysis Solutions – pg. 86

List of Figures

Figure I.1. The Immune Synapse – pg. 1

Figure I.2. Kaplan-Meijer Survival Curve as a function of Mutational Load – pg. 2

Figure II.1. Elements of ImmunoMap Algorithm – pg. 14

Figure II.2. Naïve Repertoire in Kb-SIY vs Kb-TRP2 antigens. – pg. 18

Figure II.3. Effects of Tumor on TCR Repertoire – pg. 20

Figure II.4. Effects of Tumor on TCR Repertoire in Various Lymphoid Organs – pg. 22

Figure II.5. TCR Repertoire Analysis of Patients Undergoing α-PD1 (Nivolumab) Therapy – pg. 24

Figure III.1. AI-MHC Architecture – pg. 34

Figure III.2. Network Characterization – pg. 38

Figure III.3. ROC for Class I and Class II models – pg. 39

Figure IV.1. Deep Learning Architectures – pg. 50

Figure IV.2. Unsupervised Learning Examples – pg. 53

Figure IV.3. Supervised Learning Examples – pg. 56

Figure V.1. ExCYT Pipeline & Features – pg. 70

Figure V.2. ExCYT Graphical User Interface – pg. 71

Figure V.3. Recapitulation of Myeloid Sub-Populations from Chevrier et. al. – pg. 81

Figure V.4. Recapitulation of Lymphoid Sub-Populations from Chevrier et. al. – pg. 82

Figure VI.1. Feature Extraction & Neural Net Architecture – pg. 91

Figure VI.2. Train on Data Set A, Test on Data Set A – No Image Color Augmentation – pg. 92

Figure VI.3. Train on Data Set A, Test on Data Set A – Image Color Augmentation – pg. 93

Figure VI.4. Train on Data Set A & Data Set B, Test on Data Set A – No Image Color Augmentation – pg. 93

Figure VI.5. Train on Data Set A & Data Set B, Test on Data Set A – With Image Color Augmentation – pg. 94

I. Introduction

The treatment of cancer has long relied upon the use of non-specific and toxic chemotherapies and radiation that target quickly dividing cells. As a result, many patients experience the severe side effects associated with these therapies including vomiting, nausea, fatigue, and alopecia[1]. Additionally, these therapies fail to provide durable and lasting responses in most cases of metastatic disease[2].

In recent years, there has been a significant shift in the way oncologists treat cancer with the advent of immunotherapy; the approach of harnessing the host immune system to target and eliminate malignant cells. The concept that the immune system could play a key role in prevention and progression of cancer first came in 1909 by Ehrlich, who proposed that through a mechanism of "surveillance," the immune system could detect and eliminate cancer.[6] Further work in the 2001 showed that RAG2-deficient mice (lacking T-Cells, B-cells, and NK cells) spontaneously developed adenomas of the lung and intestine[3]. With the advent of monoclonal antibodies, Ritixumab (anti-CD20) was one of the first clinically implemented forms of immunotherapy used as a first-line therapy for low-grade or follicular CD20-positive non-Hodgkin's lymphoma.

In the most recent wave of cancer immunotherapy, checkpoint blockade therapy (α-PD1,α-CTLA4) has provided the potential to unleash the immune system in novel ways. While PD1 and CTLA4, found on CD8+ T-cells and regulatory T-cells, are meant to control an overactive and potentially harmful immune response, malignant cells can signal through these molecules to suppress the immune response as a form of peripheral tolerance (**Figure 1**)[4,5]. By blocking these signals on T-cells, the 'brake' is removed and they can carry out their cytotoxic activity against the tumor.

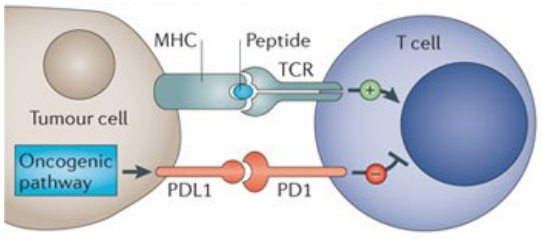


Figure 1: Malignant cells are able to induce a form of peripheral tolerance by constitutively expressing PDL1, which engages PD1 on T-cells (6).

The simplicity of checkpoint blockade is that it does not require prior knowledge of the antigenic targets of the T-cells it activates. This allows the drug to be broadly applied to various types of malignancies quickly without prior sequencing of the patient's tumor. In essence, it assumes that the patient's immune system is equipped with the means to eliminate the tumor and simply needs to be given the right advantage through checkpoint blockade. Unfortunately, while responses on checkpoint blockade can be durable and are associated with far fewer side effects than traditional chemotherapy, the response rate is relatively low. In the initial α-PD1 trials, across non-small-cell lung cancer, melanoma, and renal-cell cancer (considered highly immunogenic), the cumulative response rates were 18%, 28%, and 27%, respectively[6]. Much effort has been placed on understanding the immunological reasons for successful response to therapy so that response rates can be increased. Initial studies demonstrate that the efficacy of the checkpoint blockade is highly correlated with mutational load (**Figure 2**), suggesting that the CD8 T cells that are being activated against the tumor are neoantigen specific; targeting patient-specific passenger mutations such as missense mutations that occur due to the pathogenesis of the disease[7].

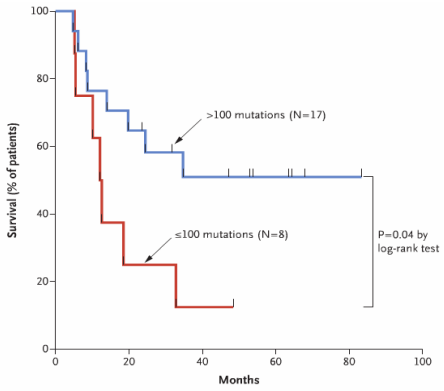


Figure 2: Kaplan-Meier Survival Curve as a function of mutational load. Patients with higher mutational loads had improved response to checkpoint blockade (7).

While preliminary studies have elicited some themes that are associated with response to immunotherapy, such as total mutational burden (TMB), there still exists a lot of weakly understood reasons for why certain patients benefit from therapy while others do not. As a result, investigators have developed and applied a variety of technologically advanced techniques such as whole-exome sequencing (WES) to understand the genetics of different tumors, immune

repertoire analysis in the form of T-cell repertoire analysis to query the adaptive immune response that is responsible for eradicating tumors, and a host of single-cell techniques including flow cytometry, CyTOF, and single-cell RNA sequencing (SC-RNAseq) to describe the phenotypic nature of the tumor microenvironment. The resulting tour de force has resulted in very large datasets that are highly complex and hold an incredible amount of information. Thus, there exists great opportunities in developing algorithms to parse these high-dimensional datasets in order to help understand the complex interaction of the immune system with cancer with the potential of developing better biomarkers and immunotherapies.

In the computer science field, there has also been a similar revolution in the fields of artificial intelligence (AI) and machine learning (ML). In fact, the greatest use of AI/ML has been for pattern recognition in large datasets and the results have been in some cases, better than human performance[8–10]. In this work, we explore a variety of machine learning methods and create tools for the field to analyze a variety of data generated from the cancer immunology field including WES, TCRSeq, and CYTOF data and demonstrate the power of these methods to provide descriptive and predictive insights that ultimately hold promise for improving our understanding of how cancer evades immune recognition and how to better treat patients of the future.

II. ImmunoMap: A Bioinformatics Tool for T-Cell Repertoire Analysis

Abstract

Despite a dramatic increase in T-cell receptor (TCR) sequencing, few approaches biologically parse the data in a fashion that both helps yield new information about immune responses and may guide immunotherapeutic interventions. To address this issue we developed a method, ImmunoMap, that utilizes a sequence analysis approach inspired by phylogenetics to examine TCR repertoire relatedness. ImmunoMap analysis of the CD8 T-cell response to self-antigen (K^b-TRP2) or to a model foreign-antigen (K^b-SIY) in naïve and tumor-bearing B6 mice showed differences in the T-cell repertoire of self- versus foreign antigen-specific responses, potentially reflecting immune pressure by the tumor, and also detected lymphoid organ-specific differences in TCR repertoires When ImmunoMap was used to analyze clinical trial data of tumor-infiltrating lymphocytes (TILs) from patients being treated with anti–PD-1, ImmunoMap, but not standard TCR sequence analyses, revealed a clinically predicative signature in pre- and post-therapy samples.

Introduction

The advent of high-throughput immune sequencing has allowed scientists and clinicians to understand antigen-specific interactions of the immune response with various pathologies from a systems-like perspective. Its initial applications showed the depth and breadth of the T-cell receptor (TCR) and B-cell receptor (BCR) repertoire[11–14]. Further applications of immune deep-sequencing have contributed to vaccine development and to tracking disease progression in malignancies[15–17]. Sequencing efforts have improved our understanding of immune responses to cancer, characterizing the TCR repertoire of circulating as well as tumor-infiltrating lymphocytes (TILs)[18,19].

With the abundance of new "big data" sets, a need has arisen to develop biologically meaningful techniques for analysis of TCR repertoire sequences. Current analysis methods fall short of providing an intuitive understanding of the immune system repertoire for two reasons. First, as purely mathematical constructs, they focus on diversity defined as a function of the number of different sequences and their respective frequencies, Shannon entropy, and ignore sequence relatedness[20,21]. Second, methods that compare different repertoires apply stringent criteria by only comparing exact TCR clonotypes (whether at the nucleotide or amino acid level) to assess similarity [19,22,23].

However, biological sequence similarity, not identity, is the relevant parameter. Ignoring sequence relatedness is a significant omission, becauseTCRs with similar, homologous sequences likely recognize related MHC/peptide targets. Several groups have sought to understand the structural aspects of the underlying TCR repertoire through a variety of techniques that cluster homologous CDR3 sequences, showing that, indeed, TCRs that recognize the same antigen have highly homologous sequences[24–26]. Although this work highlights the relevance and rationale behind

analyzing TCR sequence repertoire data via clustering methods, we sought to create structural diversity metrics for whole TCR repertoires. To address this, we developed ImmunoMap, which visualizes and quantifies immune repertoire diversity in a holistic fashion. ImmunoMap not only enables assessment of similarity between TCR sequences, but displays the scope of diversity among different repertoires. Our approach combines information about the frequency and relatedness of TCR sequences by using a sequence analysis inspired by phylogenetics to determine relatedness among cells of an antigen-specific T-cell response, as well as the similarities of a particular TCR repertoire to other repertoires.

We initially trained ImmunoMap on the TCRs used by T cells responding to K^b-TRP2, a shared self-peptide tumor antigen, and K^b-SIY, a model foreign-antigen, in a model of murine melanoma. ImmunoMap analysis showed that in naïve animals, the response to K^b-SIY was highly conserved, with many TCR sequences having high sequence homology, a biological observation that we missed when using Shannon entropy calculations. In contrast, the bulk of the self-antigen response was comprised of fewer and more distantly related sequences. The presence of tumor had a differential effect on the shaping of the repertoire in the model foreign and self-antigen responses, greatly altering the TCR repertoire of the self-antigen response, with a smaller effect on the response to the foreign antigen. To understand the clinical utility of ImmunoMap, we compared ImmunoMap to Shannon entropy analysis of TILs from melanoma patients on anti–PD-1 therapy. Whereas Shannon entropy calculations did not find any clinically relevant correlates, ImmunoMap found clinically relevant, predicative TCR signatures in patients who responded to anti–PD-1 therapy after just 4 weeks on therapy. Thus ImmunoMap proved more effective in finding a clinically useful parameter that another repertoire analysis technique could not.

Materials and Methods

Mice: C57BL/6j mice were purchased from Jackson Laboratories (Bar Harbor, ME). All mice were maintained according to Johns Hopkins School of Medicine IACUC 4-5 mice (gender and age matched) were used and pooled for each stimulation condition, based on previous T-cell expansion experiments, and each stimulation and sequencing run was performed once. Mice were randomly selected for naïve or tumor-bearing treatments and principal investigator was blinded to which mice received tumors. Murine experiments for naïve and tumor-bearing spleens were duplicated in separate pools of animals to demonstrate reproducibility of antigen-specific repertoire characteristics (Supplementary Fig. S7).

Preparation of MHC-Ig dimers and nano-aAPC: Soluble MHC-Ig dimers, K^b-Ig, was prepared and loaded with peptides as described [27].. Nano-aAPC were manufactured by direct conjugation of MHC-Ig dimer and anti-CD28 antibody (37.51; Biolegend) to MACS Microbeads (Miltenyi Biotec) as described previously[28].

Lymphocyte isolation: Mouse lymphocytes were obtained from homogenized mouse spleens after hypotonic lysis of RBC. Cytotoxic lymphocytes were isolated using a CD8 magnetic enrichment column from Miltenyi Biotec (Cologne, Germany) following the manufacturer's instructions. Lymphocytes from lymph nodes were obtained from homogenized inguinal lymph nodes and enriched with nano-aAPCs and plated for 7 days. For tumor-bearing animals, murine melanoma cell line B16-SIY, obtained with the consent of Tom Gajewski (The University of Chicago, IL, USA) through Charles Drake in 2011, and re-authenticated in the past year by flow cytometry, was injected subcutaneously after 5 days of culture, measured by calipers and harvested when tumors reach over $50mm^2$. The B16-SIY cell line is a tumor model modified to express SIY, a completely foreign epitope to the murine B6 background. In naïve mice setting experiments, it was used as a

model foreign antigen such as would be a viral epitope and in tumor-bearing animals serves as a tumor antigen. Tumor-infiltrating lymphocytes were obtained from tumors by manual digestion and washing, a density gradient centrifugation (Lympholyte Cell Separation Media, Mouse, Cedar Lane), and then tumor cells plated for 3 hours at 37°C and lymphocytes washed off and plated with nano-aAPCs ($1.25x10^9$ particles/mL). All cell lines underwent testing for mycoplasma contamination.

Enrichment and expansion: Nano-aAPC were stored at a concentration of 8.3 nM (5×10^{12} particles/mL), and all volumes refer to particles at this concentration. Ten million $CD8^+$-enriched lymphocytes at ~10^8 cells/mL were incubated with 10 μL of nano-aAPC for 1 hr at 4 °C, for an approximate bead:cell concentration of 5000:1. Cell-particle mixtures were subsequently passed through a magnetic enrichment column, the negative fraction was collected and the positive fraction eluted. Positive fractions were mixed and cultured in 96-well round-bottom plates for 7 days in complete RPMI-1640 medium supplemented L-glutamine, non-essential amino acids, vitamin solution, sodium pyruvate, β-mercaptoethanol, 10% FBS, ciproflaxin, and 1% T-cell growth factor, a cytokine cocktail derived from stimulated PBMC as described in the literature[29], in a humidified 5% $C0_2$, 37 °C incubator for 1 week. Specificity of CTLs was monitored on day 7, by FACS analysis following LIVE/DEAD cell stain (Thermo Fisher), anti-CD8 (BD Pharmingen, Cat# 553035, 53-6.7), and dimeric MHC-Ig staining. The number of antigen-specific cells was calculated by multiplying the number of total T cells by the fraction of $CD8^+$ and antigen-specific T cells; the fraction of antigen-specific cells was calculated after subtracting the non-cognate MHC staining from cognate MHC staining.

Sorting and sequencing of antigen-specific $CD8^+$ T cells: Following LIVE/DEAD cell stain (Thermo Fisher), anti-CD8 (BD Bioscience), and dimeric MHC-Ig staining, cells were sorted by

gating on cells with cognate Dimeric MHC-Ig staining over non-cognate staining. Antigen-specific $CD8^+$ T cells were sent directly for CDR3 β-chain sequencing by Adaptive Biotechnologies.

In vitro *nano-aAPC functionality assay*: 7 days following enrichment and expansion antigen specificity is confirmed by intercellular cytokine staining. Briefly, RMA-S, given by Michael Edidin (Johns Hopkins University, MD, USA) in 1996 (reauthenticated in the past year by peptide stabilization assays and cultured for 7 days prior to use) are peptide pulsed (10 μM) overnight at room temperature with relevant or no peptide and mixed 1:2 RMAS:T-cell ratio with expanded T cells. Unpulsed RMAS cells were used as background stimulation. After 6 hours, cells were washed twice with FACS wash buffer and then stained with viability dye and anti-CD8 for 20 minutes. Cells were then fixed and permeabilized with the Cytofix/Cytoperm kit (BD Biosciences) following the manufacturer's protocol. Anti-TNFα (Biolegend, Cat# 506324, MR6-XT22) was added to the cells and stained for an hour.

Precursor frequency assessment: On day 0 following $CD8^+$ T-cell isolation from splenic cells, $CD8^+$ T cells were stained with LIVE/DEAD cell stain (Thermo Fisher), anti-CD8 (BD Bioscience), and dimeric MHC-Ig staining viability stain with either unloaded or peptide-loaded MHC-Ig. Cells gated on Live cells and anti-$CD8a^+$ staining.

Collection of TILs from patients undergoing α-PD1 therapy:

Eighty-five patients, providing written consent, were accrued to a multi-arm, multi-institutional, institutional-review-board-approved, prospective study (BMS-038) to investigate the pharmacodynamic activity of nivolumab. All patients received nivolumab (3 mg/kg Q2W) until progression for a maximum of 2 years. Tumor samples were collected prior to and four weeks after initiation of nivolumab therapy. The samples were stored in RNA*later*®(Ambion). 34 patients

permitted TCR sequencing, and DNA was extracted and submitted to Adaptive Biotechnologies for survey level TCR β-chain sequencing [13,30]. Clinical response was assessed via CT scan after 24 weeks of therapy.

Deconvolution methods: Due to the fact that animals were pooled together on day 0 prior to expansion of antigen-specific cells, there was only one sequencing run. In order to determine the variance of calculated indicators of the repertoire, the reads from the sequencing file were randomly distributed into the number of bins corresponding to the number of animals that went into the experiment. This method of random deconvolution assured that the variance of the indicator by random chance was not greater than the difference observed between conditions.

Weighted repertoire dendrograms: For the antigen-specific sequencing, productive sequences with a frequency > 0.01% were taken for analysis. For anti–PD-1 clinical trial analysis, Adaptive Biotechnologies' files were first filtered to only include sequences with reads greater than or equal to 5 and then top 40% of response was taken for analysis. Sequence distances were calculated based on sequence alignments scores using a PAM10 scoring matrix and gap penalty of 30. Distance matrix was used to create a dendrogram using the Bioinformatics toolbox in MATLAB. Circles were overlaid at the end of the branches corresponding to the CDR3 sequences with diameters proportional to the frequency of the sequence. When using the terminology "weighted repertoire dendrogram," this does not infer that the distance matrix used to create the dendrogram is weighted; rather, the dendrogram is visually 'weighted' by frequency.

Dominant motif analysis: Using the cluster function in MATLAB toolbox, dendrogram was divided into homologous clusters using a homology threshold obtained from analyzing an unexpanded adult $CD8^+$ T cell population from a C57BL/6 animal (Supplementary Fig. S1). Clusters whose average sequence distance within cluster <= threshold and met a certain frequency

cutoff (3% - Supplementary Fig. S1 were denoted as “Dominant Motifs.” Cluster frequency was lowered to 1% for α-PD1 clinical trial analysis but held consistent across all patients due to the fact that this was not a single antigen-specific population of cells.

Singular and novel clone analysis: In order to define singular clones, a matrix was setup to calculate the mapped sequence distance of every unique combination of sequences in the repertoire. Using standard matrix operations within MATLAB, a singular clone was defined as a clone whose frequency was 10x the sum of all other homologous clones. Homologous clones were those who had a sequence distance determined from the dominant motif analysis. In order to define novel clones, the same approach was used, but the matrix was setup in that it calculated the mapped sequence distance of every unique combination of sequences between the two repertoires being compared. A novel clone was defined as a clone whose frequency was 10x the sum of all homologous clones in the other sample.

TCR diversity score: This measurement of diversity was calculated in a similar method as the singular & novel clone analysis. An initial matrix is created where the mapped sequence distance is calculated for every unique combination of sequences in the repertoire. Then the average of the unique combination calculations is taken, weighted by reads, and reported as the TCR diversity score. Additional details of the algorithm behind this calculation are shown in Supplementary Fig. S6.

Shannonntropy calculations: Calculation of Shannon entropy was completed by the following formula where p_i represents the frequency of each amino acid sequence and n represents the total number of sequences present in the response:

$$Shannon's\ Entropy = \sum_{i=1}^{n} p_i \ln(p_i)$$

Statistical methods: No specific statistical method was used to determine sample size for the stimulation cohorts. Two-tailed *t*-tests were used as provided by GraphPad Prism 5 software for all comparative statistics given we expect normal distributions across all experiments.

Code availability: In order to use the ImmunoMap algorithms, we have developed a MATLAB-based Graphical User Interface (GUI) that can be found along with the source code at https://github.com/sidhomj/ImmunoMap. Supplementary Fig. S8 demonstrates the use of the GUI.

Data Availability: TCR β-chain sequencing raw data for the murine experiments is found in supplementary materials.

Results

Overview of ImmunoMap Algorithms

Weighted Repertoire Dendrograms: In order to visualize the immune response, we created weighted dendrograms; combining information about sequence relatedness with information about sequence frequency. We initially applied this analysis to data (from the Adaptive Biotechnologies Data Portal[31]) on the response of tetramer-sorted human $CD8^+$ T cells to cytomegalovirus (CMV; **Fig. 1A)**. The distance from the end of the dendrogram branches denotes distance in terms of sequence homology; the size of the circles at the ends of the branches denotes frequency of the sequence, and color denotes Vβ usage. Sequence distance is determined as a function of global alignment scores (Needleman-Wunsch [32], PAM10 scoring matrix [33], Gap Penalty = 30) between all unique combination of sequences as follows:

$$Score12 = Sequence\ Alignment\ Score\ (Sequence\ 1, Sequence\ 2)$$

$$Score11 = Sequence\ Alignment\ Score\ (Sequence\ 1, Sequence\ 1)$$

$$Score22 = Sequence\ Alignment\ Score\ (Sequence\ 2, Sequence\ 2)$$

$$Sequence\ Distance = (1 - \frac{Score12}{Score11})(1 - \frac{Score12}{Score22})$$

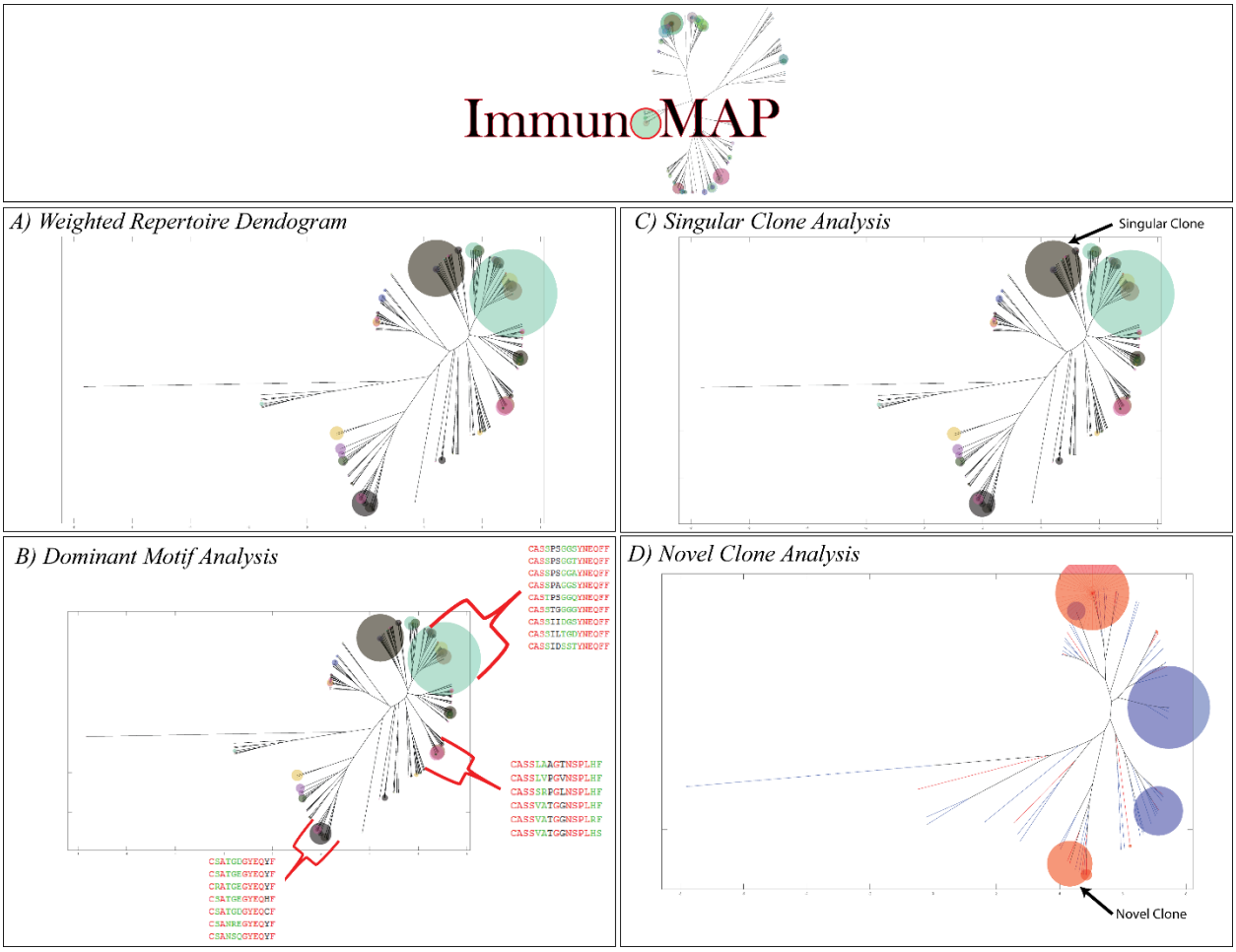


Figure 1. Elements of ImmunoMap Algorithm A) Weighted Repertoire Dendogram visualize relatedness of sequences within repertoire along with relative frequency of CDR3 amino acid sequences. B) Dominant Motif Analysis clusters homologous sequences and selects for clusters contributing to significant proportion of the response. 3 Dominant Motifs are shown representing highly represented structural motifs in this individual's CMV response. C) Singular Clone Analysis defines sequences that expand significantly over the summation of all other homologous sequences D) Novel Clone Analysis is implemented when comparing repertoires from different samples and a novel clone is defined as one that expands significantly over the summation of all homologous sequences in the other sample.

Dominant Motif Analysis: In order to parse the many sequences that are detected in antigen-specific CTL expansion, we sought to perform hierarchical clustering to determine structural motifs that dominated the response. Thresholds for sequence homology and frequency were set by analyzing the sequences of the naïve B6 $CD8^+$ repertoire, taken from the Adaptive Biotechnologies Data Portal[34], **Supplementary Fig. S1**. We used these thresholds to define homology clusters based on sequence distance and then examined clusters that met a predefined frequency threshold and termed them "dominant motifs" **(Fig. 1B)**.

Singular and Novel Structural Clones Analysis: We also defined a "singular structural clone" as one that has expanded 10x more than the summation of all other homologous clones in a sample, representing a singular solution in "sequence space." **(Fig. 1C).** When comparing two separate CMV-specific sequencing samples, from different individuals, we defined a "novel structural clone" as one that has expanded 10x more than the summation of all homologous clones to it in another sequencing sample, representing a newly expanded structural clone **(Fig. 1D)**.

TCR Diversity Score: To quantify the diversity of the entire TCR repertoire, we created a metric to quantify the relatedness of an entire sample; defined as the average mapped sequence distance of all unique combinations of sequences in a sample, weighted by number of reads per sequence.

$$Mapped\ Sequence\ Distance = 1 - \frac{1}{1 + [Sequence\ Distance]}$$

The TCR Diversity score is bounded between 0 and 1, in which a score of 0 would correspond to all TCRs in a response being identical and 1 would correspond to all TCRs being infinitely different (full details of algorithms to calculate TCR diversity score in Supplementary Fig. S6).

Naïve TCR Repertoires against Model Tumor Antigens

To understand the clonal diversity of antigen responses, $CD8^+$ T cells from naïve B6 mice were pooled and expanded against a model foreign-antigen K^b-SIY, or against a self-tumor antigen, K^b-TRP2 (180-188), as described[27,35]. Briefly, $CD8^+$ T cells were enriched and stimulated with nanoparticle artificial antigen-presenting cells (aAPCs) containing peptide-MHC-Ig molecules and cultured *in vitro* for 7 days (**Fig. 2A**). The resultant $CD8^+$ T-cell cultures were antigen-specific by both peptide-MHC-Ig staining and cytokine analysis, confirming their functional specificity (**Supplementary Fig. S2**). Initial precursor frequency was also measured in the endogenous repertoire and, even though T cells that recognize either antigen could be expanded from naïve animals, K^b-SIY antigen–specific T cells had a higher naïve precursor frequency (**Supplementary Fig. S2**). Antigen-specific populations were sorted and the CDR3 region of the TCR Vβ chain was sequenced.

ImmunoMap analysis of K^b-SIY-specific and K^b-TRP2-specific TCRs (**Fig. 2B**) visualized unique aspects of the polyclonal response for both antigens. K^b-SIY $CD8^+$ T cells consisted of clones with homologous TCR sequences; however, the naïve response to K^b-TRP2 was more clonal in nature (more high frequency clones) and used more unrelated sequences, each creating a distinct clonal variant for antigen recognition.

Dominant motif analysis showed that anti-K^b-SIY TCR had fewer, yet richer (more sequences per motif), dominant motifs than K^b-TRP2 (**Fig. 2C, D**). K^b-TRP2 specific T cells had a higher percentage of clones representing singular structural T-cell expansions and they took up a larger portion of the overall TRP-2 antigen-specific response (**Fig. 2D – *bottom***). Comparing the TCR diversity scores, K^b-SIY stimulated a more homologous response, whereas K^b-TRP2 had a more diverse response. The response to K^b-SIY had a more conserved Vβ usage, predominantly using

Vβ13, whereas the response to K^b-TRP2 exhibited a more diverse use of Vβ segments (**Supplementary Fig. S3**).

To demonstrate the advantages of the ImmunoMap analysis over traditional analytic methods, we calculated Shannon entropies for K^b-SIY vs K^b-TRP2 responses (**Fig. 2E**). Shannon entropies revealed the diversity of the K^b-SIY response to be higher than that of the K^b-TRP2 response. However, because the Shannon entropy is largely determined by the number of sequences that are present in the K^b-SIY response and not their relatedness, it missed the fact that although more sequences responded to K^b-SIY, they were more convergent than the fewer sequences that responded to K^b-TRP2. Thus, the ImmunoMap TCR diversity score and dominant motif analyses reflected novel relatedness-information that could not be seen by conventional Shannon entropy calculations.

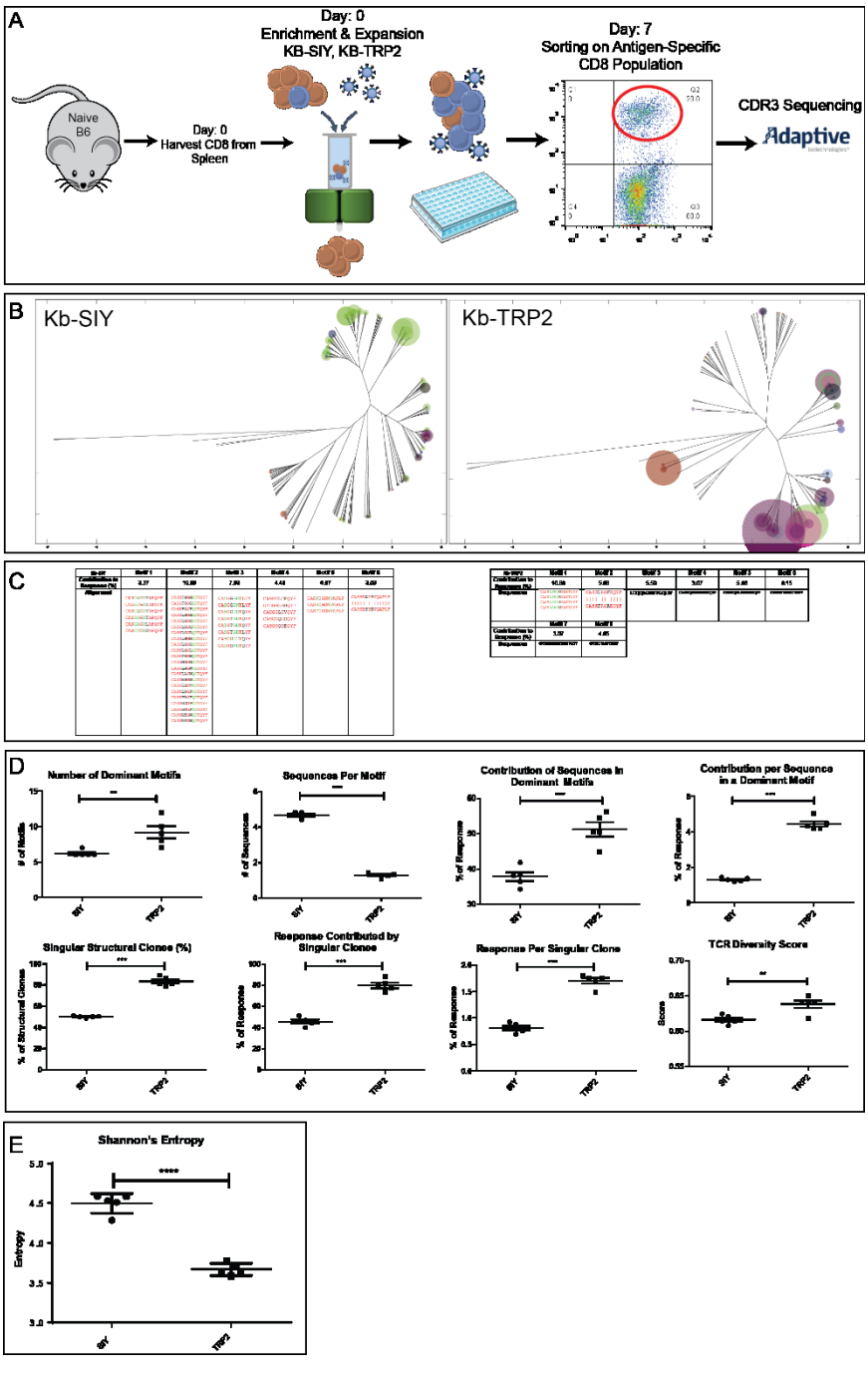


Figure 2. Naïve Repertoire to Kb-SIY vs Kb-TRP2 antigens. A) Naïve B6 Animals were harvested for CD8 T cells from spleens. CD8 T cells underwent Enrichment & Expansion protocol via nano-aAPC's and then were cultured ex-vivo for 7 days before undergoing flow-sorting for antigen-specific cells and being sent for TCR β-chain sequencing. B) Weighted Repertoire Dendograms where distance of ends of branches denotes sequence distance, size of circles denotes frequency of sequence, and color of circle denotes specific V beta segment usage. C) Demonstration of dominant motif's detected for Kb-SIY (left) and Kb-TRP2 (right). Frequency and global sequence alignment is shown (Red = Fully Conserved Amino Acids, Green = Semi-Conserved Amino Acids, Black = Nonconserved Amino Acids). D) Top Row – Quantification of Dominant Motif Analysis comparing the number of dominant motifs, the number of sequences per motif, the contribution of the sequences in the dominant motifs to the response, and the contribution to the response per sequence in a dominant motif. Bottom Row – Singular Structural Clone and TCR Diversity analysis metrics. (n=5). E) Shannon's Entropy Calculations comparing endogenous Kb-SIY to Kb-TRP2 responses. (** : P-value ≤ 0.01, *** : P-value ≤ 0.001 using the unpaired two-tailed t-test, bar represents mean ± s.e.m)

Tumor Exerts Differential Expansion Pressure on Antigen-Specific Repertoire

Although we know that tumors exert pressure on the immune response, it is not clear how this alters the repertoire of responding T cells. The ImmunoMap approach can provide insight into the biological impact of tumors on T-cell responses and TCR usage by studying TCR repertoire changes in the presence of tumor (B16-SIY)[36]. Visualization of the TCR repertoire by ImmunoMap analysis (**Fig. 3A**) showed differential effects of tumors on the repertoire of pooled splenic T cells specific for K^b-SIY or K^b-TRP2. The K^b-SIY $CD8^+$ T cell repertoire was largely unaltered in response to tumors. In contrast, as seen by ImmunoMap, the K^b-TRP2 response was not only more clonal, but also used TCR sequences that had minimal sequence homology to the TCRs seen in the naïve C57BL/6 response.

Dominant motif analysis showed that the presence of tumors increased the number of dominant motifs in the K^b-SIY response (**Fig. 3B**). In contrast, the presence of tumors decreased the number of dominant motifs in the K^b-TRP2 response, suggesting directed immune pressure on the self vs foreign antigens in the context of tumor. The dominant K^b-SIY motifs were conserved (**Fig. 3C**). In contrast, no common dominant motifs were shared in the K^b-TRP2 response in tumor-bearing animals compared to the naïve response. When examining novel structural clones (**Fig. 3D**), the K^b-TRP2 response in tumor-bearing mice had more structurally novel sequences that, combined, were a larger portion of the response as compared to the naïve response. Selective pressure by tumors on the immune response was also seen in analyzing the Vβ usage between naïve and tumor-bearing animals (**Fig. 3E**). We saw the elimination of the use of Vβ16 in the K^b-TRP2 response, and an increased use of Vβ5. In contrast, Vβ usage was conserved in the K^b-SIY response between naïve and tumor-bearing animals (**Supplementary Fig. S4)**.

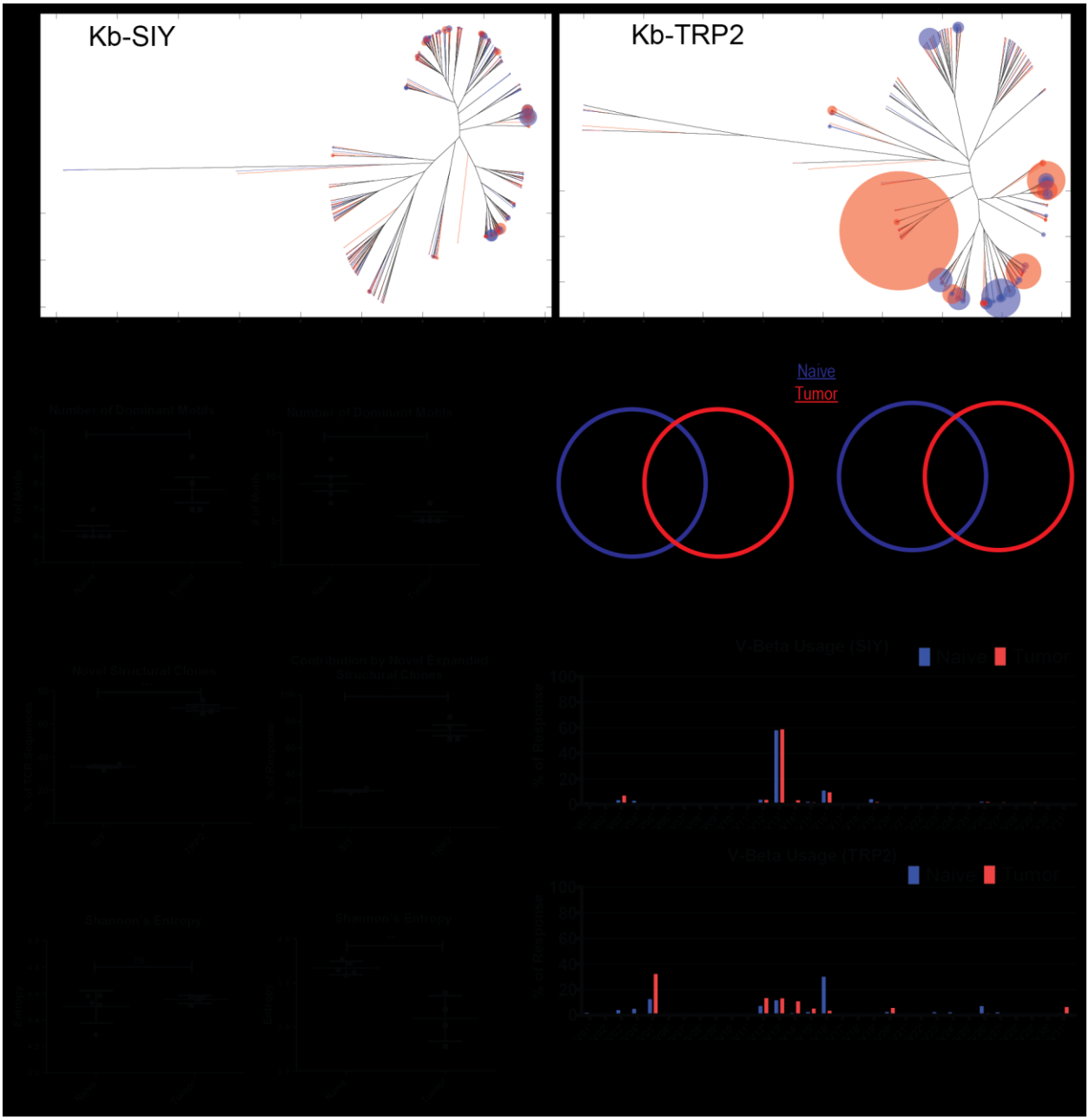


Figure 3. Effects of Tumor on TCR Repertoire. A) Overlapped weighted repertoire dendograms of tumor-bearing vs naïve antigen-specific splenic CD8 responses. (Red = Tumor-bearing repertoire. Blue = Naïve repertoire). B) Dominant Motif analysis for Kb-SIY and Kb-TRP2 responses before and after exposure to tumor (n=5 mice). C) Maintenance of Dominant Motifs between Naïve and Tumor-Bearing Repertoire. D) Novel Structural Clone Analysis (n= 5 mice). E) V Beta usage of Kb-SIY and Kb-TRP between Naïve and Tumor-Bearing Repertoire. F) Shannon's Entropy Calculations comparing endogenous vs tumor-bearing responses to Kb-SIY and Kb-TRP2. (* : P-value ≤ 0.05, *** : P-value ≤ 0.001 using the unpaired two-tailed *t*-test, bar represents mean ± s.e.m)

Additionally, when examining the effect of tumors on Shannon entropy (**Fig. 3F**), we see that although maintenance of entropy in the K^b-SIY response and its decrease in the K^b-TRP2 response generally complement the ImmunoMap dominant motif analysis, Shannon entropies are uninformative about the conservation, or lack thereof, of TCR sequence structure in response to the tumor.

Lymphoid Organ-Dependent Differences in TCR Repertoires in Tumor-Bearing Mice

We hypothesized that the influence of tumors on the repertoire may also vary depending upon the relationship of the lymphoid organ to the tumor site. This was studied by analyzing antigen-specific TCR repertoires in the spleen versus draining lymph node (dLN), and TILs in pooled tumor-bearing mice lymph nodes and tumors. ImmunoMap analysis revealed that the K^b-SIY repertoire selects for effective structural motifs as one probes compartments closer to the tumor site. This is seen as the richness of dominant motifs decreases, the response contributed by singular clones increases, and the TCR diversity score drops as one moves from the spleen towards the tumor **(Fig. 4B)**. Additionally, the structural clones expanded in the spleen, dLN, and TILs are generally conserved, as can be visualized by the dendrograms **(Fig. 4A)** and by tracking dominant motifs in the 3 lymphoid compartments **(Fig. 4C**). In contrast, the opposite trend was seen in the K^b-TRP2 response. Additionally, dominant motifs between the spleen and draining lymph node were not conserved; we were unable to expand any K^b-TRP2 specific cells from the TILs in multiple experiments (**Fig. 4 A-C**).

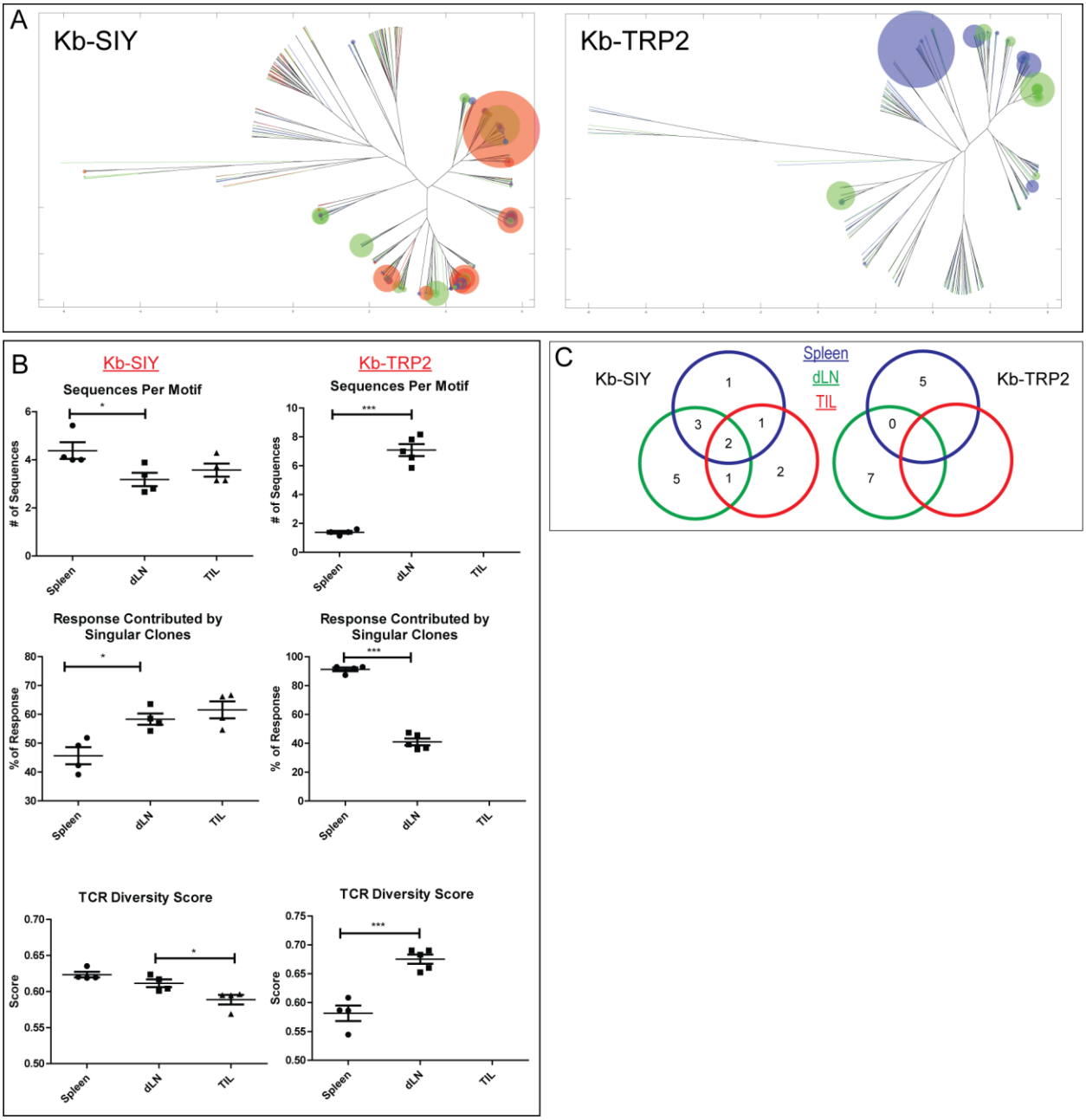


Figure 4. **Effects of Tumor on TCR Repertoire in Various Lymphoid Organs.** A) Overlapped weighted repertoire dendograms (blue = spleen, green = draining lymph node, red = TILs). B) Dominant Motif and TCR Diversity Metrics (Kb-SIY n=4 mice, Kb-TRP2 Spleen n=4 mice, Kb-TRP2 dLN n=5 mice). C) Maintenance of Dominant Motifs between various lymphoid organs. (* : P-value ≤ 0.05, *** : P-value ≤ 0.001 using the unpaired two-tailed *t*-test, bar represents mean ± s.e.m)

Analysis of anti–PD-1 Clinical Trial Data Reveals Indicators of Response

Recent studies have implicated changes in T-cell responses as important in clinical outcomes to checkpoint blockade. We therefore applied ImmunoMap analysis to clinical trial data (BMS-038) from patients with metastatic melanoma undergoing anti–PD-1 therapy (nivolumab). For this analysis, formalin-fixed, paraffin embedded scrapings were taken from 34 patients, the percentage of TILs estimated as per Adaptive protocol (Materials and Methods) and CDR3 regions of Vβ-chains sequenced before and while on therapy (Fig. 5A). The number of TCRs sequenced in all samples analyzed was not significantly different (Supplementary Fig. S5).

ImmunoMap was used to compare the TCR repertoire before and after 4 weeks of anti–PD-1 therapy (all ImmunoMap metrics in BMS038Results.xlsx). Weighted repertoire dendrograms (**Fig. 5B**) revealed distinct differences between responders and nonresponders. Dominant motif analysis (**Fig. 5C**) showed that patients who had more dominant motifs prior to initiation of therapy had more favorable responses to therapy. Additionally, those patients who had a decrease in their TCR diversity score (**Fig. 5C**) on therapy had more favorable outcomes to therapy. In contrast, no clinically relevant signature could be found by Shannon entropy calculations (**Fig. 5D**). Thus ImmunoMap analysis was superior in its ability to reveal repertoire characteristics that could predict response to therapy after only four weeks of treatment.

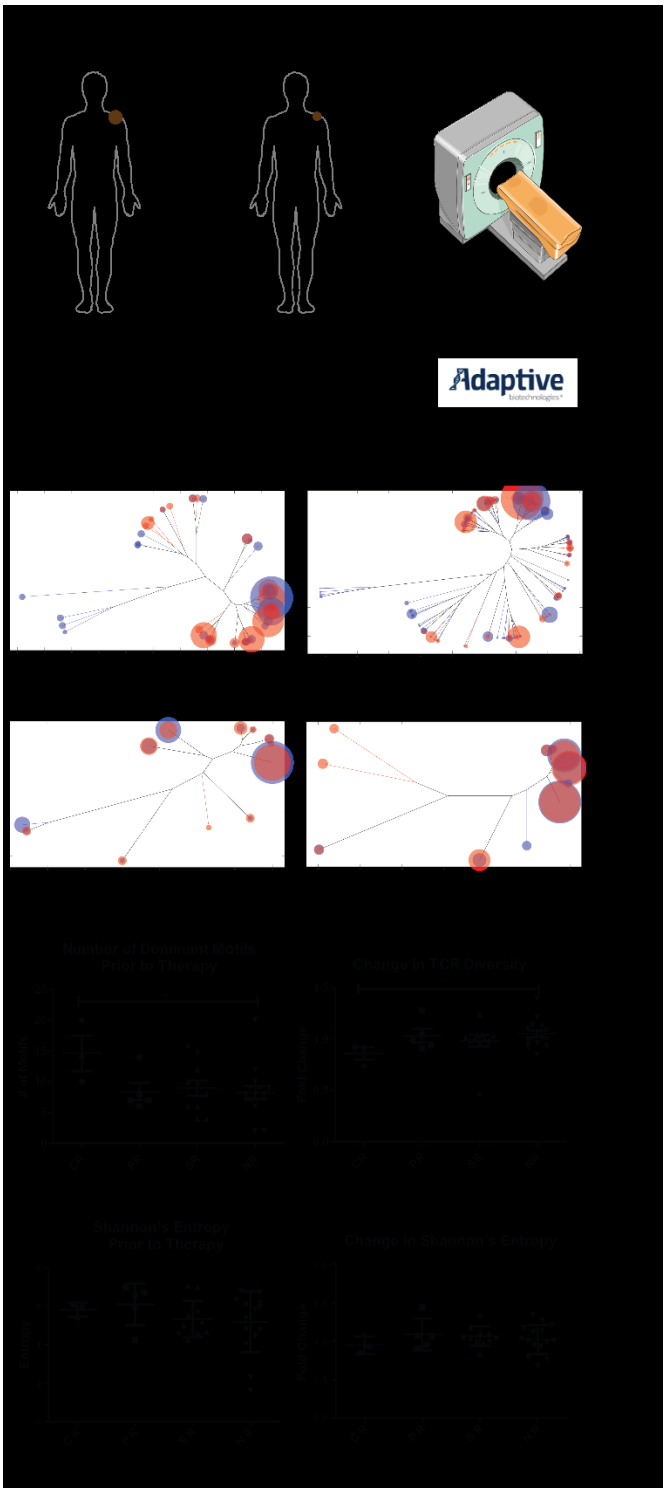


Figure 5. TCR Repertoire Analysis of Patients Undergoing α-PD1 (Nivolumab) Therapy. A) Clinical Protocol for sample collection and response stratification. Pre-therapy biopsies were taken from tumor sites prior to initiation of therapy. 4 weeks after initiation of α-PD1 therapy, on-therapy biopsies were taken from same tumor sites. TIL extraction was completed sent to Adaptive Biotechnologies for CDR3 β-chain sequencing. B) Token weighted repertoire dendograms for each of the cohorts of responders. C) Dominant Motif and TCR Diversity analysis (CR = 3, PR=5, SR=11, NR=15).D) Shannon's Entropy Calculations for responses prior and after initiation of α-PD1. (* : P-value ≤ 0.05 using the Mann Whitney test, bar represents mean ± s.e.m)

Discussion

Here we introduce ImmunoMap, a bioinformatics approach to analyze TCR repertoire sequence data, and used it to characterize repertoire changes in responses to model murine tumors and in patients undergoing immunotherapy for melanoma. By combining information about sequence relatedness and frequency, ImmunoMap allows an intuitive appreciation of TCR repertoire characteristics that reconciles the structure and function of the repertoire.

ImmunoMap analysis comparing foreign (K^b-SIY) and self (K^b-TRP2) antigens showed distinct differences in the naïve repertoire to these two different antigens. Although interesting, the conclusions of this analysis cannot be expanded to all foreign vs self-antigens. The presence of more dominant motifs in the K^b-TRP2 response in combination with greater clonality suggests that central and peripheral tolerance mechanisms limited clonal responses, with more distinct clones occupying a larger portion of the TRP2-specific repertoire. Self-reactive clones, with TRP2-specific TCRs would either be removed during central thymic development or tolerized in the periphery, explaining the inability to find more numerous TCR sequences per dominant motif[37–39]. Because our analysis was conducted on expanded antigen-specific populations, our results demonstrate the "expansion potential" of the antigen-specific T-cell repertoire for a model foreign and shared tumor antigen in the setting of both naïve and tumor-bearing animals. It is possible that the limited TCR relatedness of TRP2 responses could be due to the lower precursor frequency in naïve animals, and the T cells that have the ability to expand do not cluster in the same dominant motif due to lower initial cell frequency. The impact of pooling animals prior to expansion and sequencing must also be considered. In this scenario, one could be selecting for "public" clones and possibly enriching for these parts of the repertoire over "private" clones, unique to each animal. Although individual mice are genetically identical, VDJ recombination occurs as an

independent process in each animal and the primary TCR repertoire capable of responding to a given antigen could vary between individual animals. Therefore, the effects on shaping of the repertoire may be most relevant to "public" or conserved sequences. Finally, the higher TCR diversity score of K^b-TRP2 alongside with the higher number of dominant motifs suggests that the immune system has to reach further to find solutions to bind the cognate antigen/MHC complex.

Although prior work on TCR clustering has focused on understanding the structural aspects that confer antigen-specificity[24,25], the effects of perturbations to the immune system on antigen-specific responses has not been studied. With ImmunoMap, we studied the changes in repertoire in response to tumor. We observed that the effects of tumor on the anti-self-K^b-TRP 2 peptide repertoire indicate that tumors exert greater pressure on the self- than on the foreign-antigen. Not only did the presence of the tumors correlate with an increase the clonality of the response to self, via decreases in the number of dominant motifs and increases in their contribution to the net response, but tumor-bearing mice could shift their response to different, presumably suboptimal, motifs. Additionally, the differences in repertoire characteristics among various lymphoid organs for the two different model antigens indicates that tumors effectively eliminated the expansion of certain clones from its microenvironment. The consequences of these findings are relevant to both antigen-discovery and targeting for immune therapies related to treating cancer. Due to limitations of personalized antigen-specific therapy, targeting shared antigens like MART1, a self-antigen specific for melanocytes, has been a mainstay of antigen-specific cancer immunotherapy[40–42]. This approach has typically relied on TCR transgenic models in which a single TCR clone is chosen as the source for the antigen-specific receptor[43–45]. Given our analysis, several problems with this approach become apparent: (1) antigen-specific expansion not only generates a diversity of TCR sequences but one that spans the entire sequence distance of the naïve repertoire, (2) self-antigen

expansion represents a limited repertoire and arsenal against a given epitope due to effects of tolerance, and (3) the tumor can exert pressure on the self-antigen–specific immune response in a more profound way than in the case of a foreign antigen. Our findings call into question the approach of using self or over-expressed antigens as targets for immune therapy and highlight the importance of exploring responses to neoantigens, novel MHC-specific epitopes that arise from mutations in a patient's individual malignancy[46–50] .

We also have used ImmunoMap algorithms to understand mechanisms of successful immune responses to cancer against 4T1, a murine breast cancer model[51]. In that model, when analyzing TILs from animals treated with anti–CTLA-4, radiation, or the combination of these therapies, we found that the TCR structural repertoire before therapy from TILs was highly conserved, seemingly targeting a single antigen, whereas after combination therapy, the structural response broadened within the TILs and each individual animal developed its own uniquely expanded repertoire[51].

Finally, we used ImmunoMap to study TILs from clinical trial specimens to determine if structural diversity is an important parameter in determining successful immune responses to cancer immunotherapy. Our analysis revealed that patients who had more dominant motifs prior to therapy responded more favorably to therapy. Additionally, the change in TCR diversity suggest that patients who respond to therapy converge on a solution of successful TCR sequences and thus their repertoire is actually less diverse after therapy. In contrast to previous work by Madi *et. al* that demonstrated a structural broadening of the peripheral repertoire to anti–CTLA-4 therapy in melanoma patients, but did not correlate this finding with response, we focused our analysis on studying changes in the repertoire within the TILs and could determine structural signatures of response[26]. Although our findings are significant, we note the scope of the clinical trial was limited,

which impacted the distribution of clinical responses. Nevertheless, taken together, ImmunoMap analysis revealed that patients with a broader repertoire prior to therapy have a higher probability of expanding effective TCR sequences and converging on them.

ImmunoMap not only has potential for the clinical monitoring of patients on therapy, through predictions of their likelihood to respond, but enables the acquisition of biological insights about antigen-specific immune responses that could alter current immune therapies.

III. AI-MHC: an-allele integrated deep learning framework for improving Class I & Class II HLA-binding

Abstract

Motivation: The immune system has potential to present a wide variety of peptides to itself as a means of surveillance for pathogenic invaders. This means of surveillances allows the immune system to detect peptides derives from bacterial, viral, and even oncologic sources. However, given the breadth of the epitope repertoire, in order to study immune responses to these epitopes, investigators have relied on *in-silico* prediction algorithms to help narrow down the list of candidate epitopes, and current methods still have much in the way of improvement.

Results: We present Allele-Integrated MHC (AI-MHC), a deep learning architecture with improved performance over the current state-of-the-art algorithms in human Class I and Class II MHC binding prediction. Our architecture utilizes a convolutional neural network that improves prediction accuracy by 1) allowing one neural network to be trained on all peptides for all alleles of a given class of MHC molecules by making the allele an input to the net and 2) introducing a global max pooling operation with an optimized kernel size that allows the architecture to achieve translational invariance in MHC-peptide binding analysis, making it suitable for sequence analytics where a frame of interest needs to be learned in a longer, variable length sequence. We assess AI-MHC against internal independent test sets and compare against all algorithms in the IEDB automated server benchmarks, demonstrating our algorithm achieves state-of-the-art for both Class I and Class II prediction.

Availability and Implementation: AI-MHC can be used via web interface at baras.pathology.jhu.edu/AI-MHC

Introduction

The ability for T-cells to recognize various epitopes is of paramount importance to mounting a potent immune response and ultimately protecting the host[52]. The relevance for understanding the 'epitome' for humans to viruses, bacteria, and even various cancers has been vital for advances in vaccine development, understanding how pathogens escape immune recognition, and even predicting how cancer patients will respond to immunotherapy[53–55]. Despite how much is known about epitope production including processing by the immunoproteasome, transport into the endoplasmic re.ticulum (ER), and binding and presentation via major histocompatibility (MHC) molecules, prediction of presented epitopes to the immune system is still a difficult task[56,57].

The complexity of the task has led many groups to use advanced methods in machine learning and artificial intelligence to learn patterns in known MHC-binding peptides in order to recognize these patterns when seen in unknown peptides[58,59]. Artificial neural networks (ANN's) have been employed by some of the leading algorithms to date to act as feature extractors in order to recognize patterns[59,60]. Artificial neural networks, due to their flexibility in terms of changing their capacity, serve as universal function approximators, and therefore can learn patterns difficult for humans to pick up on. Building on the principal of using neural networks, groups have recently begun to utilize a type of neural network architecture termed convolutional neural networks (CNN's) which were originally developed for the purpose of image classification where features in an image can be found in different locations and different orientations. By being translationally invariant to features, these networks put together the presence of multiple features in an image in order to make a decision as to what object is present in the image[8]. This concept as applied to

sequence analysis has been exploited in analyzing DNA-protein binding domains as well as predicting HLA Class I binding[61–63]. [61–63]

While these most recent advances in neural network architectures have improved the accuracy of these algorithms, there are still areas for improvement. As a general shortcoming, most neural-network based methods of conducting MHC-binding predictions create several models across different alleles and different sequence lengths. The result of this process is that while the entire data set of known allele/peptide pairings is large, the data becomes split between models where each model can only learn sequence features for a subset of the peptides. However, it is known that neural networks, especially deep learning models, show the most increase in performance when more data is provided for training. *Andreatta et. al* demonstrated that using a gapped-sequence alignment method, they could feed variable length sequences into a fixed-input neural network by providing an additional parameter that specified the length of the original sequence ($L \leq 8$, $L = 9$, $L = 10$, $L \geq 11$), showing an improvement in MHC Class I binding prediction from being able to leverage more data in one model.

In order to best leverage the amount of data available for known MHC binding, we developed Allele-Integrated MHC (AI-MHC), a unified architecture capable of predicting binding for either all Class I or all Class II alleles, regardless of sequence length. By allowing MHC allele to be an input into the network and joining this with a global max pooling operation following convolutions across the peptide sequence, our architecture is able to leverage the most amount of data in a single model. This approach achieves state-of-the-art performance for Class I and Class II predictions.

Materials and Methods

Dataset

In order to train the Class I network, we pulled linear epitopes from the Immune Epitope Database (www.iedb.org) who had Class I restriction in humans with quantitative measurements of ic50 by purified MHC competitive radioactive and purified MHC competitive fluorescence assays, defining binding as peptide/allele pairings with ic50's < 500nm. We transform the ic50 values by the equation (1) to scale from 0 to 1 where values below 1 nM are set to 1 nM and values above 50,000 nM are set to 50,000 nM.

$$Affinity = 1 - \frac{log_{10}(ic50)}{log_{10}(50{,}000)} \quad (1)$$

We additionally added another large data set of Class I binding predictions from *Kim et. al*[64]. We then restricted our training to entries where the full allele was provided (i.e. HLA-A*02:01) and then aggregated multiple peptide/allele pairings, taking the median value as a consensus where there were multiple peptide/allele pairings. In order to train the Class II network, we used a large data set published by *Jensen et. al*, following the same data preprocessing as described above[65].

For the purpose of comparing against other algorithms, we collected all the benchmarks from the IEDB automated server benchmarks for both Class I and Class II and restricted our analysis to benchmarks collected from competitive quantitative assays given our network was trained on data from these types of assays.

Network architecture

The conventional neural network architecture as initially conceived for image classification tasks generally follows the format of stacking multiple convolutional layers with some type of non-linear activation and generally a max pooling step[66]. This approach transforms a photograph that is wide and tall with few features (RGB channels) to one that is compressed but with many features. The max pooling operations reduce the size of the photograph as it passes through the network but still maintain local spatial information. Applying this architecture to biological sequence analysis where peptides have variable lengths becomes problematic since neural networks require fixed size inputs. In the image classification world, this can be solved by rescaling or padding photographs so they all have the same pixel-by-pixel dimensions. Rescaling works well since RGB channels are continuous variables where down-sampling or interpolation algorithms can be applied but fails to translate to sequence analysis as sequences do not have a continuous numerical representation. In order to tackle this problem, *Vang et. al.* took an approach where they trained the network for a fixed size input. However, this approach would prevent training an entire allele's set of peptides together which should significantly improve learning since the features between 9 and 10mers are most likely highly conserved. In order to be able to train a single model for an entire class of HLA molecules, we employed an approach where each peptide zero-padded (right) into a 15-mer window for Class I and 40-mer window for Class II. **(Figure 1A)**. This allows the network to take in sequences up to 15-mer in length for Class I predictions and up to 40-mer in length for Class II.

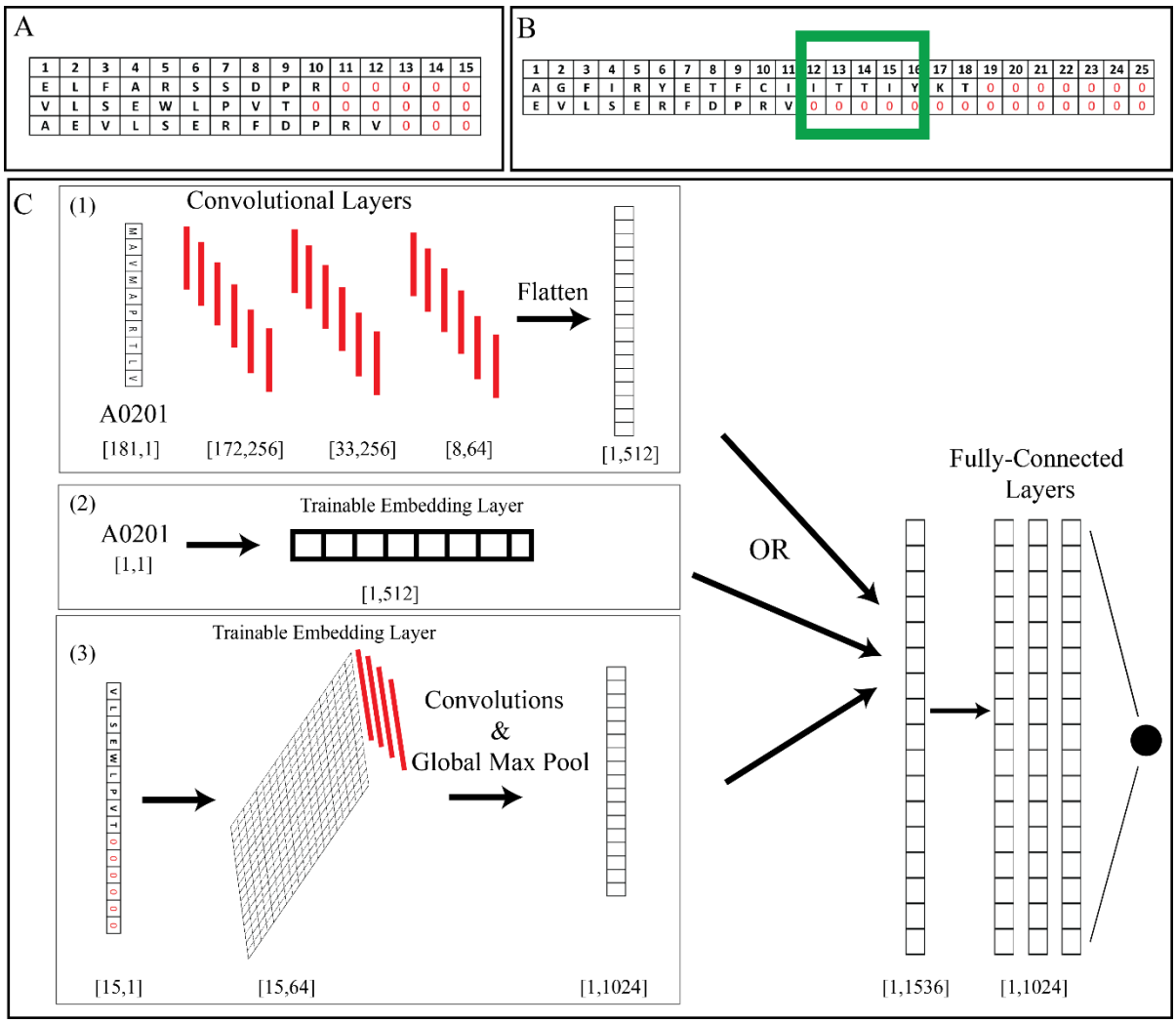


Figure 1: AI-MHC Architecture. A) Zero-padding scheme for handling variable length sequences. B) Green window highlights problem with max-pooling in segments as areas of null sequence can be directly compared to areas of real sequence. C) AI-MHC is designed to take a peptide/allele pair which are transformed with either (1) convolutional layers or (2) trainable embedding layers learning vector representations of alleles and (3) amino acids. 1024 10-mer convolutions with global max pooling are applied to the sequence resulting in a [1,1024] feature map for each sequence. The sequence feature map is then concatenated to the [1,512] allele feature map. This long-form vector [1,1536] is then followed by 3 fully-connected layers with 50% dropout and utilizing leaky relu activations functions with a final output node with sigmoid activation.

Since certain amino acids may share similar functional properties with others, we wanted to train an embedding that captures properties of each amino acid as the network is trained for prediction. In previous work, *Vang et. al* trained an embedding by using the Word2Vec algorithm to vectorize each amino acid based on its contextual use within the epitome. We chose to instead train the embedding with the classification task in mind as we believe this should learn the most salient embedding for the task at hand and our integrated approach would allow for the most amount of data to be leveraged towards training this embedding matrix.

In order to analyze this type of input to the network, we chose to use parallel convolutions of kernel length of 10-mers, knowing this should be large enough to encompass the 9-mer core that represents the length of the interacting peptide with the MHC molecule[67]. In comparison to other methods of biological sequence comparisons that use sequence alignment algorithms to assess conserved motifs, this network learns 1024 10-mer 'trainable' motif detectors[68]. The critical piece of the algorithm at this point is how it handles the max pooling step following these convolutions. Since our inputs can have variable lengths with zero-padding, if we chose a max-pooling strategy that divided the sequence into segments, we could be comparing segments in some windows where there was no sequence information to windows where there was sequence information **(Figure 1B)**. This would be especially problematic with Class II molecules where the input length can be highly variable. However, by conducting a global max-pooling operation, one is able to detect the relevant binding frame regardless of where it lies within the larger sequence.

In order to train one unified architecture to predict binding based on sequence and allele input, we required an input to the network to be an allele paired to a given peptide. In order to integrate information about the allele into the network, we experimented with two methods: 1) applying convolutional layers to the actual protein sequences[69] of the MHC alleles to extract

structural features of each allele **(Figure 1C-1)** or 2) training an embedding layer of 512 dimensions in order to learn properties of each allele **(Figure 1C-2)**. This is particularly advantageous because by training on a large dataset of all epitopes for various alleles, the net is able to learn features or train an embedding that can understand which alleles share similar properties and therefore, may share similar binding characteristics. Following either this convolutional feature extraction or embedding, this 512-dimensional vector is then joined with the 1024-dimensional feature vector for the sequence. In experimenting with both approaches, we found no difference in overall performance of the classifier and chose to implement a trainable embedding layer, as this was more computationally efficient. At this point, 3 fully connected layers (combining features extracted from MHC allele and peptide sequence) are implemented in which has final layer has a single output from a sigmoid activation, modeling the nM binding of the given MHC allele to peptide sequence pairing. The entire architecture was implemented with Google's TensorFlow™ deep learning library.

Results

Neural Network Characterization

The presented neural network architecture contains two critical features that facilitate the use of the largest combined dataset of a given class of MHC for training; 1) translational invariance by convolutional layers that utilize a global max-pooling operation and kernel size that encompasses the entire possible length of the binding interaction between the peptide and MHC molecule and 2) integration of MHC allele as paired input with peptide sequence via an embedding layer, thereby enabling the entirety of the MHC Class I data to be used ensemble for training as compared to stratification by MHC allele. Besides creating larger datasets by combining data from different alleles and varying sequence lengths, the architecture developed allows the network to learn the

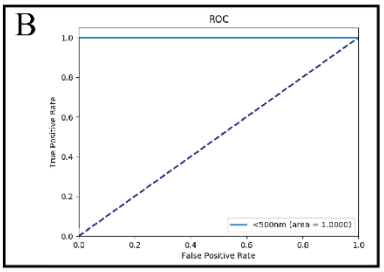
properties of both amino acids and MHC molecules during its training, since each have a trained embedding matrix based on the data. In particular, an MHC embedding layer learns features of the MHC, allowing the model to learn from a larger dataset and translate knowledge of binding between alleles in the same supertype, sharing similar binding properties. To prove these points, we conducted two experiments that asses the invariance of the network as well as assess the quality of the embedding in its ability to cluster similar amino acids and HLA alleles and translate knowledge across alleles in the same supertype.

In order to test the invariance of the network, we created a synthetic dataset of 10,000 peptides of varying lengths between 8-11 amino acids resembling either A0201 binding peptides, L at P2, V at P9[70], or a scrambled sequence containing a L and V at random positions **(Figure 2A)**. We provided the network the ability to learn one 10-mer feature, as this is the extent of information the network should need to make this classification correctly and were able to show that despite the L-V motif being placed in various frames in variable length peptides, our network was able to achieve perfect classification accuracy, as would be expected from a detect system that exhibited translational invariance **(Figure 2B**).

A

Positive	Negative
LDRRVMEVFQ	WVFVVTDVLT
LLLTAPLMCV	PQVTLYYK
ACLDKWYITV	VRVLVDELFS
MLGDHFSNV	LGHSDAVN
HLTHMWMPVV	ERIDTVCQLV
LCLSLGDVGD	IENLGVMK
LWEYFTSV	RLRIHVDQTC
MLYGKDERVA	FVEVLYCC
LWRWKEYV	KMADDLMVPE
LTRGGATVNY	LAETVFPCA

B



D

	Training Count	AUC (individual model)	AUC (unified model)
A0201	12324	0.942	0.956
A0202	4077	0.916	0.941
A0203	6244	0.936	0.973
A0206	5561	0.875	0.937
A0211	1084	0.826	0.865
A0212	1183	0.898	1.000
A0216	919	0.778	0.991
A0217	332	0.600	0.800
A0219	1244	0.909	1.000
A0250	135	1.000	1.000
A6802	5499	0.939	0.897
A6901	2565	0.670	0.923

C

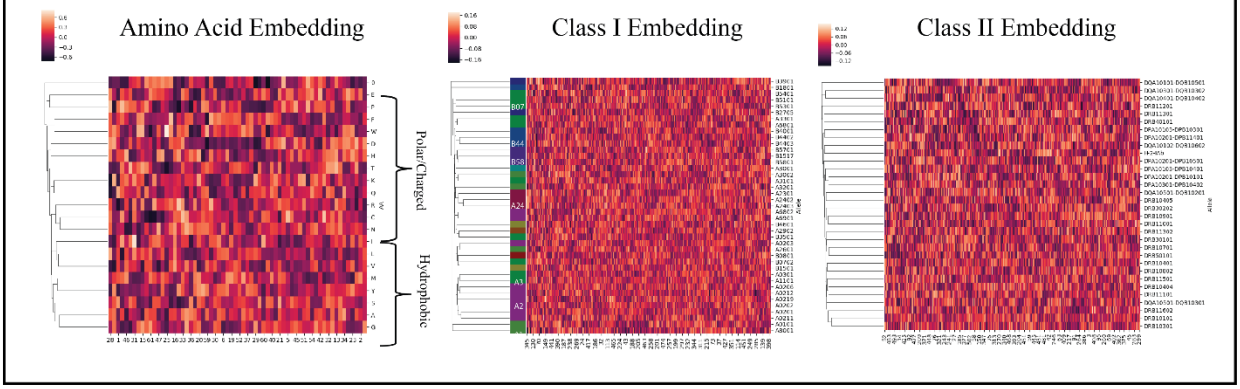


Figure 2: Network Characterization. A) Examples from synthetic dataset meant to mimic A0201 binding motifs (Leucine at P2, Valine at P9) in various frames within 8-11mer sequences. Red amino acids correspond to leucine and valine placed in correct and incorrect frames. B) Receiver Operating Characteristic of AI-MHC on synthetic dataset. C) Trained embedding layers were extracted from the network graph for amino acid, Class I, Class II embeddings and are visualized with clustermaps.

After training on MHC Class I and Class II data, we examined the embedding layers for both the amino acids and MHC alleles and noted that indeed amino acids with similar biophysical properties and MHC alleles in the same supertype **(Supplementary Table 1)**[71] were indeed clustered together **(Figure 2C)**, suggesting the network had learned which amino acids and HLA molecules share similar binding properties. While there does not exist a formal definition of supertypes for Class II molecules, we saw a similar clustering of related Class II molecules as well **(Figure 2C).** In order to assess the benefit of training an integrated model, we trained each of the sub-alleles of the HLA-A2 supertype in either individual or a unified model and compared their AUC values. We noted significant improvements in performance 10 of the 12 A2 supertype alleles **(Figure 2D)**, suggesting the network was able to translate its knowledge about the MHC-A2 supertype across its sub-alleles.

Class I Metrics

We collected a total of 148,540 unique allele/peptide pairing with ic50 values from the IEDB (www.iedb.org) and a previously published dataset by *Kim et.al* spanning 86 HLA-A,B,C,E alleles **(Supplementary Table 2)**. For the purpose of training, we split these data sets into a train set of 95% and split the remaining 5% for validation and testing, resulting in a train size of 141,113 peptides, a validation size of 3,713, and a test size of 3,714. The network was trained on the train data while validation data was used to determine when to stop training the neural network. In this set of peptides, our model achieved an overall AUC of 0.956 on

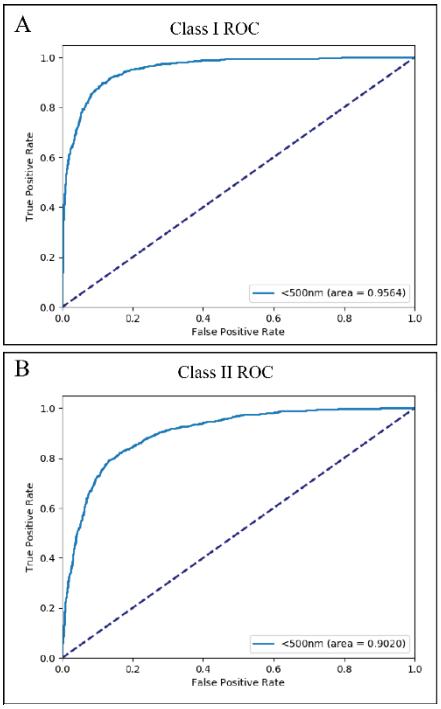


Figure 3: ROC for Class I and Class II models.

our internal independent test set **(Figure 3A)**, generally achieving higher AUC values for where there was more data available for a given MHC molecule **(Table 1)**.

In order to gauge where our algorithm stood against the current state-of-the-art algorithms, we pulled all the ic50 benchmarks from the IEDB (www.tools.iedb.auto_bench/mhci/weekly/) to assess the performance of our algorithms against the 11 provided algorithms. Since it is unclear whether these are considered independent benchmarks as the IEDB cannot verify that the tested allele/peptide pairings have not been seen by the benchmarked algorithms, it is difficult to truly compare performance at an algorithmic level. Nonetheless, we removed all records in the benchmarks from our training data before assessing the performance of our models. Of the 47 available benchmarks, AI-MHC performed the best on 9/36 datasets (next highest was NetMHC 3.4 with 7/36) where we had at least 10 peptide examples for training to a given MHC allele. **(Table 2)**. Full comparison of all algorithms on all benchmarks in **Supplemental Table 3.**

Allele	Peptide Counts	AUC
A0101	4609	0.950
A0201	12324	0.956
A0202	4077	0.941
A0203	6244	0.973
A0206	5561	0.937
A0211	1084	0.865
A0212	1183	1.000
A0216	919	0.991
A0217	332	0.800
A0219	1244	1.000
A0250	135	1.000
A0301	7195	0.968
A1101	6248	0.970
A2301	2416	0.953
A2402	3191	0.995
A2403	1227	0.990
A2501	960	1.000
A2601	4307	0.956
A2602	631	0.955
A2603	522	1.000
A2902	2548	0.980
A3001	2717	0.857
A3002	1847	0.775
A3101	5621	0.968
A3201	1089	1.000
A3215	74	1.000
A3301	3510	0.963
A6601	52	1.000
A6801	3708	0.960
A6802	5499	0.897
A6901	2565	0.923
A8001	1164	1.000
B0702	4513	0.937
B0801	3298	0.924
B0802	1000	0.983
B1501	4178	0.963
B1503	594	0.814
B1517	1446	0.945
B1801	2594	0.896
B2705	3433	0.897
B3501	3198	0.982
B3801	492	1.000
B3901	1623	0.906
B4001	3199	0.992
B4002	964	0.933
B4402	2117	0.991
B4403	1382	1.000
B4501	953	0.909
B4601	1806	0.955
B5101	2718	0.956
B5301	1616	0.938
B5401	1110	1.000
B5701	2781	0.963
B5801	3118	0.978
B8301	333	1.000

Table 1: AUC Values for Class I Alleles

Algorithm	Benchmarks Tested	# of Best Performances
AI-MHC	36	9
ARB	33	3
IEDB Consensus	28	2
NetMHC 3.4 (ANN)	33	7
NetMHC 4.0 (ANN)	3	3
NetMHCcons	30	6
NetMHCpan 2.8	33	4
NetMHCpan 3.0	3	1
PickPocket	30	6
SMM	36	3
SMMPMBEC	30	3
mhcflurry	5	3

Table 2: Class I – IEDB Benchmark Performance. We collected all benchmark datasets from the IEDB for which our algorithm had at least 10 training examples for the allele tested to assess performance against 11 of the available algorithms. # of Best Performances refers to the number of benchmarks a given algorithm 'won'.

Class II Metrics

In order to test whether this type of architecture would also be relevant in predicting Class II binding, we collected a total of 134,281 unique allele/peptide pairings with ic50 values from a previously published dataset by *Jensen et.al* spanning 80 alleles **(Supplementary Table 4)**. For the purpose of training, we again split this data set the same way as with the Class I training, resulting in a train size of 127,566 records, a validation size of 3,357, and a test size of 3,358. Training was completed in the same way as described above. In our internal independent data set, our model achieved an overall AUC of 0.902 **(Figure 3B & Table 3)**. In comparison to AUC values published by *Jensen et.al* on the same data set, our model outperforms all recorded AUC values from the NetMHCIIpan-3.2 (AUC = 0.858 ,0.861, 0.826). Furthermore, we pulled all ic50 benchmarks from the IEDB (http://tools.iedb.org/auto_bench/mhcii/weekly/) to assess the performance of our algorithm against the 6 provided algorithms. Once again, we benchmarked our algorithm by removing entries from our training set that appeared within the IEDB benchmarks. Of the 54 available benchmarks, our model performed the best on 18/48 datasets which we had at least 10 training examples, the second highest number of best performances to NetMHCIIpan-3.1 **(Table 4)**. Full comparison of all algorithms on all benchmarks in **Supplemental Table 5.**

Allele	Peptide Counts	AUC
DPA10103-DPB10201	787	0.967
DPA10103-DPB10301	1563	0.932
DPA10103-DPB10401	2725	0.868
DPA10103-DPB10601	584	1.000
DPA10201-DPB10101	2447	0.966
DPA10201-DPB10501	2470	0.939
DPA10201-DPB11401	2302	0.929
DPA10301-DPB10402	2641	0.914
DQA10101-DQB10501	2946	0.917
DQA10102-DQB10501	833	1.000
DQA10102-DQB10502	800	0.897
DQA10102-DQB10602	2747	0.829
DQA10103-DQB10603	462	0.917
DQA10104-DQB10503	883	0.875
DQA10201-DQB10202	944	0.855
DQA10201-DQB10301	827	0.847
DQA10201-DQB10303	761	0.949
DQA10201-DQB10402	768	0.976
DQA10301-DQB10301	207	1.000
DQA10301-DQB10302	3111	0.945
DQA10303-DQB10402	567	0.789
DQA10401-DQB10402	2890	0.869
DQA10501-DQB10201	2897	0.928
DQA10501-DQB10301	3585	0.925
DQA10501-DQB10302	847	0.816
DQA10501-DQB10303	564	0.938
DQA10501-DQB10402	749	0.943
DQA10601-DQB10402	565	0.911
DRB10101	10412	0.851
DRB10301	5352	0.859
DRB10401	6317	0.871
DRB10404	3657	0.870
DRB10405	3962	0.923
DRB10701	6325	0.888
DRB10801	937	0.929
DRB10802	4465	0.903
DRB10901	4318	0.888
DRB11001	2066	0.951
DRB11101	6045	0.890
DRB11201	2384	0.980
DRB11301	1034	0.961
DRB11302	4477	0.894
DRB11501	4850	0.879
DRB11602	1699	0.965
DRB30101	4633	0.918
DRB30202	3334	0.868
DRB30301	884	0.825
DRB40101	3961	0.857
DRB40103	846	1.000
DRB50101	5125	0.867
H-2-IAb	1794	0.922
H-2-IAd	774	0.835
H-2-IAs	190	1.000
H-2-IEd	245	0.750
H-2-IEk	68	1.000

Table 3: AUC Values for Class II Alleles

Algorithm	Benchmarks Tested	# of Best Performances
AI-MHC	48	18
Comblib matrices	18	0
Consensus IEDB method	36	6
NN-align	34	2
NetMHCIIpan-3.1	48	22
SMM-align	34	3
Tepitope (Sturniolo)	29	1

Table 4: Class II – IEDB Benchmark Performance. We collected all benchmark datasets from the IEDB for which our algorithm had at least 10 training examples for the allele tested to assess performance against 11 of the available algorithms. # of Best Per

Conclusion

In this work, we present an integrated deep learning architecture to predict MHC Class I and Class II binding, able to achieve state-of-the-art performance through utilizing innovative changes in architecture allowing the network to be trained on effectively larger datasets, which is a well-known requirement to better training deep neural networks. This is accomplished through training an entire class of MHC alleles in a unified model by learning an embedding layer for the allele allowing leveraging of binding information between alleles of the same supertype. Furthermore, the architecture becomes flexible to sequence input length by utilizing a global max-pooling operation across the input peptide sequence following convolutions to achieve translational invariance where a frame of interest needs to be learned in the context of a longer, variable length peptide sequence.

In attempting to assess the performance of our algorithm, we noted the difficulty in making equivalent comparisons to other algorithms in the field as there are no clear train/test datasets that all algorithms can be benchmarked against in sense that we could determine what data should be training versus independent test for any given algorithm. In an attempt to conduct a robust analysis, we first created an internal independent test set that was not used for training purposes, as per our methods section. Our results for Class I and Class II (AUC – Class I = 0.956 & AUC – Class II = 0.902) suggest our algorithm is one of the top performing algorithms. However, without the ability to train/test other algorithms, it is not possible to directly assess exactly where our algorithm stood. For Class I assessment, even when removing all IEDB examples from our training, our algorithm still had the highest number of best performances on benchmarks where we had sufficient training examples. For Class II, we felt we were able to make a fairer comparison as NetMHCIIpan-3.2 released a large dataset on which they trained/tested. By using the same data and their reported

AUC values, we were confident that our algorithm was truly out-performing what is considered the best Class II prediction algorithm by ~4% AUC despite not having more 'best performances' in the IEDB benchmarks. While our dilemma in assessing performance against other algorithms is not a new one, we suggest that there needs to be a method, such as the annual ImageNet Challenge, by which algorithms can be compared in a fair way where training/testing datasets are equivalent across all algorithms to truly assess the best algorithmic approaches. That being said, given the volume of data we collected for MHC class I and II in conjunction with the ability of our algorithm to more fully leverage each of the sets of data in total, our approach of isolating an internal independent test set still allows for the evaluation of performance across thousands of allele/peptides.

Finally, we have provided a user-friendly website for use of our algorithms for both Class I and Class II predictions with performance metrics provided for each allele. We believe this level of transparency in the allele-level performance is important to better inform the user of the confidence in any prediction based on the number of peptides tested and the internal AUC achieved for any given allele.

IV. DeepTCR: a deep learning framework for revealing structural concepts within TCR Repertoire

Abstract

Deep learning algorithms have been utilized to achieve excellent performance in pattern-recognition tasks, such as in image and vocal recognition[8,66]. The ability to learn complex patterns in data has tremendous implications in the genomics world, where sequence motifs become learned 'features' that can be used to predict functionality, guiding our understanding of disease and basic biology[61,63,72,73]. T-cell receptor (TCR) sequencing assesses the diversity of the adaptive immune system, and while prior conventional biological sequence analysis tools have been insightful, they can miss signals in the data due to their rigidity[68,74,75]. We present DeepTCR, a broad collection of unsupervised and supervised deep learning methods able to uncover structure in highly complex and large TCR sequencing data. We demonstrate its utility across multiple basic science and clinical examples, including learning antigen-specific motifs and understanding immunotherapy-related shaping of repertoire. We further extract meaningful motifs from the trained network as a means of explaining the sequence concepts that have been learned to accomplish a given task. Our results show the flexibility and capacity for deep neural networks to handle the complexity of high-dimensional genomics data for both descriptive and predictive purposes.

Next-Generation Sequencing (NGS) has allowed a comprehensive description and understanding of the complexity encoded at the genomic level in a wide variety of organisms. The applications of NGS have grown rapidly as this technology has become a molecular microscope for understanding the genomic basis for the fundamental functions of the cell[76]. In parallel to this explosion of NGS applications, in the machine learning world, deep learning has seen a similar expansion of applications as computational resources have grown, large advances in algorithms and programming libraries have distributed these capabilities to many scientific communities, and in particular, big data has transcended all facets of daily life. As a result of these two technological revolutions, there exists many opportunities to apply deep learning in genomics as the data generated from NGS is very large and highly complex.

T-cell receptor sequencing (TCRSeq) is an application of NGS that has allowed scientists across many disciplines to characterize the diversity of the immune system[77–79] (Supplementary Fig. 9). By selectively amplifying and sequencing the highly diverse CDR3 region of the β-chain of T-cells, scientists have been able to study the diverse repertoire the immune system generates to probe both foreign and native potential antigens. With this new sequencing technology, there has arisen a need to develop analytical tools to parse and draw meaningful concepts from the data. In recent work, investigators have applied conventional sequence analytics, where either targeted motif searches or sequence alignment algorithms have been applied to begin parsing the structural data within TCRSeq[68,74,75]. However, since many of these approaches were initially conceived to analyze longer biological strings for the purpose of identifying evolutionary changes at the DNA or protein level, problems can arise when applying them to TCRSeq data in which the strings being compared are quite short and the end regions are highly conserved. Finally, while these methods

are considered unsupervised machine learning approaches, there has been little in the way of using supervised approaches to guide the learning process.

We present DeepTCR, a package of both unsupervised and supervised deep learning methods for analysis of TCRSeq at both the sequence and sample level in order to learn concepts in the data that may be used for both descriptive and predictive purposes. In order to demonstrate the utility of these algorithms, we collected three previously published datasets including samples sorted by antigen-specificity (*Glanville_2017* & *Sidhom_2017),* and samples taken from cohorts of tumor-bearing mice treated with various immunotherapies *(Rudqvist_2017)*[51,68,74] (Supplementary. Fig. 10,11).

The first class of algorithms we developed are unsupervised deep learning methods that learn the underlying distribution of the sequence data in high-dimensional space for the purpose of 1) clustering TCR sequences that likely recognize the same antigen and 2) for the first time quantifying similarity between whole repertoires based on their structural composition. We implement both a variational autoencoder (VAE) and generative adversarial network (GAN) to perform dimensionality reductions and data re-representations at a sequence level, using convolutional layers in order to learn motifs that describe the distribution of data. We first implemented the VAE as autoencoders have been previously described as a common dimensionality reduction/data re-representation technique[80,81]. When implemented with trainable convolutional layers, they can become powerful as a data re-representation technique for images, allowing downstream analysis such as clustering of similar images. Our implementation of a variational autoencoder starts by taking a TCR sequence that is embedded in a fixed-length vector with zero right padding (Fig. 1A). We then use a trainable embedding layer, as described in *Sidhom et. al*, to learn meaning of the amino acids, moving them from a discrete to continuous numerical

space[73]. This is followed by convolutional layers, ultimately reducing the sequence to a latent space that is described as a multi-dimensional unit gaussian distribution. The sequence is then reconstructed from the latent space through the use of deconvolutional layers and the transposition of the trainable embedding layer that was used at the beginning of the network. The network is then optimized with a gradient-descent based algorithm minimizing a reconstruction loss and variational loss, which acts as a mode of regularization. Since this algorithm is primarily trained to minimize the reconstruction loss, the concept of sequence length is learned within the network. However, since TCR sequences are variable length sequences that describe a structural part of the TCR, they can contain length-independent motifs that are required for antigen-specificity, as has been previously demonstrated by *Glanville et. al*[74].

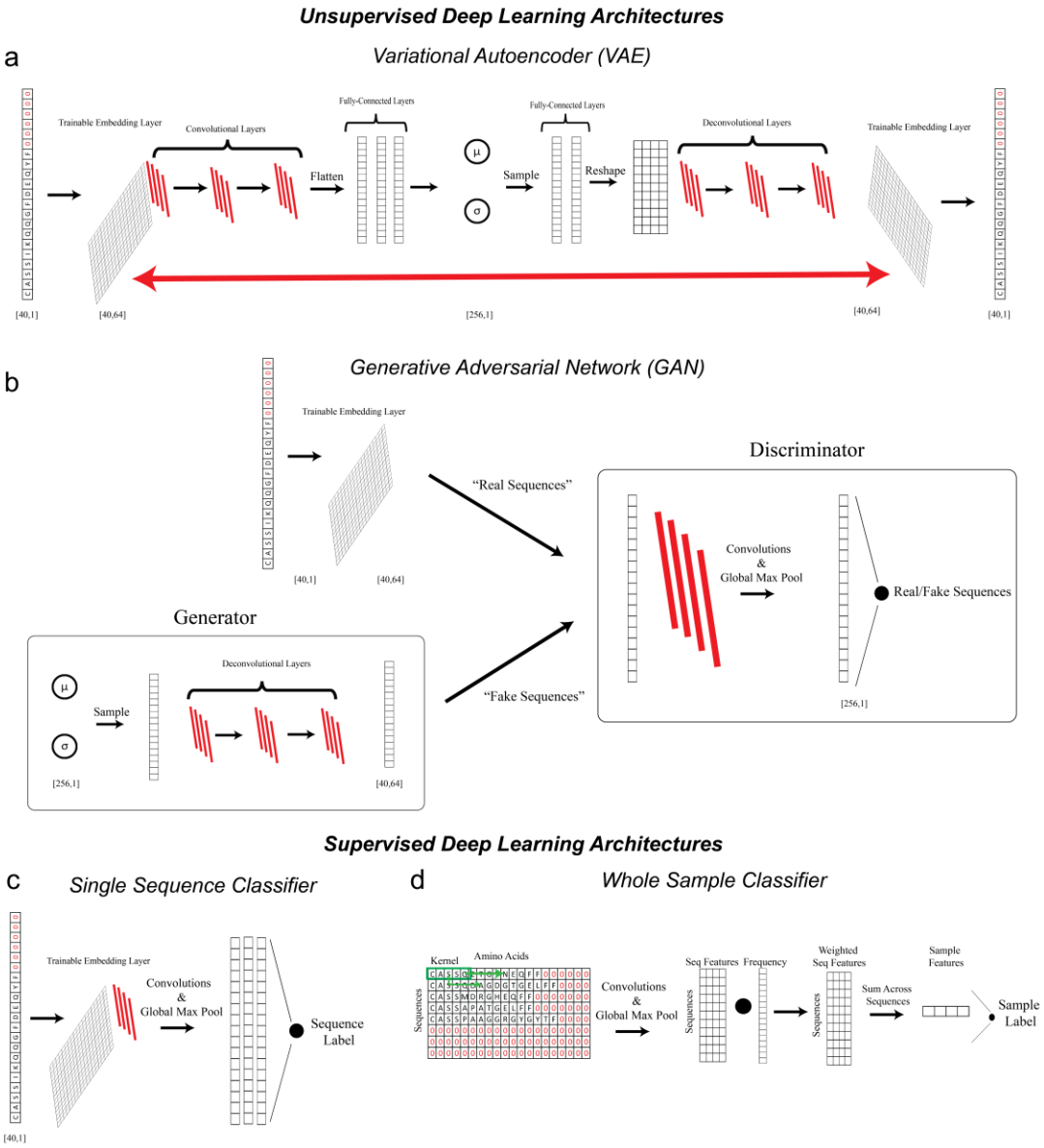


Figure 1. Deep Learning Architectures. (a) The variational autoencoder (VAE) is designed to take as a discrete input the amino acid sequence of the TCR sequence with a right zero-padding scheme. A trainable embedding layer is used to transform the sequence from discrete to continuous numerical domain. Convolutional and fully connected layers transform the sequence into a latent representation that is parametrized by a multi-dimensional unit gaussian. Reconstruction of the sequence occurs via fully connected and deconvolutional layers followed by the transposition of the same trainable embedding layer used at the beginning of the network. **(b)** The generative adversarial network (GAN) consists of the generator and discriminator, separate networks trained with separate objective functions. The generator samples from a multi-dimensional unit gaussian to create a 'fake' TCR sequence. The discriminator learns to distinguish 'real' from 'fake' sequences through one layer of convolutions with a global max pooling operation to provide translational invariance to the network. Of note, the generator's output is the continuous and not discrete representation of the TCR sequence. The latent space used for downstream analysis is the penultimate layer of the discriminator, here described as having dimensions of [256,1]. **(c)** The single sequence classifier follows a conventional convolutional neural network architecture consisting of one convolutional layer with global max pooling and three fully connected layers to a final classification layer. **(d)** The whole sample classifier utilizes a kernel that scans in a horizontal fashion across all sequences in the file resulting in a sequences-by-features tensor. This is then multiplied by the frequency vector for each sequence to derive weighted sequence features. These are then summed across the sequence space to compute sample level features that are fed into a classification layer.

In order to implement an unsupervised deep learning method that could learn features in a length independent fashion, we utilized a GAN architecture (Fig. 1B)[82]. This model consists of two networks, the generator and discriminator that train in an adversarial manner, optimizing separate objective functions (Supplementary Fig. 12). The generator attempts to model the distribution of sequencing data through a generative process where a latent vector is randomly sampled from a multi-dimensional unit gaussian and deconvolutional layers are used to create a TCR sequence. The discriminator is a network that is trained to distinguish between sequences from the 'real' data and sequences from the 'fake' generated data. Aside from being used to model biological sequence data, our implementation of the GAN differs from previously described architectures as it uses a discriminator that has only one convolutional layer with a global max pooling operation to achieve translational invariance to relevant motifs as described by *Sidhom et. al*[73]. In this manner, this network is designed to model the underlying sequence distribution in a length independent manner.

In order to assess how well these unsupervised methods could learn relevant features of TCR sequences, we used a previously published dataset of 2067 sequences for 7 specificities used to train GLIPH, a state-of-the-art method for clustering TCR sequences[74]. We note that both unsupervised deep learning methods are able to cluster sequences of the same specificity (Fig. 2A) and at the whole sample level can make meaningful comparisons between antigen-specific repertoires (Fig 2B). When assessing the specificity of these methods to cluster sequences in groups specific to a given antigen, the VAE demonstrates comparable performance to GLIPH while only requiring the CDR3 β-chain sequence (Fig 2C), demonstrating that 94.48% of clustered TCRs (14% of all sequences clustered) were correctly grouped with other sequences of common specificity. Furthermore, both the VAE and GAN maintain a high clustering accuracy while clustering more sequences. Finally, to assess the characteristics of the clusters formed at various

thresholds, we examined the number of clusters and the variance of the lengths of sequences the clusters contained (Fig 2D). While the VAE and GAN comparably cluster sequences of common specificity and create the same number of clusters doing so, the GAN clusters sequences of different lengths far more than the VAE.

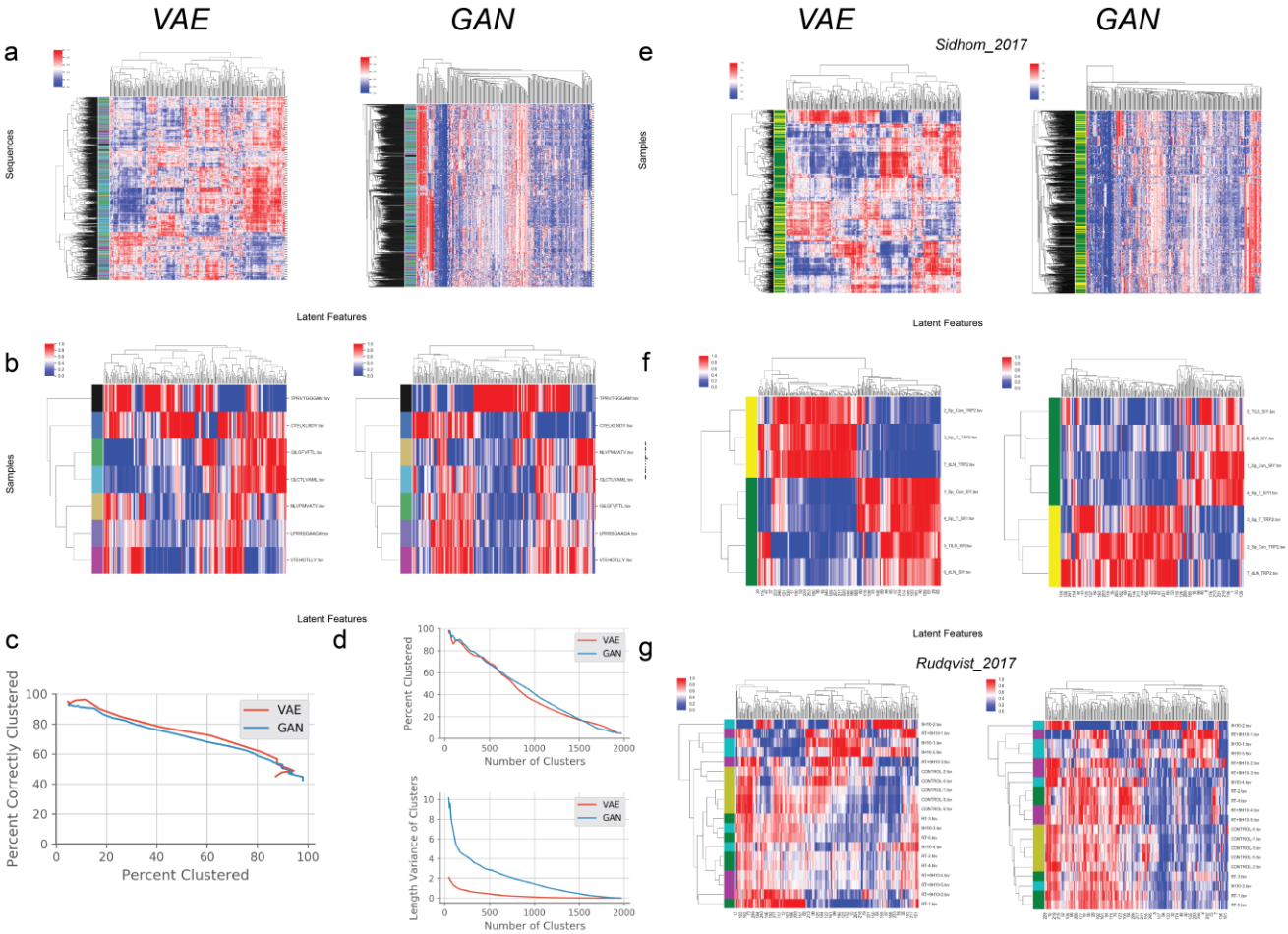


Figure 2. Unsupervised Learning Examples. (a) Heatmaps of sequence-by-features for 7 tetramer sorted populations of antigen-specific T-cells taken from *Glanville_2017* dataset. **(b)** Heatmaps of samples-by-features for 7 tetramer sorted populations of antigen-specific T-cells taken from *Glanville_2017* dataset. **(c)** Clustering specificity of VAE and GAN across various clustering thresholds following hierarchical clustering of sequences by their learned features. **(d)** Number of clusters vs percent of sequences clustered at various clustering thresholds for both VAE and GAN *(top).* Number of clusters vs length variance of sequences within clusters for both VAE and GAN *(bottom)* **(e)** Heatmaps of sequences-by-features for antigen-specific sequences taken from sorted SIY and TRP2 specific T-cells created by VAE and GAN. (SIY = green, TRP2 = yellow) **(f)** Heatmaps of samples-by-features for antigen-specific samples takes from sorted SIY and TRP2 specific T-cells created by VAE and GAN. (SIY = green, TRP2 = yellow) **(g)** Heatmaps of samples-by-features for the tumor-infiltrating lymphocyte (TIL) samples taken from various immunotherapies (Control = yellow, 9H10 = Cyan, RT = Green, RT + 9H10 = Magenta). Heatmaps for sequences-by-features in Supplementary Fig. 14.

When applying these two separate approaches on the *Sidhom_2017* dataset, we note the VAE and GAN are able to comparably cluster antigen-specific sequences as well as antigen-specific samples (Fig. 2 E & F). However, when clustering sequences from their latent representations, we noted that the variances in the length of sequences in a given cluster were much smaller from the VAE as opposed to the GAN (Supplementary Fig. 13) as we would expect, since the VAE learns length dependent features while the GAN does not. When applying these two types of unsupervised approaches to the *Rudqvist_2017* dataset, we note that both methods identify that the control mice have highly conserved structural profiles, as was described in the initial publication (Fig 2G). Our experience using these unsupervised approaches demonstrates they can be useful not only to cluster TCR sequences of high homology but also to compare repertoires at a wholistic level, allowing a method for the first time to quantify similarity between repertoires based on their overall structural composition.

As noted in these datasets, there are often labels associated to TCRSeq, which can either be applied at the sequence or sample level. To accommodate labels at the single sequence level, we designed a simple convolutional neural network that learns sequence specific motifs in a length independent fashion (Fig 1C) to correctly classify sequences by their labels. The second, and arguably the more interesting architecture, is a supervised multi-instance deep learning algorithm that is able to learn meaningful concepts that may lie within large samples of many sequences, either being obscured by the noise of many irrelevant sequences or are weakly predictive at the single sequence level (Fig 1D). This whole sample multi-instance classifier uses convolutional kernels that scan the entire file, learning features for each sequence. These features are then weighted by the frequency of the sequences. Finally, these features are summed across the sequences to give a weighted average of a feature/motif within a sample. We first applied the single

sequence classifier to the *Glanville_2017* dataset and noted there was a weak predictive signature that could differentiate the sequences with better performance for antigens with more TCR sequences (Fig 3A). However, when creating samples *in-silico* that used a given number of unique sequences per sample, we found an increase in predictive performance at the whole sample level as more sequences were used in each sample, demonstrating the ability of a 'weak learner' to become more predictive when provided with more evidence in aggregate (Fig 3B). When applied to the *Sidhom_2017* dataset, we note again that while the sequence level classifier is able to achieve reasonable performance, the whole sample classifier does far better as it is able to use an entire sample of sequences to make a prediction (Fig 3C). This point is further demonstrated in the *Rudqvist_2017* dataset as these samples are from tumor-infiltrating lymphocytes (TIL) where much of the signal comes from background repertoire, making it difficult for a sequence-level classifier to work. However, in the whole sample classifier, we see improved performance with particular improvement in the RT and Control groups as they have profound structural signatures (Fig 3D). Ultimately, the nature of how the T-cell receptor binds its cognate peptide-MHC makes prediction at the single sequence level difficult; however, when multiple instances of a concept are present, the whole sample classifier can leverage this data in order to make more accurate predictions. Finally, given that usually only the β-chain is sequenced, we acknowledge that this presents a considerable limitation in ultimately predicting antigen-specificity as both chains are important for recognition. Thus, we would expect improved performance across all described algorithms when provided both sequencing data for the α and β chains.

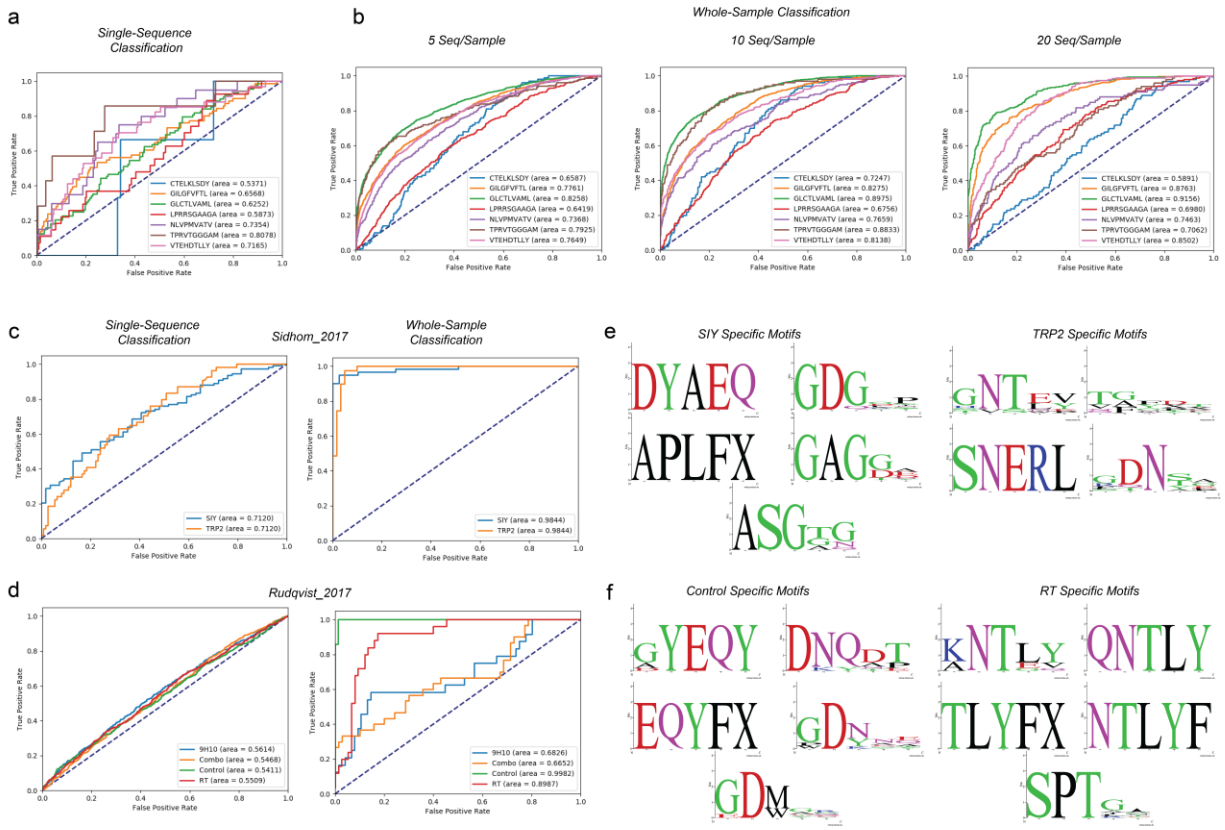


Figure 3. Supervised Learning Examples. (a) Receiver Operating Characteristic (ROC) curve for single-sequence classifier on sequences taken from 7 T-cell specificities from *Glanville_2017* dataset. **(b)** All unique sequence from *Glanville_2017* dataset were allocated randomly without replacement into *in-silico* samples so no sequences were shared among any samples. Samples were created with either 5,10, or 20 sequences/file and whole-sample classifier was used to assess predictive power at the whole sample level. **(c,d)** Receiver Operating Characteristic (ROC) curves for single sequence and whole sample classifier for *Sidhom_2017* and *Rudqvist_2017* datasets. **(e,f)** Representative motifs learned by whole sample classifier for cohorts that had highly predictive structural signatures.

While there is inherent value in predictive models as they can be used as biomarkers for disease, there has been much effort in improving the 'explainability' of deep learning models for the purpose of understanding what the network learned. In the context of TCRSeq, being able to extract knowledge from the network can inform relevant motifs for antigen-specific recognition. Therefore, we established a method by which we could query for differentially used motifs at the cohort level from the trained network allowing us to identify relevant cohort-specific motifs (Fig 3 E & F). By extracting the index along with the value of the feature following its convolution and

global max pooling operation, we can identify where in the sequence a kernel is being maximally activated and use this to derive the motif being learned. These supervised methods demonstrate how predictive models can also be used to generate descriptive results that can inform our understanding of the mechanisms at play.

NGS has become one of the largest sources of big data in the biological sciences, and deep learning is a promising modality for analyzing big data where features need to be learned. In this work, we present DeepTCR, a collection of unsupervised and supervised deep learning approaches to characterize TCRSeq data for both descriptive and predictive purposes. Our unsupervised approaches use newly developed techniques from the deep learning community including the variational autoencoder and generative adversarial network to relate and compare repertoires at the sequence and sample level. We further develop supervised methods in applications where labels can greatly help the learning process, such as when there is buried signal in a large sample of sequences. While DeepTCR has been developed for analyzing TCRSeq data, the concepts introduced within the designed architectures could translate and be applied to a variety of other NGS-based technologies. These types of technologies could yield an entire new area of biomarker discovery as well as improving our own understanding of the complex concepts within the genomic code.

Methods

Data Curation

TCR sequencing files were collected as raw tsv formatted files (Supplementary Fig. 1) from the various sources cited within the manuscript. Sequencing files were parsed to take the amino acid sequence of the CDR3 after removing unproductive sequences. Clones with different nucleotide sequences but the same amino acid sequence were aggregated together under one amino acid sequence and their reads were summed to determine their relative abundance. Within the parsing code, we additionally specified to ignore sequences that used non-IUPAC letters (*,X,O) and removed sequences that were greater than 40 amino acids in length. For the purpose of the algorithm, the maximum length can be altered but we chose 40 as we did not expect any real sequences to be longer than this length.

TCRSeq Quantitative Metrics

Basic TCRSeq analyses were initially done to characterize all samples presented in the manuscript. In order to characterize the distribution of sequences by frequency, we computed the clonality for all samples and characterized the distribution of sequences by their lengths (Supplementary Fig 3). The code used to generate these plots and do this analysis is attached in supplementary material.

Data Transformations

In order to allow a neural network to train from sequence data, we converted the amino acids to numbers between 0-19 representing the 20 possible amino acids. These were then one-hot encoded as to provide a categorical and discrete representation of the amino acids in numerical space. This process was applied prior to all networks being trained.

Training VAE

In order to train the VAE, following creation of the computational graph as described in the manuscript and main figure, we applied an Adam Optimizer (learning rate = 0.001) to minimize a reconstruction loss and a variational loss. The reconstruction loss is the cross-entropy loss between the reconstructed sequence (S) and the one-hot encoded tensor of the input sequence (L) across the *i*th position in the sequence (1). The variational loss is the Kullback–Leibler (KL) divergence between the distributions of the latent variables and a unit gaussian (2).

$$R_{Loss} = -\sum_{i} L_i \log(S_i) \quad (1)$$

$$V_{loss} = D_{KL}(N(\mu(X), \sigma(X)||N(0, I)) \quad (2)$$

The variational loss serves as a regularizer to the network as it prevents overfitting of the network and direct memorization of sequence to latent space and allows for meaningful downstream clustering of the sequences in their latent representation. The variational autoencoder was trained until the reconstruction accuracy over the penultimate 10 iterations was greater than 80%. Features for all sequences were then extracted from the latent space and used to create either heatmaps of sequences by features or a weighted average of the features by the frequency of the sequence was used to construct heatmap of samples by features.

Training GAN

In order to train the VAE, following creation of the computational graph as described in the manuscript and main figure, we applied an RMSProp Optimizer (learning rate = 0.0002) to simultaneously minimize the discriminator (3) and generator (4) loss. The training of a GAN can

be thought of an abstraction of the minimax algorithm where these two networks train in an adversarial fashion until the networks reach a Nash's equilibrium.

$$d_{loss} = -\frac{1}{2}E_{X\sim p_{data}}\log[D(x)] + -\frac{1}{2}E_Z\left[\log\left(1 - D\big(G(z)\big)\right)\right] \quad (3)$$

$$g_{loss} = -\frac{1}{2}E_Z\left[\log D\big(G(z)\big)\right] \quad (4)$$

Since this was the first example we could find of a GAN being used in biological sequence analysis, there were several modifications to the traditional GAN architecture used for image analysis to allow our network to train in a meaningful fashion. The first of these modifications was the input into the discriminator. When a generator is conventionally trained, it outputs an image with a given x by y dimensionality with 3 RGB dimensions that are continuous. However, in our applications, biological sequences are represented in a discrete space and this presented hurdles in getting the generator to create discrete representations. Therefore, the network was trained to output continuous sequence representations that were already embedded in a continuous domain. In a sense, the generator inputs its data in the middle of the discriminator, after the real data has already been embedded in a trainable embedding layer. The second point of alteration to the traditional GAN comes from the need for the discriminator to be a dimensionality reduction operation as oppose to the generator creating real sequences. In order to learn length invariant features, our discriminator has only one convolutional layer where the kernel is global max pooled across the length of the sequence. This operation creates our latent representation which is immediately fed into the final neuron for classification. The nature of this operation results in the generator creating sequences which are a conglomeration of motifs found in the original data as there is no feedback to the generator about length of the sequences in the original distribution of data. Finally, given the simplicity of the discriminator, we found the network was highly

susceptible to mode collapse, a type of failure where the generator outputs only sequence because it successfully fools the discriminator every time. In order to enforce a wide variety of generated motifs and sequences, we applied a feature matching algorithm where we add an additional loss to the generator (5).

$$g_{loss_2} = \frac{1}{n_{features}} \sum \left\| \mu\big(F(x)\big) - \mu\big(F\big(G(z)\big)\big) \right\| \quad (5)$$

This loss is the absolute difference between the average feature values for a batch of real data and fake data. This loss acts as a regularizer to encourage the generator to create diverse batches of sequences, capturing the entire distribution of the data. While this technique worked fairly well, we found the network could occasionally still suffer from mode collapse and further work is needed in the area of using GAN's for short sequences.

Finally, the network was trained in alternating fashion between the generator optimizer and discriminator optimizer over each iteration of the network. Training was halted when the average discriminator loss over the penultimate 10 iterations fell below 1.0 and the generator loss did not fall at least 1% in the penultimate 30 iterations (Supplementary Fig 4). We noted this type of early stopping criterion resulted in the generator initially fooling the discriminator quite easily until the discriminator learned the appropriate features to distinguish real from fake data. At this point, the generator loss would grow and eventually be unable to create sequences capable of fooling the discriminator and the training process was stopped at this point.

Training Single Sequence Classifier

In order to train the single sequence classifier, we followed a traditional conventional neural network architecture where a single translationally invariant convolutional layer was applied to

the sequence followed by three fully connected layers to a final classification layer. The network was trained using an Adam Optimizer (learning rate = 0.001) to minimize the cross-entropy loss between the softmaxed logits and the one-hot encoded representation of the discrete categorical outputs of the network. Training was conducted by using 75% of the data for the training set, and 25% for validation and testing. The validation group of sequences was used to implement an early stopping algorithm.

Training Whole Sample Classifier

Designing an architecture for whole sample multi-instance classification presented unique challenges that were specific to the way TCRSeq data is generated. Not only are the length of individual sequences variable length but the length of the individual files can vary in length as well in terms of number of unique sequences. Since neural networks required fixed-size inputs, this required not only a padding scheme for the sequences but also a padding scheme for the files. When a given dataset was imported, we applied a right zero padding scheme to each of the sequences but then we padded all zero sequences until every file had the same number of sequences. When this tensor is fed into the network, convolutional layers with dimensionality of [1, kernel] are then used to scan across the entire file of sequences. This results in feature values for each sequence in file. Additionally, since TCRSeq is a count-based NGS technology, there are quantitative measurements for each sequence that can be represented as a frequency of the entire file. This frequency is then used to weight the features. At this point, the network takes a sum of the features across all the sequences for a given file, computing a weighted average of all learned features over the entire sample. This vector of weighted average features is then fed directly into the classification layer. The network is trained with an Adam Optimizer (learning rate = 0.001) to minimize the cross-entropy loss between the softmaxed logits and the one-hot encoded

representation of the discrete categorical outputs of the network. Training splits and early stopping algorithms are the same as described above for the single sequence classifier except for in the cases of the *Sidhom_2017* and *Rudqvist_2017* datasets as the number of samples (7 & 20) were too small to create proper sized train/validation/test sets. Therefore, we used a leave-one-out training strategy where we trained on all but one sample until the training loss plateaued and then predicted on the one-out. Due to the generally small nature of these cohorts, in either traditional train/valid/test or leave-one-out splits, we employed monte carlo cross-validation, randomly selecting samples for train/test and iterating a number of times to approximate the predictive signature in the dataset.

Motif Identification

Neural networks are often treated as 'black boxes' where their value is largely in their predictive performance and not in understanding how the neural network is accomplishing its task. However, in the area of the biological sciences, there is not only the desire to create predictive tools but use these tools to inform our own understanding of the mechanisms at play. This area of research is often termed as improving the 'explainability of neural networks. In biological sequence analytics such as DeepTCR, investigators want to be able to extract the features/motifs the neural network learned to accomplish its task. For the supervised learning architectures, we were able to identify motifs the network had learned by extracting the indices of where the kernels were activated following the global max pooling layer. The result of this operation is the network not only extracts the maximum value of a kernel over the length of the sequence but also deduces its position within the sequence. This can be then used to not only pick up which features are activated on a given sequence but where in the sequence this activation occurs, allowing us to identify the motifs that

any given neuron in the net is learning. Sequence logos were created with https://weblogo.berkeley.edu/logo.cgi.

Code and Data availability

DeepTCR was written using Google's TensorFlow™ deep learning library and is available as a python package. Source code, comprehensive documentation, and use-case tutorials along with all data used in this manuscript can be found at https://github.com/sidhomj/DeepTCR. DeepTCR can either be installed directly from Github or from PyPI at https://pypi.org/project/DeepTCR/.

Acknowledgements

The authors thank the MARC/SU2C Foundation for providing financial support for the work of developing algorithmic pipelines presented in this manuscript. We also would like to thank James R. White for editorial assistance and reviewing the DeepTCR codebase.

V. ExCYT: A Graphical User Interface for Streamlining Analysis of High-Dimensional Cytometry Data

Abstract

With the advent of flow cytometers capable of measuring an increasing number of parameters, scientists continue to develop larger panels to phenotypically explore characteristics of their cellular samples. However, these technological advancements yield high-dimensional data sets that have become increasingly difficult to analyze objectively within traditional manual-based gating programs. In order to better analyze and present data, scientists partner with bioinformaticians with expertise in analyzing high-dimensional data to parse their flow cytometry data. While these methods have been shown to be highly valuable in studying flow cytometry, they have yet to be incorporated in a straightforward and easy-to-use package for scientists who lack computational or programming expertise. To address this need, we have developed ExCYT, a MATLAB-based Graphical User Interface (GUI) that streamlines the analysis of high-dimensional flow cytometry data by implementing commonly employed analytical techniques for high-dimensional data including dimensionality reduction by t-SNE, a variety of automated and manual clustering methods, heatmaps, and novel high-dimensional flow plots. Additionally, ExCYT provides traditional gating options of select populations of interest for further t-SNE and clustering analysis as well as the ability to apply gates directly on t-SNE plots. The software provides the additional advantage of working with either compensated or uncompensated FCS files. In the event that post-acquisition compensation is required, the user can choose to provide the program a directory of single stains and an unstained sample. The program detects positive events in all channels and uses this select data to more objectively calculate the compensation matrix. In summary, ExCYT provides a comprehensive analysis pipeline to take flow cytometry data in the

form of FCS files and allow any individual, regardless of computational training, to use the latest algorithmic approaches in understanding their data.

Introduction

Advances in flow cytometry as well as the advent of mass cytometry has allowed clinicians and scientists to rapidly identify and phenotypically characterize biologically and clinically interesting samples with new levels of resolution, creating large high-dimensional data sets that are information rich[83–85]. While conventional methods for analyzing flow cytometry data such as manual gating have been more straightforward for experiments where there are few markers and those markers have visually discernable populations, this approach can fail to generate reproducible results when analyzing higher-dimensional data sets or those with markers staining on a spectrum. For example, in a multi-institutional study, where intra-cellular staining (ICS) assays were being performed to assess the reproducibility of quantitating antigen-specific T cell responses, despite good inter-laboratory precision, analysis, particularly gating, introduced a significant source of variability[86]. Furthermore, the process of manually gating population of interests, besides being highly subjective is highly time consuming and labor intensive. However, the problem of analyzing high-dimensional data sets in a robust, efficient, and timely manner is not one new to the research sciences. Gene expression studies often generate extremely high-dimensional data sets (often on the order of hundreds of genes) where manual forms of analysis would be simply infeasible. In order to tackle the analysis of these data sets, there has been much work in developing bioinformatic tools to parse gene expression data[87]. These algorithmic approaches have just been recently adopted in the analysis of cytometry data as the number of parameters has increased and have proven to be invaluable in the analysis of these high dimensional data sets[88,89].

Despite the generation and application of a variety of algorithms and software packages that allow scientists to apply these high-dimensional bioinformatic approaches to their flow

cytometry data, these analytical techniques still remain largely unused. While there may be a variety of factors that have limited the widespread adoption of these approaches to cytometry data[90], the major hindrance we suspect in use of these approaches by scientists, is a lack of computational knowledge. In fact, many of these software packages (*i.e.,* flowCore, flowMeans, and OpenCyto) are written to be implemented in programming languages such as R that still require substantive programming knowledge. Software packages such as FlowJo have found favor among scientists due to simplicity of use and 'plug-n-play' nature, as well as compatibility with the PC operating system. In order to provide the variety of accepted and valuable analytical techniques to the scientist unfamiliar programming, we have developed ExCYT, a graphical-user interface (GUI) that can be easily installed on a PC/Mac that pulls many of the latest techniques including dimensionality reduction for intuitive visualization, a variety of clustering methods cited in the literature, along with novel features to explore the output of these clustering algorithms with heatmaps and novel high-dimensional flow/box plots.

ExCYT is a graphical user interface built in MATLAB and therefore can either be run within MATLAB directly or an installer is provided that can be used to install the software on any PC/Mac. The software is included with this manuscript. We present a detailed protocol for how to import data, pre-process it, conduct t-SNE dimensionality reduction, cluster data, sort & filter clusters based on user preferences, and display information about the clusters of interest via heatmaps and novel high-dimensional flow/box plots **(Figure 1)**. Axes in t-SNE plots are arbitrary and in arbitrary units and as such as not always shown in the figures for simplicity of the user interface. The coloring of data points in the "t-SNE Heatmaps" is from blue to yellow based on the signal of the indicated marker. In clustering solutions, the color of the data point is based arbitrary on cluster number. All parts of the workflow can be carried out in the single panel GUI

(Figure 2 & Table 1). Finally, we will demonstrate the use of ExCYT on previously published data exploring the immune landscape of renal cell carcinoma in the literature, also analyzed with similar methods. The sample dataset we used to create the figures in this manuscript along with the protocol below can be found at https://premium.cytobank.org/cytobank/projects/875, upon registering an account.

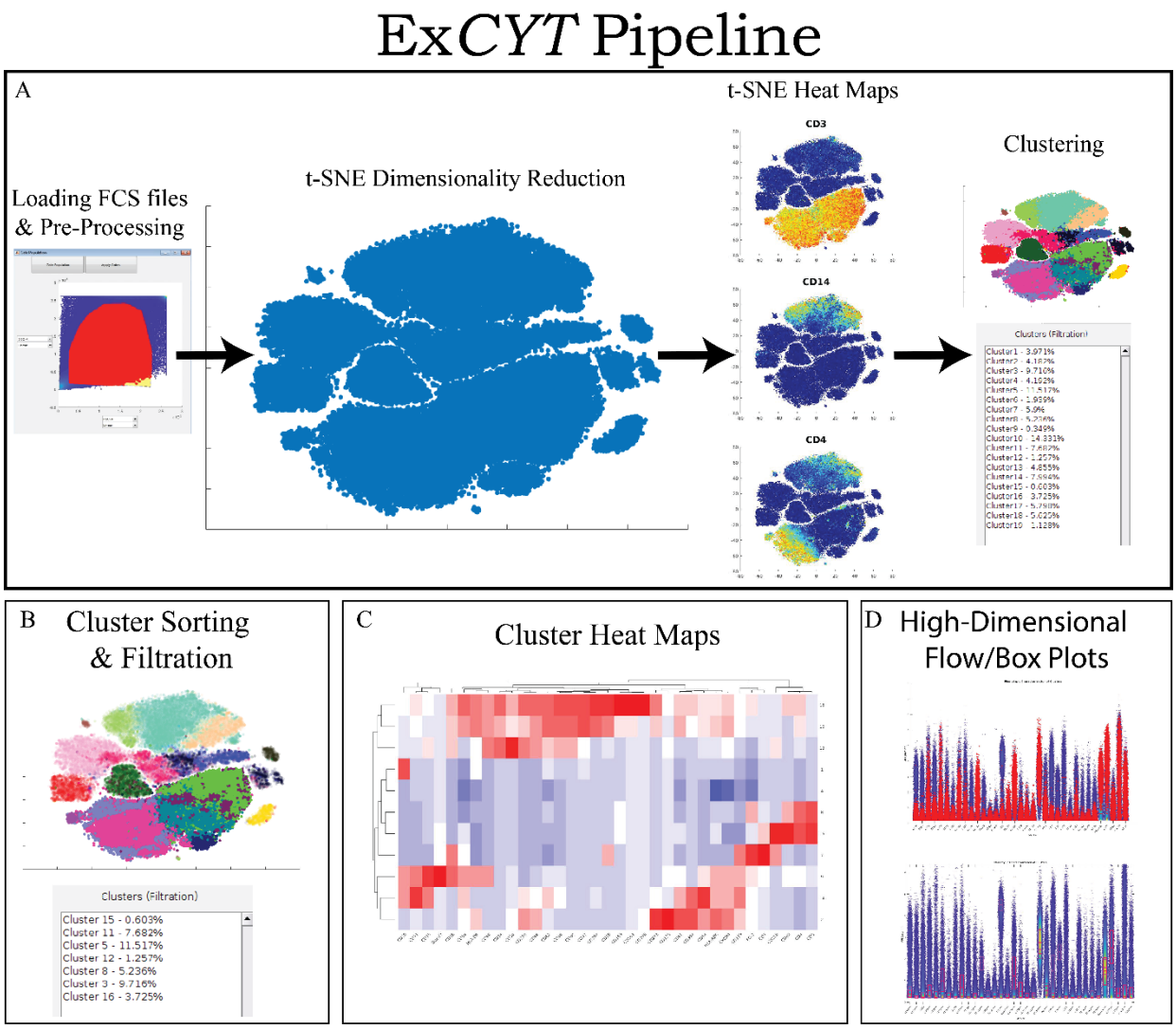


Figure 1: ExCYT Pipeline & Features. (A) ExCYT begins by importing raw FCS data, applying optional compensation, gating, and random subsampling prior to downstream analysis. This ensures all events being analyzed are relevant to the experiment being analyzed. t-SNE dimensionality reduction is then performed to visualize all events and t-SNE heatmaps can be generated to visualize phenotypic distributions. Finally, a variety of clustering algorithms can be applied on either t-SNE transformation or high-dimensional raw data. **(B)** Novel sorting and thresholding features allow users to quickly sort through possibly hundreds of clusters to find ones of interest. **(C)** Heatmaps of clusters can be created to examine how multiple clusters compare to each other as well as which markers co-associate. **(D)** Novel high-dimensional flow/box plots can be generated as a form of back-gating clusters on original data while appreciating the high-dimensional nature of the data.

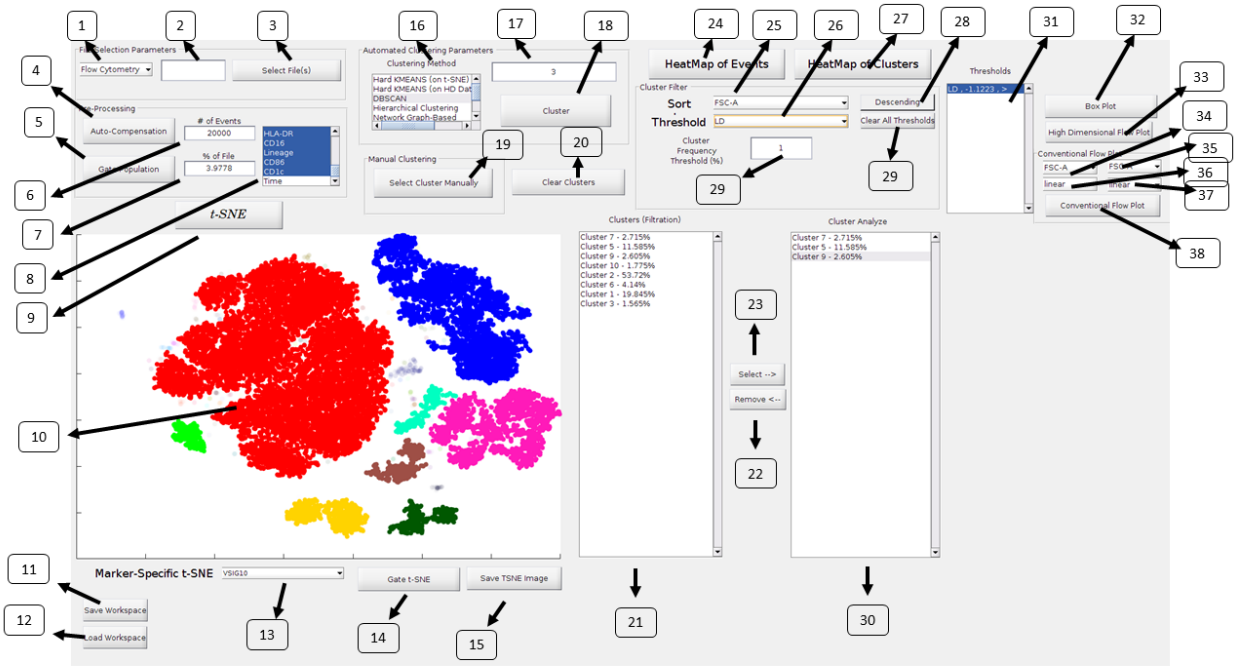


Figure 2: ExCYT Graphical User Interface: The ExCYT graphical user interface allows for a streamline work flow working from the left to right of the panel as the user imports their data, conducts t-SNE dimensionality reduction, clustering, and final cluster analysis and visualization.

No.	Description	Name (in GUI)
1	Select type of Cytometry	NA
2	Random subsampling of raw data	NA
3	Select files for analysis	Select File(s)
4	Auto-compensation of raw data based on directory of single stains provided to software	Auto-Compensation
5	Gating to select events for t-SNE and clustering analysis	Gate Population
6	Random subsampling of gated data (absolute number)	NA
7	Random subsampling of gated data (percent of gated population)	NA
8	Select channels for analysis	NA
9	Run t-SNE dimensionality reduction	t-SNE
10	t-SNE Window	NA
11	Save workspace	Save Workspace
12	Load Workspace	Load Workspace
13	Create t-SNE heatmap on select marker	NA
14	Gate t-SNE to re-do t-SNE analysis of select population	Gate t-SNE
15	Save t-SNE window as image	Save TSNE Image
16	Select Clustering Algorithm	Clustering Method
17	Enter Clustering Parameter for given algorithm	NA
18	Cluster Analysis	Cluster
19	Draw Clusters Manually	Select Cluster Manually
20	Clear All Clusters to redo cluster analysis	Clear Clusters
21	Show Clusters under current filter conditions	Clusters (Filtration)
22	Remove select clusters from Cluster Analyze listbox	Remove <--
23	Add cluster to Cluster Analyze listbox	Select -->
24	Create conventional heatmap of all events in analysis	HeatMap of Events
25	Sort clusters by select marker	Sort
26	Set threshold by select marker	Threshold
27	Create conventional heatmap of select clusters from Cluster Analyze listbox	HeatMap of Clusters
28	Flip order of sort	Ascending/Descending
29	Clear all thresholds	Clear All Thresholds
30	Set frequency threshold for clusters	Cluster Frequency Threshold (%)
31	List of current thresholds active on 'Clusters (Filtration)' listbox	Thresholds
32	High Dimensional Box Plot	High Dimensional Box Plot
33	High Dimensional Flow Plot	High Dimensional Flow Plot
34	Horizontal axis parameter for conventional flow plot	NA
35	Vertical axis parameter for conventional flow plot	NA
36	Data transformation for conventional flow plot on horizontal axis	NA
37	Data transformation for conventional flow plot on vertical axis	NA
38	Create conventional flow plot	Conventional Flow Plot
39	Show Clusters for Analysis	NA

Table 1. Overview of All Functions Present in the ExCYT GUI

Protocol

1. Collecting and Preparing Cytometry Data

1.1. Place all single stains in a folder by themselves and label by the channel name (by fluorophore, not marker).

2. Data Importation & Pre-Processing

2.1. To pause or save throughout this analysis pipeline, use the **Save Workspace** button at the bottom left of the program to save the workspace as a '.MAT' file that can later be loaded via the **Load Workspace** button. Do not run more than one instance of the program at a time. Therefore, when loading a new workspace, make sure to check there is no other instance of ExCYT running.

2.2. To begin analysis pipeline, first select type of cytometry (Flow Cytometry or Mass Cytometry – CYTOF), under the **File Selection Parameters,** and 2000 events to sample from the file. Once data has been successfully imported, a dialogue box will pop up informing the user that the data has been successfully imported.

2.3. Press the **Auto-Compensation** button to conduct an optional auto-compensation step, as done by Bagwell & Adams[91]. Select the directory containing single stains. Select the unstained sample within the user interface dialogue.

2.3.1. Place a forward/side-scatter gate on any of the samples in this directory that will be used to select events to calculate the compensation matrix. It is recommended to use the unstained sample for this purpose. At this point, an algorithm has been implemented to set consistent thresholds at the 99th percentile of the unstained sample to define positive events in each of the single stains to calculate the compensation matrix. When this is finished, a dialogue box will inform the user that the compensation has been performed.

2.4. Next, press **Gate Population** and select the populations of cells of interest, as is the convention in flow cytometry analyses. When population of cells is selected, enter 10,000 events or % of File to use for downstream analysis.

2.5. Next, select the channels to be used for analysis in the listbox in the far right of the Pre-Processing box.

3. **t-SNE Analysis**

3.1. Press the **t-SNE** button to have the program begin start to compute the reduced dimensionality data set for visualization in the window below the t-SNE button. To save image of t-SNE, press **Save TSNE Image**. On a machine with 8 CPU @ 3.4 GHz each and 8 GM RAM this step should take about 2 minutes for 10,000 events, 10 minutes for 50,000 events, and 20 minutes for 100,000 events.

3.2. To create a 't-SNE heatmap', as seen in several CYTOF publications[92,93], select an option from the **Marker-Specific t-SNE** pop-up menu such as CD64 or CD3. A figure will pop up showing a heatmap representation of the t-SNE plot that can be saved for figure generation.

3.3. Select areas of interest in the t-SNE plots by the user for further downstream analyses using the **Gate t-SNE** button.

4. **Cluster Analysis**

4.1. To begin clustering analysis, select option in **Clustering Method** listbox such as DBSCAN with a distance factor of 5 in dialogue box to the right of the listbox. Press the **Cluster** button.

4.2. Use one of the following options for automated clustering algorithms found in the 'Automated Clustering Parameters' panel:

4.2.1. *Hard KMEANS (on t-SNE)*: Apply k-means clustering to the reduced 2-dimensional t-SNE data and requires the number of clusters to be provided to the algorithm[94].

4.2.2. *Hard KMEANS (on HD Data)*: Apply k-means clustering to the original high-dimensional data that was given to the t-SNE algorithm. Once again, the number of clusters needs to be provided to the algorithm.

4.2.3. *DBSCAN:* Apply the clustering method of clustering, called Density-Based Spatial Clustering of Applications with Noise[95] that clusters the reduced 2-dimensional t-SNE data and requires a non-dimensional distance factor that determines the general size of the clusters. This type of clustering algorithm is well suited to cluster the t-SNE reduction as it is able to cluster non-spheroidal cluster that are often present in the reduced t-SNE representation. Additionally, due to the fact that it operates on the 2-dimensional data, it is one of the faster clustering algorithms.

4.2.4. *Hierarchical Clustering:* Apply the conventional hierarchical clustering method to the high-dimensional data where the entire Euclidean distance matrix is calculated between all events before providing the algorithm a distance factor that sets the size of the cluster.

4.2.5. *Network Graph*-Based: Apply a clustering method that has been most recently introduced into analyzing flow cytometry data when there are rare subpopulations that the user wants to detect[93,96]. This method relies on first creating a graph that determines the connections between all events in the data. This step consists of providing an initial parameter to create the graph, which is the number of k-nearest neighbors. This parameter generally governs the size of the clusters. At this point, another dialogue box pops up asking the user to employ one of 5 clustering algorithms that is applied to the graph. These include 3 options to maximize the modularity of the graph, the Danon Method, and a spectral clustering algorithm[96–100]. If one wants a generally faster clustering solution, we recommend Spectral Clustering or the Fast Greedy Modularity Maximization. While

the Modularity Maximization methods along with the Danon method determine the optimal number of clusters, Spectral Clustering requires the number of clusters to be given to the program.

4.2.6. *Self-Organized Map:* Employ an artificial neural network to cluster the high-dimensional data.

4.2.7. *GMM – Expectation Maximization*: Create a Gaussian Mixture Model using Expectation Maximization (EM) technique to cluster the high-dimensional data[101]. This type of clustering method also requires the user to input the number of clusters.

4.2.8. *Variational Bayesian Inference for GMM*: Create a Gaussian Mixture Model but unlike EM, it can automatically determine the number of the mixture components k[102]. While the program does require a number of clusters to be given (larger than the expected number of clusters), the algorithm will determine the optimal number on its own.

4.3. To study a particular area of the t-SNE plot, press the **Select Cluster Manually** button to draw a set of user-defined clusters. Of note, clusters cannot share members (*i.e.,* each event can only belong to 1 cluster).

5. **Cluster Filtration**

5.1. Set(s) of clusters identified either manually or via one of the automatic methods described above can be filter via as follows.

5.1.1. To sort clusters (in the **Cluster Filter** panel) by any of the markers measured in the experiment, select an option from the **Sort** pop-up menu. To set whether the order is ascending or descending, press the **Ascending/Descending** button to the right of the **Sort** pop-up menu. This will update the list of Clusters in the 'Clusters (Filtration)' listbox and re-order them in descending

order of median cluster expression of that marker. The percentage denoted in the 'Clusters (Filtration)' listbox denotes the percent of the population that this cluster represents.

5.1.2. To set a minimum threshold value for a given cluster across a certain channel, select an option from the **Threshold** pop-up menu such as CD65 and set a threshold at around 0.75. Either type a value in the numerical box below the graph or use the slide-bar to set a threshold. Once threshold is set, press **Add Above Threshold** or **Add Below Threshold** to specify the direction of threshold. Once this threshold has been set, it will be listed in the Thresholds box next to the 'Cluster Filter' panel where the marker, the threshold value, and the direction will be listed so the user is aware of which thresholds are currently being applied. Finally, the t-SNE plot will update by blurring out clusters that do not meet the requirements of the filtration and the 'Clusters (Filtration)' listbox will update to show clusters that meet the filtration requirements.

5.1.3. To set a minimum threshold for frequency of a cluster, enter a numerical cut-off in the **Cluster Frequency Threshold (%)** box in the Cluster Filter panel such as 1%.

6. **Cluster Analysis & Visualization**

6.1. To select clusters for further analysis and visualization, select clusters In **Clusters (Filtration)** listbox and press the **Select →** button to move them to the **Cluster Analyze** listbox.

6.2. To create heatmaps of clusters, select the clusters of interest in the **Cluster Analyze** listbox and press the **HeatMap of Clusters** button. When this button is pressed, a figure will pop up containing a heat map along with dendrograms on the cluster and parameter axes. The dendrogram on the vertical axis will group clusters by those that are closely related while the dendrogram on the horizontal axis will group markers that are co-associated. To save heatmap, press **File | Export Setup | Export.**

6.3. To create a 'High Dimensional Box Plot' or 'High Dimensional Flow Plot,' select the clusters of interest in the **Cluster Analyze** listbox and press either the **High Dimensional Box Plot** button or the **High Dimensional Flow Plot** button. These plots can be used to visually assess the distribution of given channels of various clusters across all dimensions.

6.4. To show clusters in traditional 2D flow plots, select the transformation (linear, log10, arcsinh) and channel in the **Conventional Flow Plot** panel and press **Conventional Flow Plot.**

Representative Results

In order to test the usability of ExCYT, we analyzed a curated data set published by Chevrier *et al.* titled 'An Immune Atlas of Clear Cell Renal Carcinoma' where the group conducted CyTOF analysis with an extensive immune panel on tumor samples taken from 73 patients[93]. Two separate panels, a myeloid and lymphoid panel, were used to phenotypically characterize the tumor microenvironment. The objective of our study was to recapitulate the results of their t-SNE and cluster analysis, showing that ExCYT could be used to come to the same conclusions as well as show additional methods of visualization and cluster analysis.

In the original manuscript, the group described 22 T cell clusters identified by the lymphoid panel and 17 cell clusters identified by the myeloid panel. In **Figures 3 & 4** of the publication, the group shows heatmaps of clusters, t-SNE plots with color-coded clustering solutions, and t-SNE heatmaps in subpanels A, B, & C. In order to perform the analysis, we obtained the manually gated data from Cytobank and sampled 2000 events from each file or took the entire file if it had less than 2000 events, following the analysis pipeline illustrated in the original manuscript. At this point, we sampled a total of 100,000 events via our post-gating subsampling parameter, conducted t-SNE analysis, and used a variety of clustering methods to explore the data in various ways.

First, we examined the myeloid panel by following the same analysis pipeline as the original manuscript by completing the t-SNE analysis and creating heatmaps of the various markers (**Figure 3A**). While the original manuscript normalized the t-SNE heatmaps to the 99th percentile of each marker, ExCYT does not do this type of normalization for its heatmaps. However, similar distributions of marker co-expression were observed as described in the original manuscript. We then applied a Network Graph-Based method of clustering the data by creating the graph with 100 k-nearest neighbors and clustering the graph via optimizing the modularity of the graph by using the Fast-Greedy implementation within ExCYT, where we found 19 sub-populations of cells (**Figure 3B**). When comparing the heatmap of these clusters created by ExCYT with the heatmap published in the original manuscript, we noted that we were able to identify similar clusters of myeloid cells (**Figure 3C**). Of note, the original manuscript identified and contrasted two sub-populations of myeloid cells that we identified in our analysis defined by HLA-DR^{int}CD68^{int}CD64^{int}CD36^{+}CD11b^{+} (Cluster 13) and HLA-DR^{+}CD4^{+}CD68^{+}CD64^{+}CD36^{-} CD11b^{-} (Cluster 18). Visualization by high-dimensional box plot of these two populations revealed statistically significant differences (Mann-Whitney) in the six markers mentioned (**Figure 1D**).

Next, we analyzed the lymphoid panel with a more conventional and faster hierarchical clustering approach. This approach yielded similar marker distributions via t-SNE heatmaps (**Figure 4A**). Furthermore, clustering of the data via hierarchical clustering (**Figure 4B**), demonstrated similar clusters of lymphoid cells (**Figure 4C**). Of note, we also identified the unique regulatory T cell population from the original manuscript defined as CD4^{+}CD25^{+}Foxp3^{+}CTLA-4^{+}CD127^{-} (Cluster 17) via our high-dimensional flow plot (**Figure 4D**).

Finally, we wanted to employ a method within ExCYT to quickly and quantitatively assess co-associations among markers. We began by using a hard k-means clustering algorithm to lay

down 5000 clusters on the two-dimensional t-SNE data (**Figure 4E**). We then used the median expression of all the markers of all these clusters to create a heatmap from these clusters (**Figure 4F**). Since these heatmaps cluster rows as well as columns that are similar, this method of abstracting the data by applying a fine mesh of clusters and then creating a heatmap allows us to pick up co-associations easily, such as the co-association of Tim-3, PD-1, CD38, and 4-1BB.

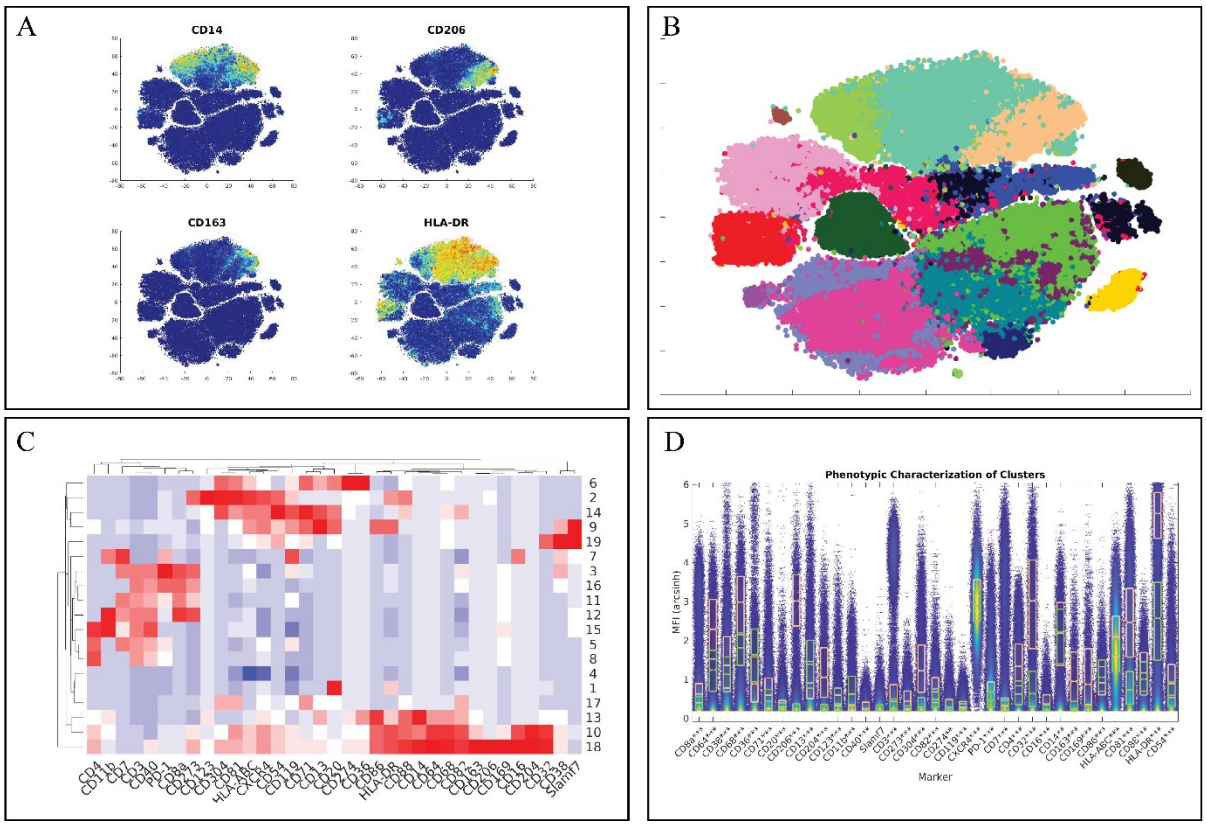


Figure 3: Recapitulation of Myeloid Sub-Populations from Chevrier *et al.* (A) Token t-SNE heatmaps of myeloid panel **(B)** t-SNE plot of myeloid panel color coded by Network-Graph clustering algorithm **(C)** Heatmap of clusters identified by clustering solution on myeloid panel **(D)** Comparative high dimensional box plot comparing contrasting myeloid subpopulations (Clusters 13 & 18) referenced in original manuscript

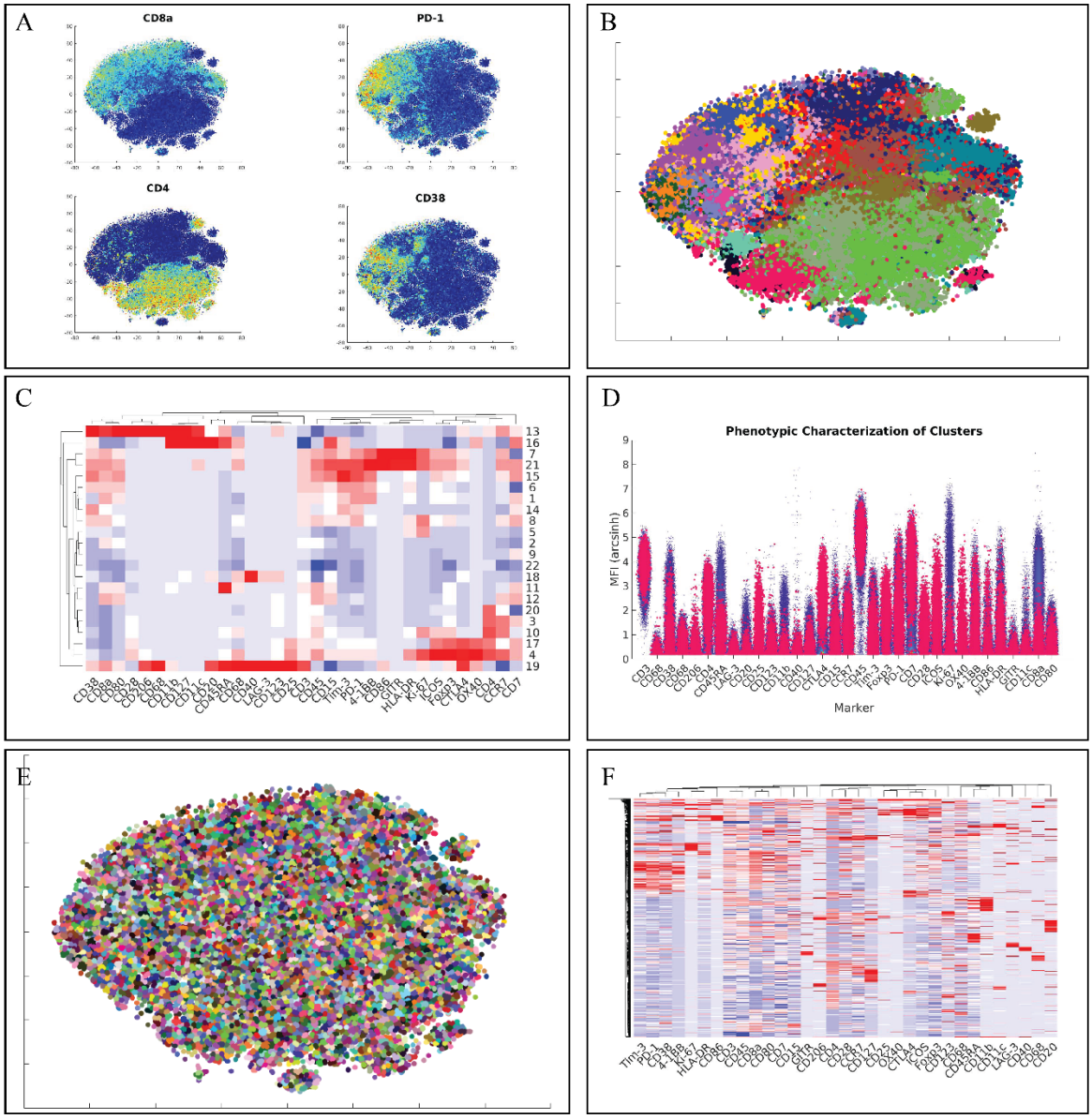


Figure 4: Recapitulation of Lymphoid Sub-Populations from Chevrier *et al.* (A) Token t-SNE heatmaps of lymphoid panel **(B)** t-SNE plot of lymphoid panel color coded by hierarchical clustering algorithm **(C)** Heatmap of clusters identified by clustering solution on lymphoid panel **(D)** High dimensional flow plot of identified regulatory T cell population (Cluster 17) in original manuscript **(E)** Clustering solution of 5000 cluster hard k-means analysis on t-SNE data **(F)** Heatmap of clusters identified by k-means clustering solution on lymphoid panel showing marker co-associations.

Discussion

Here we present ExCYT, a novel graphical user interface running MATLAB-based algorithms to streamline analysis of high-dimensional cytometry data, allowing individuals with no background in programming to implement the latest in high-dimensional data analysis algorithms. The availability of this software to the broader scientific community will allow scientists to explore their flow cytometry data in an intuitive and straightforward workflow. Through conducting t-SNE dimensionality reduction, applying a clustering method, being able to sort/filter through these clusters quickly, and make flexible, customizable heatmaps and high-dimensional flow/box plots, scientists will be able to not only understand the uniquely defined subpopulations in their samples but will be able to create visualizations that are intuitive and easily understood by their colleagues.

While the program is flexible in handling a variety of data types (conventional flow cytometry vs mass cytometry), there are a few considerations for optimal utility of the program. The first of these is regarding the data quality, specifically of flow cytometry data. Proper compensation and resolution of overlapping emission spectra is of paramount importance. Poorly compensated data can inadvertently lead to false co-associations of markers and formation of clusters that are not of true biological significance. Therefore, it is highly advisable that the input data is of sound quality before proceeding with the t-SNE analysis and further downstream analysis. Furthermore, use of the automatic compensation algorithm implemented in ExCYT requires clear single stains for all channels in order to accurately calculate the compensation parameters.

Another important consideration for use of ExCYT is when concatenating multiple FCS files into one analysis (as demonstrated in this manuscript), they must be comparable across all channels. First, this means that the same panel needs to be used across all samples *and* that there

is no drift between samples across all channels. For example, if one were to read two samples on separate days and stained CD8 in FITC on both days but the voltage of the cytometer was set differently on one day resulting in a slightly shifted CD8 population, one could generate false clusters in the downstream analysis, as this shift was generated as a function of instrument variation and not due to biological significance. While future versions of ExCYT may be able to normalize samples to their single stains, at this point, careful consideration must be made that FCS files can be compared to each other before importing them into ExCYT.

Finally, the process of clustering is not one that is absolute/rigid. Different clustering algorithms and parameters can generate different clustering solutions. Whether the solution of the algorithm is appropriate is for the user to determine by synthesizing their understanding of the biology with the clustering solution. For example, when understanding the immune environment of tumors, one may be interested in macroscopic clusters (*i.e.,* T cells vs B cells vs Myeloid cells) while another may be interested in subpopulations of macroscopic clusters. The resolution of the clusters is determined by the user and therefore, no single clustering solution is 'correct.' This is one of the main advantages of using the high dimensional flow plots available in ExCYT. The ability to visualize the distribution of a given cluster across all channels can help the user determine whether they have clustered in not only a biologically relevant way but in a way that is relevant to the scientific question being asked in the experiment. While our goal is to provide a plethora of methods used in the literature to cluster high-dimensional flow cytometry data while providing additional methods of clustering, we recommend using methods such as k-means and DBSCAN to explore the data via quickly iterating on cluster number and size and move towards network-graph and gaussian-mixed model approaches for more robust but more time-consuming approaches.

Given these considerations, ExCYT is still a highly flexible and valuable tool for exploring high dimensional cytometry data, and offers unique/differentiating features than other available packages available to conduct this type of analysis **(Table 2)**. First, ExCYT differentiates itself over most flow cytometry analysis approaches utilizing dimensionality reduction and clustering algorithms by its ability to be used without any scripting/programming knowledge. Additionally, by aggregating many clustering algorithms cited throughout the literature, we believe we provide the most options for clustering data. Finally, our unique feature of cluster filtration and sorting along with display via novel high dimensional flow plots, allows users to explore the characteristics of their clusters quickly and efficiently, making the process of 'discovering' rare subpopulations simple and efficient.

Name of Software/Package	**ExCYT**	**CYT**	**FCS Express**	**flowCore**	**openCyto**	**FlowMeans**
Program Type	Matlab	Matlab	Stand-Alone Application	R	R	R
Price to User	Free	Free	$1,000	Free	Free	Free
Graphical User Interface	Yes	Yes	Yes	No	No	No
Dimensionality Reduction Techniques	t-SNE	t-SNE,PCA	t-SNE, PCA, SPADE	none	none	none
Clustering Algorithms	K-Means DBSCAN Hierarchical Clustering Self-Organized Map Multiple Network-Graph Based Methods GMM - EM GMM - Variational Bayesian Inference	K-Means GMM - EM Single Network-Graph Based Method (Phenograph)	K-Means	none	automation of manual gating workflow	K-Means
Ability to Sort/Filter Clusters	Yes	No	No	No	No	No
High Dimensional Flow Plots	Yes	No	No	No	No	No

Table 2: Overview of Software-assisted Flow Cytometry Analysis Solutions

VI. Convolving Pre-Trained Convolutional Neural Networks at Various Magnifications to Extract Diagnostic Features for Digital Pathology

Abstract

Deep learning is an area of artificial intelligence that has received much attention in the past few years due to both an increase in computational power with the increased use of graphics processing units (GPU's) for computational analyses and the performance of these class of algorithms on visual recognition tasks. They have found utility in applications ranging from image search to facial recognition for security and social media purposes. Their continued success has propelled their use across many new domains including the medical field, in areas of radiology and pathology in particular, as these fields are thought to be driven by visual recognition tasks. In this paper, we present an application of deep learning, termed 'transfer learning', using ResNet50, a pre-trained convolutional neural network (CNN) to act as a 'feature-detector' at various magnifications to identify low and high level features in digital pathology images of various breast lesions for the purpose of classifying them correctly into the labels of normal, benign, in-situ, or invasive carcinoma as provided in the ICIAR 2018 Breast Cancer Histology Challenge (BACH).

Introduction

While artificial intelligence and machine learning have revolutionized many scientific fields, perhaps no other method has had the widespread adoption and practical use as much as deep artificial neural networks, or otherwise known as 'deep learning.' Of note, deep learning has had a profound impact on tasks associated with visual recognition, bringing about technologies capable of object classification and image recognition[66]. With the reduction to practice in many fields such as social media and communication technologies, there has been a recent advent to bring deep learning into the medical field. One of the initial applications of deep learning in the medical field was by a group at Stanford led by Sebastian Thrun who used transfer learning to classify skin lesions into benign, non-neoplastic, and malignant subtypes[10]. This proof of concept has led an initiative to bring deep learning into other visual recognition tasks in the medical field including radiology and digital pathology[103,104]. Not only have these approaches been shown effective in providing diagnostic power, comparable to medical professionals, but they also have shown promise to help learn features possibly missed by humans that can help differentiate various pathologies[105].

Despite the promise of deep learning for applications in the medical field, due to the novelty of these applications, there exists little labeled data for supervised machine learning. However, prior work in the fields of deep learning has shown the power of a technique called 'transfer learning,' the idea being that pre-existing designed architectures (i.e. ResNet50, AlexNet, VGG16) that have been trained on possibly millions of images for over 1000 image classes can serve as 'professional feature detectors' for new visual recognition tasks where data may not be as abundant. While it may seem far fetched that a convolutional neural network (CNN) trained to recognize dogs and cats could recognize relevant features in medical imaging, Sebastian

Thrun and his group demonstrated this exact concept could be utilized, generating results even better than when training a CNN de-novo to diagnose skin lesions[10]. In this manuscript, we propose a method by which one can use ResNet50, a pre-trained CNN, as a feature detector for classification of normal, benign, in-situ, and invasive breast carcinoma. We propose a method by which convolving this pre-trained net at various magnifications of labeled pathology image tiles can serve to detect low and high-level features in digital pathology that can be ultimately used for the task of lesion classification.

Methodology

Data-Set

For the task of classifying various types of breast pathology images, we were provided a set of 400 images (100 per class) of normal, benign, in-situ, and invasive breast carcinoma. Additionally, with a set of 20 whole slide images (WSI), we were able to extract an additional 648 image tiles of invasive, in-situ, and normal breast tissue with the assistance of a pathologist. For the remaining of the manuscript, we will refer to the first set as Data Set A and the latter set as Data Set B.

Color Normalization & Image Augmentation

In order to account for the variation in color we conducted an image color augmentation routine in which we characterized each of the tiles provided in the first part of this competition using the Reinhard method[106] , which transforms the RGB color image to the CIELAB colorspace. We then converted the CIELAB projections into the HSV colorspace and then selected 5 representative tiles from the spectrums observed at the 10th, 25th, 50th, 75th, and 90th percentiles for both hue (H) and value (V). We then made 10 color transformation for each input image to the 10

representative tiles identified above using the Reinhard color transfer method, thus constituting our image color augmentation approach.

Neural Network Architecture

In initial tests using ResNet50 as a pre-trained feature extractor for the pathology image tiles, we noticed a lack of resolution and detail as many of these pre-trained CNN's take fairly small images (244 x 244 pixels). In order to maintain resolution of important features of the pathology while also approaching the problem with the inspiration of how a pathologist examines slides, we decided to convolve ResNet50 at two magnifications (100x100 μm, 400x400 μm) with a stride length that was half of the kernel length (**Figure 1**). The output of each convolution was a [n-windows, 2048] feature map on which we took the maximum value for each ResNet50 feature. This feature extraction was done with the Keras implementation of ResNet50 where the top layer was not included and an average pooling was done to obtain the 2048 features for each window. At the end of this step, each magnification has a 2048-element vector that reflects the presence of a given ResNet50 feature at a given magnification of the pathology image tile. At this point, we concatenate the 2048-element vector from both magnifications used to create a [2,2048] tensor that is then flattened and used as an input to a trainable fully-connected layer of 512 neurons, trained with a 20% dropout rate, followed by the final multi-class classification layer for the 4 output classes (normal, benign, in-situ, and invasive) with a softmax activation layer. Creation/training of this part of the neural network was implemented in tensorflow.

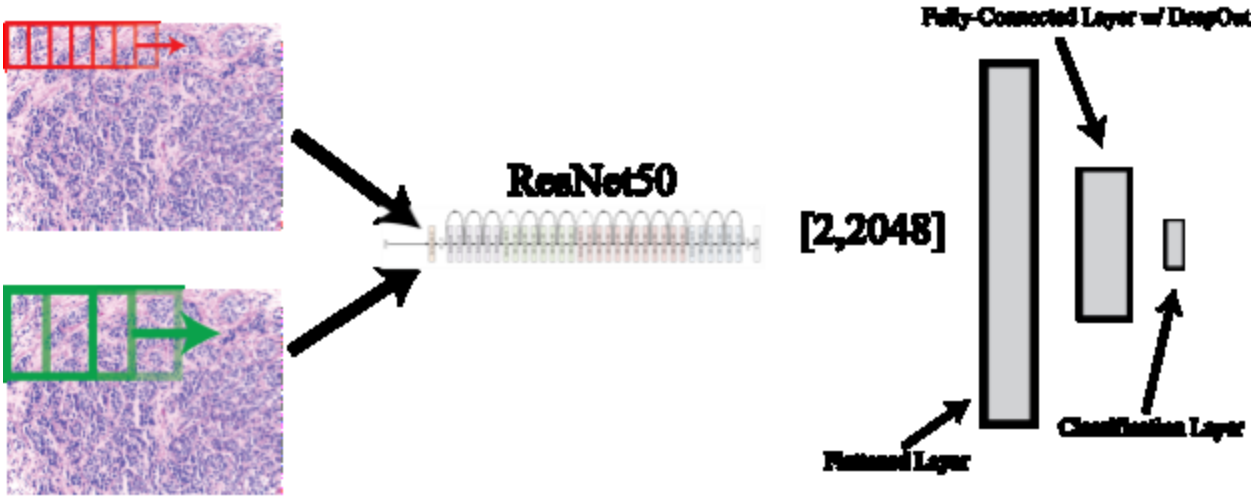


Figure 1. Feature Extraction & Neural Net Architecture. ResNet50 was convolved across a given image tile at two magnifications to generate a ResNet50 feature map for each magnification for each image tile. These feature maps were then flattened into a fully connected layer, on which a secondary fully-connected layer (512 neurons) followed by a final output classification layer with a softmax activation.

Training

In order to train the fully-connected layers of the CNN, we split up our data-set into training, validation, and test sets. We used an 90/5/ split across these respective sets, 5 batches of images per epoch, with 100-fold cross-validation to assess performance via receiver operating characteristic (ROC) curves, confusion matrices to examine individual class accuracy, and measured overall accuracy of the algorithm. We implemented an early stopping approach where after a minimum of 100 epochs, we stopped training when the validation loss did not decrease by 2% in the previous 50 epochs or the validation accuracy did not increase by 5% in the previous 200 epochs. After 100-fold training, we averaged the weights of each training session to arrive at final kernel and bias weights for the fully connected layers of the graph.

Results

We wanted to assess the ability of neural network to learn the underlying pathologies from two separate sources of pathology images so we conducted a series of experiments where we ran 100x Monte Carlo cross validation with a train/test split of 90/10 percent. We then varied the data available for training, as shown in the figures below while assessing performance on the original Data Set A. We varied the training data based on combination of Data Sets A & B (as described in the methods above) and the use of the image color augmentation approach we implemented, which is described in the methods.

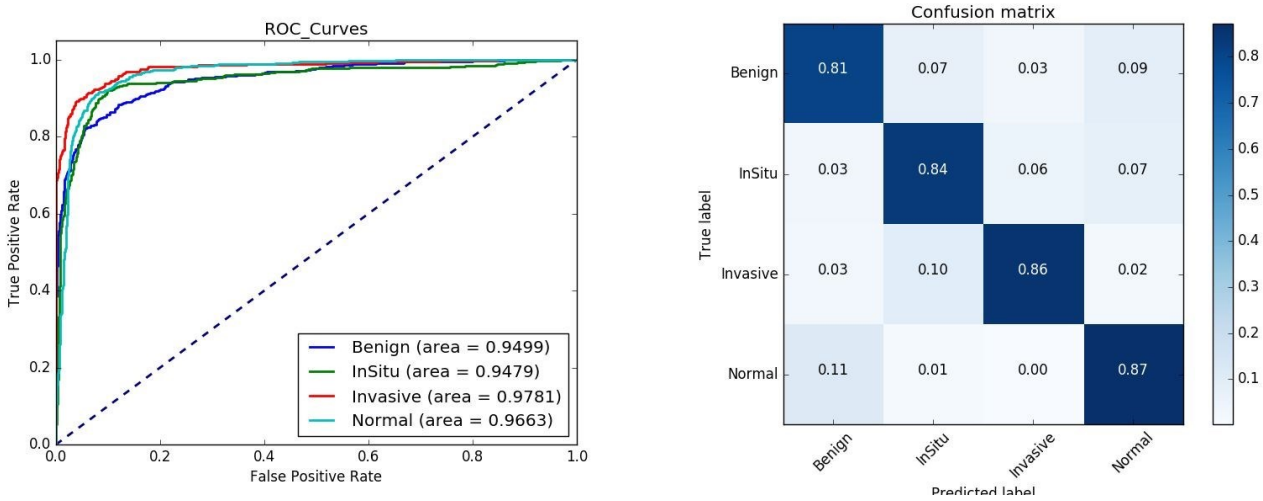


Figure 2. Train on Data Set A, Test on Data Set A - No Image Color Augmentation

With these 400 images, we obtained an overall accuracy of 84.3% across all four classes.

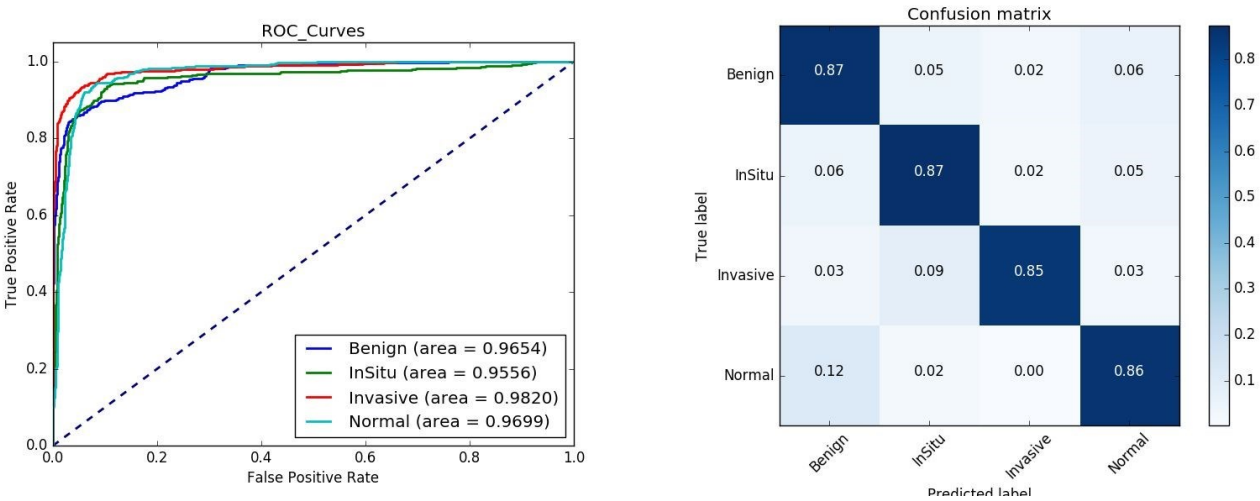


Figure 3. Train on Data Set A, Test on Data Set A - Image Color Augmentation

When introducing augmented images into the training set, we saw an increase in overall accuracy up to 86.2%.

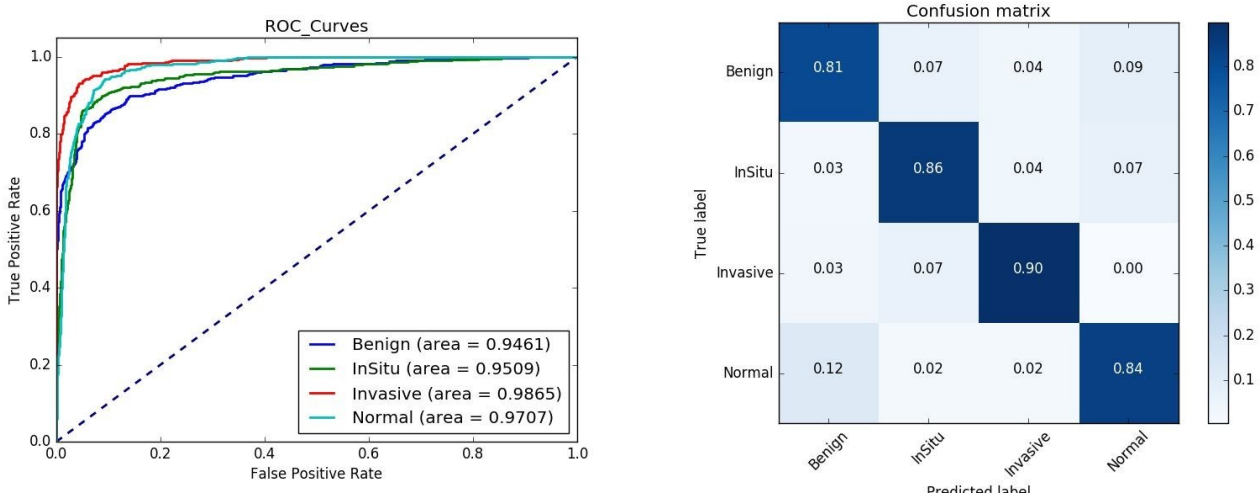


Figure 4. Train on Data Set A & Data Set B, Test on Data Set A - No Image Color Augmentation

When introducing a second data set into our training set, we saw an increase in overall accuracy up to 85.3%.

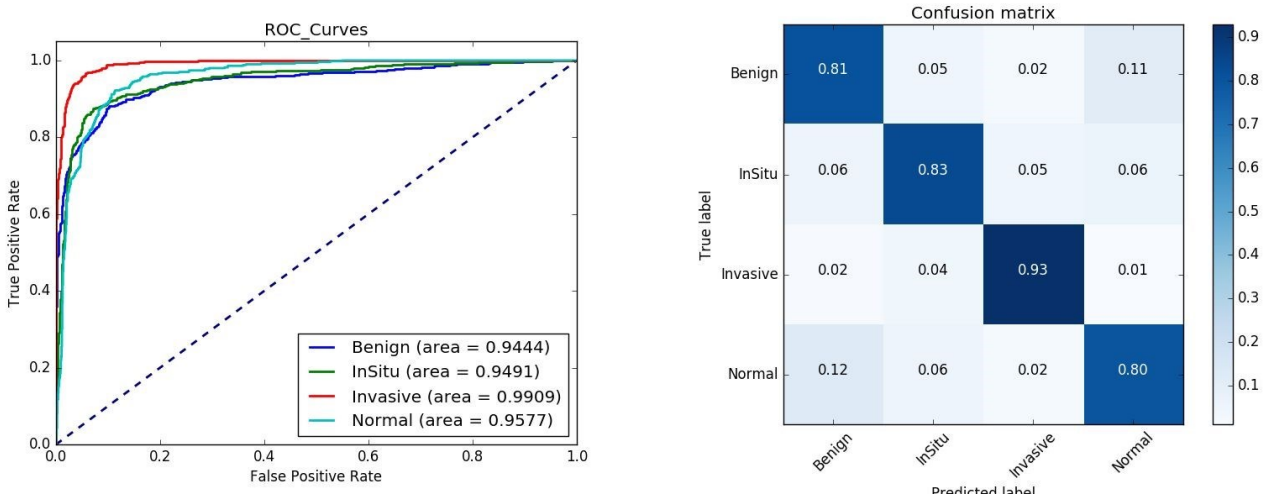


Figure 5. Train on Data Set A & Data Set B, Test on Data Set A - With Image Color Augmentation

When introducing color augmentation into the training of both data sets A and B, we observed an overall accuracy of 84.4%.

In comparing figure 2 and 3, we observed that the introduction of our image color augmentation approach increased the accuracy in the validation data by a small factor of approximately 2%. In comparing figure 2 and 4, we also noted a small increase of approximately 1 % in performance when including the additional images (Data Set B) derived from the WSI data also provided in the ICIAR 2018 BACH challenge. Somewhat surprisingly, when comparing figure 2 to figure 5, we did not observe an increase in performance with image color augmentation applied to the training set that include data sets A & B.

Conclusion

Here we present a method of convolving ResNet50, a pre-trained convolutional neural network, across pathology image tiles at various magnifications to identify low and high-level features that can be later fed into trainable fully-connected layers for the purpose of accurately classifying various types of breast lesions. While we were able to get a respectable accuracy through this

method, we believe we were still making mistakes in classification that would be considered obvious to a pathologist. While we never explored the idea of re-training or fine-tuning the weights *within* ResNet50, we hypothesize this approach may be able to push the performance further beyond what we have been able to show here.

Furthermore, when examining the clinical utility of such an algorithm, we felt future directions should focus differentiating cancer from non-cancer primarily as algorithms such as this one will initially have value in the capacity of being screening tests. We believe in order to make this a viable, clinically useful algorithm, future efforts should be placed in ruling out normal images with a high degree of confidence, regardless of the false positive rate for cancer detection by the algorithm.

VII. Conclusion

In this work, we demonstrate the power of artificial intelligence to have tremendous potential for generating insights in a variety of datasets within the field of cancer immunology. Through the development of ImmunoMap and DeepTCR, we exhibit how two types of machine learning approaches can provide structural insights into T-cell repertoire. Ultimately, tools such as these will be useful to understanding the antigenic-composition of the adaptive immune response in cancer. AI-MHC attempts to understand the other side of the immune synapse in attempting to better predict Class I and Class II antigens for the purpose of improved neoantigen prediction. Finally, we demonstrate how conventional high-dimensional analytics can be packaged into useful tools for the broader community through the development of ExCYT, a software package for high-dimensional cytometry analysis. I believe the concepts developed within this doctoral work will be foundational for a comprehensive assessment of all the players that are involved in the immune system-cancer interaction that will lead to breakthroughs in understanding that advance therapy and treatment of cancer patients.

VIII. References

1. Sun, C. C. *et al.* Rankings and symptom assessments of side effects from chemotherapy: insights from experienced patients with ovarian cancer. *Supportive Care in Cancer* **13,** 219–227 (2005).

2. O'Shaughnessy, J. Extending Survival with Chemotherapy in Metastatic Breast Cancer. *The Oncologist* **10,** 20–29 (2005).

3. Shankaran, V. *et al.* IFNγ and lymphocytes prevent primary tumour development and shape tumour immunogenicity. *Nature* **410,** 1107–1111 (2001).

4. Korman, A., Peggs, K. & in immunology, A. J. Checkpoint blockade in cancer immunotherapy. (2006).

5. Pardoll, D. M. The blockade of immune checkpoints in cancer immunotherapy. *Nature Reviews Cancer* **12,** 252–264 (2012).

6. Topalian, S., Hodi, F. & of …, B. J. Safety, activity, and immune correlates of anti–PD-1 antibody in cancer. (2012).

7. Snyder, A., Makarov, V. & of …, M. T. Genetic basis for clinical response to CTLA-4 blockade in melanoma. (2014).

8. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436 (2015).

9. Gulshan, V., Peng, L., Coram, M., Stumpe, M. & Jama, W. D. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. (2016).

10. Esteva, A., Kuprel, B., Novoa, R., Ko, J. & Nature, S. Dermatologist-level classification of skin cancer with deep neural networks. (2017).

11. Boyd, S. D. *et al.* Measurement and Clinical Monitoring of Human Lymphocyte Clonality by Massively Parallel V-D-J Pyrosequencing. *Science Translational Medicine* **1,** 12ra23-12ra23 (2009).

12. Vollmers, C., Sit, R. V., Weinstein, J. A., Dekker, C. L. & Quake, S. R. Genetic measurement of memory B-cell recall using antibody repertoire sequencing. *Proceedings of the National Academy of Sciences* **110,** 13463–13468 (2013).

13. Robins, H. S. *et al.* Comprehensive assessment of T-cell receptor β-chain diversity in αβ T

cells. *Blood* **114,** 4099–4107 (2009).

14. Wang, C. *et al.* High throughput sequencing reveals a complex pattern of dynamic interrelationships among human T cell subsets. *Proceedings of the National Academy of Sciences* **107,** 1518–1523 (2010).

15. Jackson, K. *et al.* Human Responses to Influenza Vaccination Show Seroconversion Signatures and Convergent Antibody Rearrangements. *Cell Host & Microbe* **16,** 105–114 (2014).

16. Logan, A. C. *et al.* High-throughput VDJ sequencing for quantification of minimal residual disease in chronic lymphocytic leukemia and immune reconstitution assessment. *Proceedings of the National Academy of Sciences* **108,** 21194–21199 (2011).

17. Grupp, S. A. *et al.* Chimeric Antigen Receptor–Modified T Cells for Acute Lymphoid Leukemia. *The New England Journal of Medicine* **368,** 1509–1518 (2013).

18. Carreno, B., Magrini, V. & … B.-H. M. A dendritic cell vaccine increases the breadth and diversity of melanoma neoantigen-specific T cells. (2015).

19. Aerwood, Emerson, R. & Immunology …, S. D. Tumor-infiltrating lymphocytes in colorectal tumors display a diversity of T cell receptor sequences that differ from the T cells in adjacent mucosal tissue. (2013).

20. Ecology, J. L. Partitioning diversity into independent alpha and beta components. (2007).

21. Stewart, J., Lee, C., Ibrahim, S. & Molecular …, W. P. A Shannon entropy analysis of immunoglobulin and T cell receptor. (1997).

22. Venturi, V., Kedzierska, K., Tanaka, M. & of …, T. S. Method for assessing the similarity between subsets of the T cell receptor repertoire. (2008).

23. Victor, T.-S. C., Rech, A., Maity, A. & Nature, R. R. Radiation and dual checkpoint blockade activate non-redundant immune mechanisms in cancer. (2015).

24. Glanville, J., Huang, H., Nau, A., Hatton, O. & Nature, W. L. Identifying specificity groups in the T cell receptor repertoire. (2017).

25. Dash, P., Fiore-Gartland, A., Hertz, T. & Nature, W. G. Quantifiable predictive features define epitope-specific T cell receptor repertoires. (2017).

26. Madi, A., Poran, A., Shifrut, E. & Elife, R.-Z. S. T cell receptor repertoires of mice and humans are clustered in similarity networks around conserved public CDR3 sequences. (2017).

27. Oelke, M., Maus, M., Didiano, D. & medicine, J. C. Ex vivo induction and expansion of antigen-specific cytotoxic T cells by HLA-Ig–coated artificial antigen-presenting cells. (2003).

28. Perica, K., Medero, A., rai, … C. Y. & and Medicine, B. Nanoscale artificial antigen presenting cells for T cell immunotherapy. (2014).

29. Oelke, M., Moehrle, U., Chen, J. & Cancer …, B. D. Generation and purification of CD8+ melan-A-specific cytotoxic T lymphocytes for adoptive transfer in tumor immunotherapy. (2000).

30. Carlson, C., Emerson, R. & Nature …, S. A. Using synthetic templates to design an unbiased multiplex PCR assay. (2013).

31. Suessmuth, Y., Mukherjee, R., Watkins, B. & Blood, K. D. CMV reactivation drives post-transplant T cell reconstitution and results in defects in the underlying TCRβ repertoire. (2015).

32. Needleman, S. & of molecular biology, W. C. A general method applicable to the search for similarities in the amino acid sequence of two proteins. (1970).

33. of molecular biology, A. S. Amino acid substitution matrices from an information theoretic perspective. (1991).

34. Carey, A., Gracias, D. & of …, T. J. Rapid evolution of the CD8+ TCR repertoire in neonatal mice. (2016). doi:10.4049/jimmunol.1502126

35. Perica, K., Bieler, J., Schütz, C. & ACS …, V. J. Enrichment and expansion with nanoscale artificial antigen presenting cells for adoptive immunotherapy. (2015). doi:10.1021/acsnano.5b02829

36. Blank, C., Brown, I., Peterson, A., iotto & research, I. Y. PD-L1/B7H-1 inhibits the effector phase of tumor rejection by T cell receptor (TCR) transgenic CD8+ T cells. (2004). doi:10.1158/0008-5472.CAN-03-3259

37. Kyewski, B. & Immunol., K. L. A central role for central tolerance. (2006).

38. Hogquist, K., Baldwin, T. & Immunology, J. S. Central tolerance: learning self-control in the thymus. (2005).

39. Piccirillo, C. & in immunology, T. A. Cornerstone of peripheral tolerance: naturally occurring CD4+ CD25+ regulatory T cells. (2004).

40. Chodon, T., Comin-Anduix, B. & Cancer …, C. B. Adoptive transfer of MART-1 T-cell receptor transgenic lymphocytes and dendritic cell vaccination in patients with metastatic

melanoma. (2014). doi:10.1158/1078-0432.CCR-13-3017

41. Wang, F., Bade, E., Kuniyoshi, C., Spears, L. & Cancer …, J. G. Phase I trial of a MART-1 peptide vaccine with incomplete Freund's adjuvant for resected high-risk melanoma. (1999).

42. Rosenberg, S., Zhai, Y. & of the …, Y. J. Immunizing patients with metastatic melanoma using recombinant adenoviruses encoding MART-1 or gp100 melanoma antigens. (1998).

43. Hanson, H., Donermeyer, D., Ikeda, H. & Immunity, W. J. Eradication of established tumors by CD8+ T cell adoptive immunotherapy. (2000).

44. Morgan, R. A. *et al.* Cancer Regression in Patients After Transfer of Genetically Engineered Lymphocytes. *Science* **314,** 126–129 (2006).

45. Vatakis, D., Koya, R. & of the …, N. C. Antitumor activity from antigen-specific CD8 T cells generated in vivo from genetically engineered human hematopoietic stem cells. (2011). doi:10.1073/pnas.1115050108

46. Dudley, M. & Cancer, R. S. Adoptive-cell-transfer therapy for the treatment of patients with cancer. (2003).

47. Speiser, D., Miranda, R. & of …, Z. A. Self antigens expressed by solid tumors do not efficiently stimulate naive or activated T cells: implications for immunotherapy. (1997). doi:10.1084/jem.186.5.645

48. Schumacher, T. & Science, S. R. Neoantigens in cancer immunotherapy. (2015). doi:10.1126/science.aaa4971

49. Lennerz, V., Fatho, M. & of the …, G. C. The response of autologous T cells to a human melanoma is dominated by mutated neoantigens. (2005). doi:10.1073/pnas.0500090102

50. Gros, A. *et al.* Prospective identification of neoantigen-specific lymphocytes in the peripheral blood of melanoma patients. *Nature Medicine* **22,** nm.4051 (2016).

51. Rudqvist, N., Pilones, K. & Immunol …, L. C. Radiotherapy and CTLA-4 blockade shape the TCR repertoire of tumor-infiltrating T cells. (2018).

52. Cell, G. MHC-dependent antigen processing and peptide presentation: providing ligands for T lymphocyte activation. (1994).

53. Ott, P. A. *et al.* An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature* **547,** 217–221 (2017).

54. Timm, J., Lauer, G. & of …, K. D. CD8 epitope escape and reversion in acute HCV infection. (2004).

55. Łuksza, M. *et al.* A neoantigen fitness model predicts tumour response to checkpoint blockade immunotherapy. *Nature* (2017). doi:10.1038/nature24473

56. Vyas, J., der Veen, V. A. & Immunology, P. H. The known unknowns of antigen processing and presentation. (2008).

57. Neefjes, J., Jongsma, M. & Reviews …, P. P. Towards a systems understanding of MHC class I and MHC class II antigen presentation. (2011).

58. Nielsen, M. *et al.* Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Science* **12,** 1007–1017 (2003).

59. Andreatta, M. & Bioinformatics, N. M. Gapped sequence alignment using artificial neural networks: application to the MHC class I system. (2015).

60. Nielsen, M., Lundegaard, C. & computational …, B. T. Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan. (2008). doi:10.1371/journal.pcbi.1000107

61. Zeng, H., Edwards, Liu, G. & Bioinformatics, G. D. Convolutional neural network architectures for predicting DNA–protein binding. (2016).

62. Sabour, S., Frosst, N. & in Information, H. G. Dynamic routing between capsules. (2017).

63. Han, Y. & bioinformatics, K. D. Deep convolutional neural networks for pan-specific peptide-MHC class I binding prediction. (2017).

64. Kim, Y. *et al.* Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions. *BMC bioinformatics* **15,** 241 (2014).

65. Jensen, K., Andreatta, M., Marcatili, P. & … B. S. Improved methods for predicting peptide binding affinity to MHC class II molecules. (2018). doi:10.1111/imm.12889

66. Krizhevsky, A., Sutskever, I. & in neural, H. G. Imagenet classification with deep convolutional neural networks. (2012).

67. Matsumura, M., Fremont, D. & Science, P. P. Emerging principles for the recognition of peptide antigens by MHC class I molecules. (1992). doi:10.1126/science.1323878

68. Sidhom, J.-W. *et al.* ImmunoMap: A Bioinformatics Tool for T-Cell Repertoire Analysis.

Cancer Immunology Research **6,** canimm.0114.2017 (2017).

69. Szolek, A., Schubert, B., Mohr, C. & … S. M. OptiType: precision HLA typing from next-generation sequencing data. (2014).

70. Drijfhout, J., Brandt, R., D'Amaro, J. & immunology, K. W. Detailed motifs for peptide binding to HLA-A∗ 0201 derived from large random sets of peptides using a cellular binding assay. (1995).

71. Sidney, J., Peters, B. & BMC …, F. N. HLA class I supertypes: a revised and updated classification. (2008).

72. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* **33,** 831–838 (2015).

73. Sidhom, J.-W., Pardoll, D. & Baras, A. AI-MHC: an allele-integrated deep learning framework for improving Class I & Class II HLA-binding predictions. *bioRxiv* 318881 (2018). doi:10.1101/318881

74. Glanville, J. *et al.* Identifying specificity groups in the T cell receptor repertoire. *Nature* **547,** 94–98 (2017).

75. Dash, P. *et al.* Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* **547,** 89–93 (2017).

76. Buermans, H. P. J. & den Dunnen, J. T. Next generation sequencing technology: Advances and applications. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease* **1842,** 1932–1941 (2014).

77. ... *et al.* Tumor and microenvironment evolution during immunotherapy with nivolumab. (2017).

78. Gerlinger, M., Quezada, S. & of …, P. K. Ultra-deep T cell receptor sequencing reveals the complexity and intratumour heterogeneity of T cell clones in renal cell carcinomas. (2013).

79. Wang, G., Dash, P. & translational …, M. J. T cell receptor αβ diversity inversely correlates with pathogen-specific antibody levels in human cytomegalovirus infection. (2012).

80. Pu, Y., Gan, Z., Henao, R., Yuan, X. & in neural …, L. C. Variational autoencoder for deep learning of images, labels and captions. (2016).

81. Doersch, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

82. Goodfellow, I., Pouget-Abadie, J. & in neural …, M. M. Generative adversarial nets. (2014).

83. Benoist, C. & Science, H. N. Flow cytometry, amped up. (2011). doi:10.1126/science.1206351

84. Ornatsky, O., Bandura, D., Baranov, V. & of …, N. M. Highly multiparametric analysis by mass cytometry. (2010).

85. Tanner, S., Bandura, D. & and Applied …, O. O. Flow cytometer with mass spectrometer detection for massively multiplexed single-cell biomarker assay. (2008).

86. Maecker, H., Rinfret, A. & BMC …, D. P. Standardization of cytokine flow cytometry assays. (2005).

87. Brazma, A. & letters, V. J. Gene expression data analysis. (2000).

88. Pyne, S., Hu, X., Wang, K. & of the …, R. E. Automated high-dimensional flow cytometric data analysis. (2009). doi:10.1073/pnas.0903028106

89. Ge, Y. & Bioinformatics, S. S. flowPeaks: a fast unsupervised clustering for flow cytometry data via K-means and density peak finding. (2012).

90. systems research, V. V. Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. (2000).

91. Bagwell, C. & of the of, A. E. Fluorescence spectral overlap compensation for any number of flow cytometry parameters. (1993).

92. Lavin, Y., Kobayashi, S., Leader, A., Amir, E. & Cell, E. N. Innate immune landscape in early lung adenocarcinoma by paired single-cell analyses. (2017).

93. Chevrier, S., Levine, J., Zanotelli, V. & Cell, S. K. An immune atlas of clear cell renal cell carcinoma. (2017).

94. Hartigan, J. & of the Wong, - MAC. Algorithm AS 136: A k-means clustering algorithm. (1979). doi:10.2307/2346830

95. Ester, M., Kriegel, H., Sander, J. & Kdd, X. X. A density-based algorithm for discovering clusters in large spatial databases with noise. (1996).

96. Levine, J., Simonds, E., Bendall, S. & Cell, D. K. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. (2015).

97. Blondel, V., Guillaume, J. & of statistical …, L. R. Fast unfolding of communities in large networks. (2008).

98. Martelot, L. E. & Journal, H. C. Fast multi-scale detection of relevant communities in large-scale networks. (2013).

99. review MEJ, E. Fast algorithm for detecting community structure in networks. (2004). doi:10.1103/PhysRevE.69.066133

100. Barbara, H. J., CA & of California, U. An efficient matlab algorithm for graph partitioning. (2004).

101. processing magazine, M. T. The expectation-maximization algorithm. (1996).

102. 대한토목학회지 B. C. Pattern recognition and machine learning, 2006. (2012).

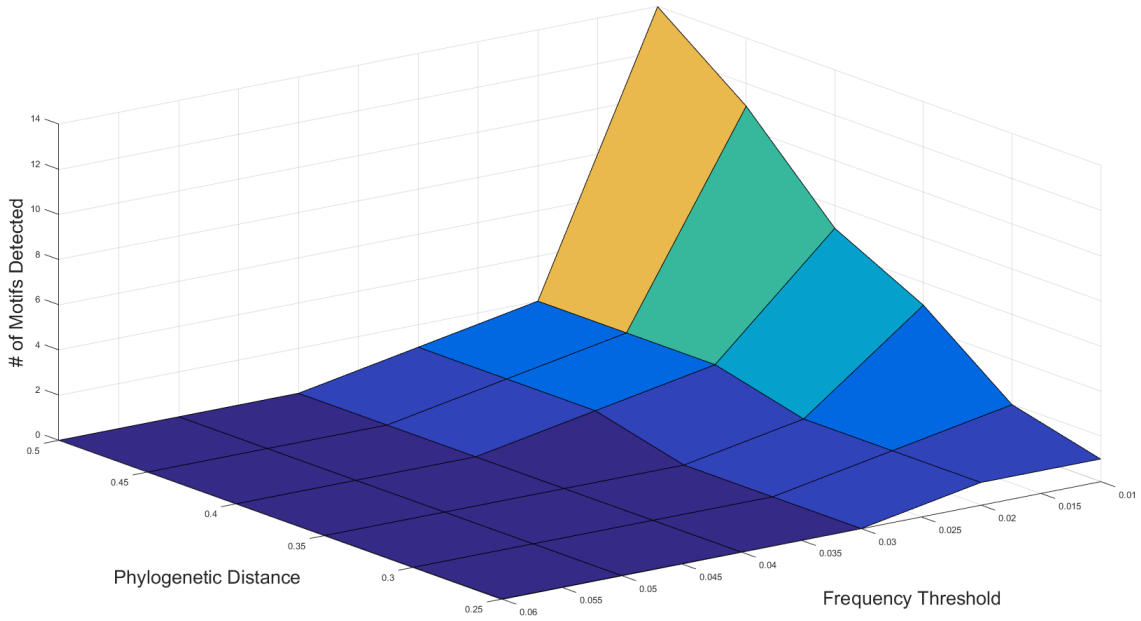
103. Djuric, U., Zadeh, G., Aldape, K. & precision oncology, D. P. Precision histology: how deep learning is poised to revitalize histomorphology for personalized cancer care. (2017).

104. Lee, J., Jun, S., Cho, Y. & journal of …, L. H. Deep learning in medical imaging: general overview. (2017). doi:10.3348/kjr.2017.18.4.570

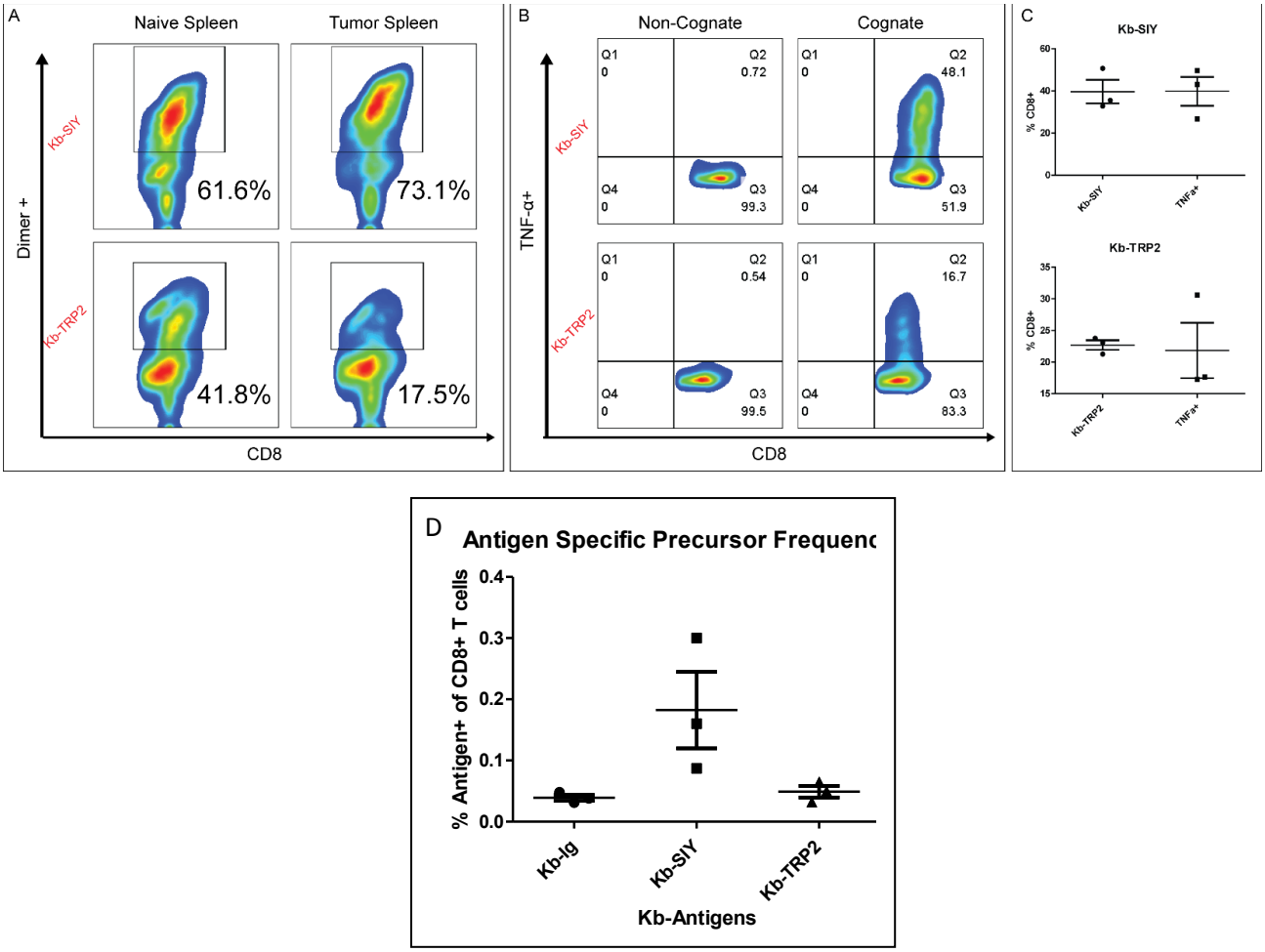
105. Kao, C. & preprint arXiv:1707.05809, M. L. A Novel Deep Learning Architecture for Testis Histology Image Classification. (2017).

106. Reinhard, E., Adhikhmin, M. & graphics and …, G. B. Color transfer between images. (2001).

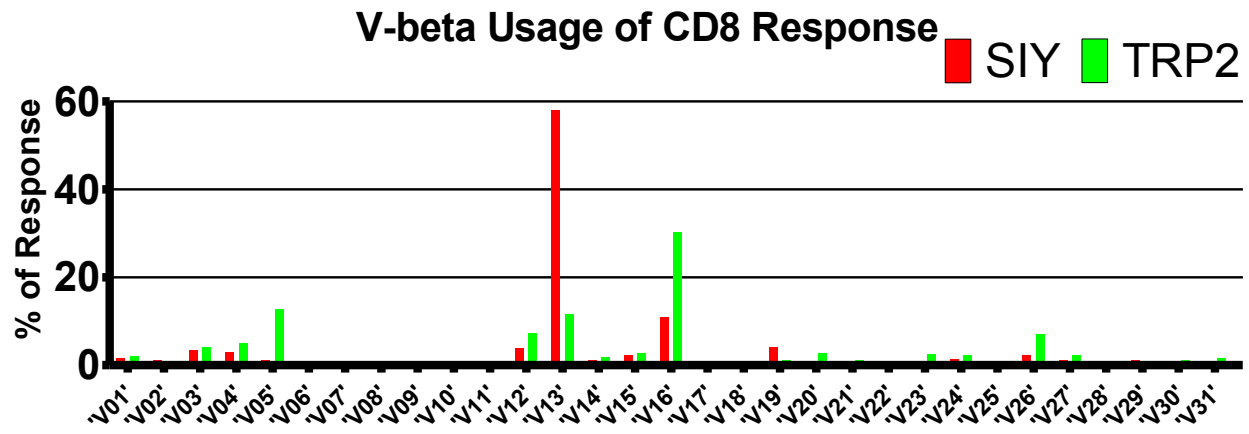
IX. Appendices



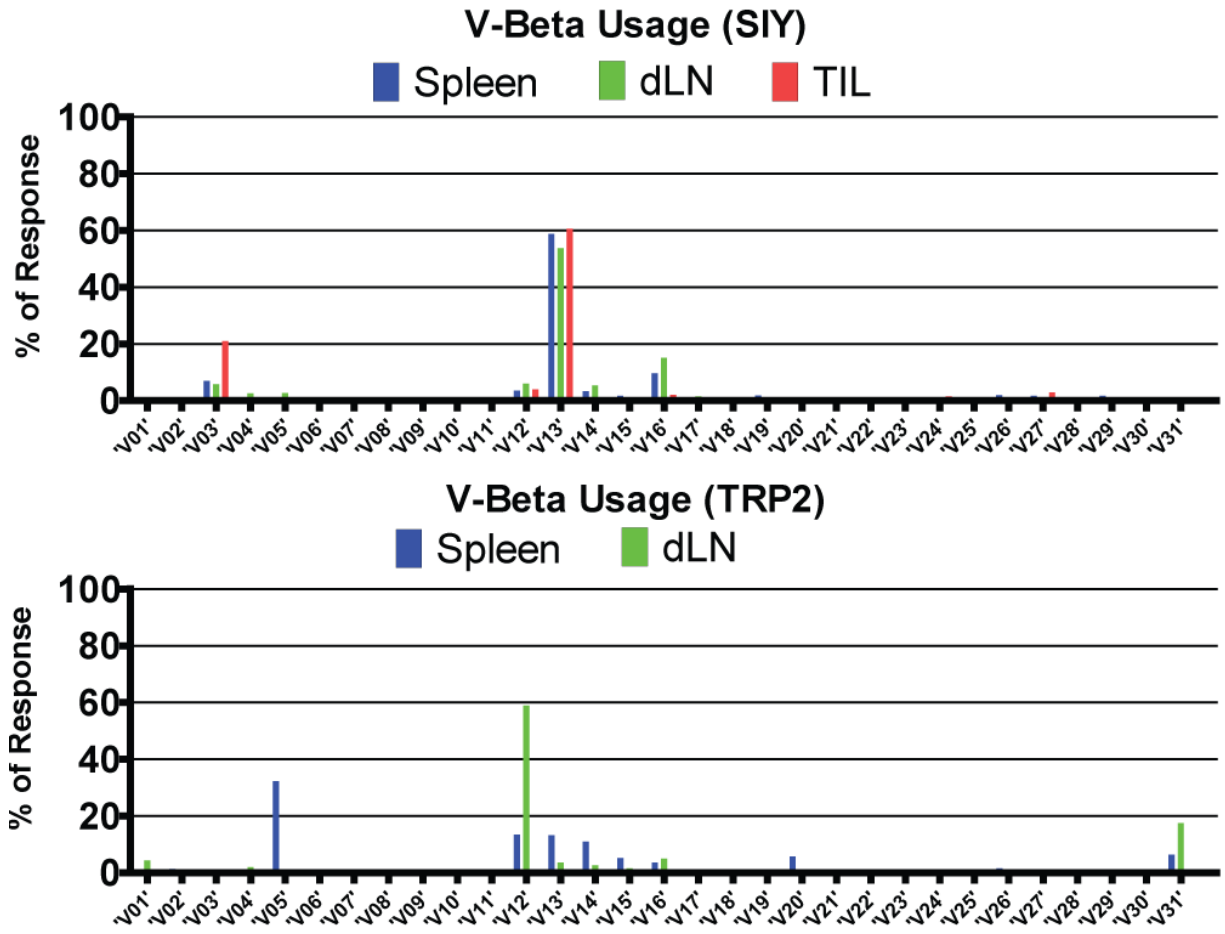
Supplementary Figure 1. In order to set thresholds for motif detection, two variables were optimized on a Naïve Adult B6 CD8 Repertoire (taken from Adaptive Biotechnologies ImmunoSeq Sample Data). Motif detection was completed across a range of phylogenetic distances and frequency thresholds and number of motifs detected was monitored. Since our analysis was looking for dominant motifs above what is present in an unexpanded population, we chose a frequency threshold of 0.03 and Phylogenetic Distance threshold of 0.35, at which 0 motifs were detected in Naïve B6 background.



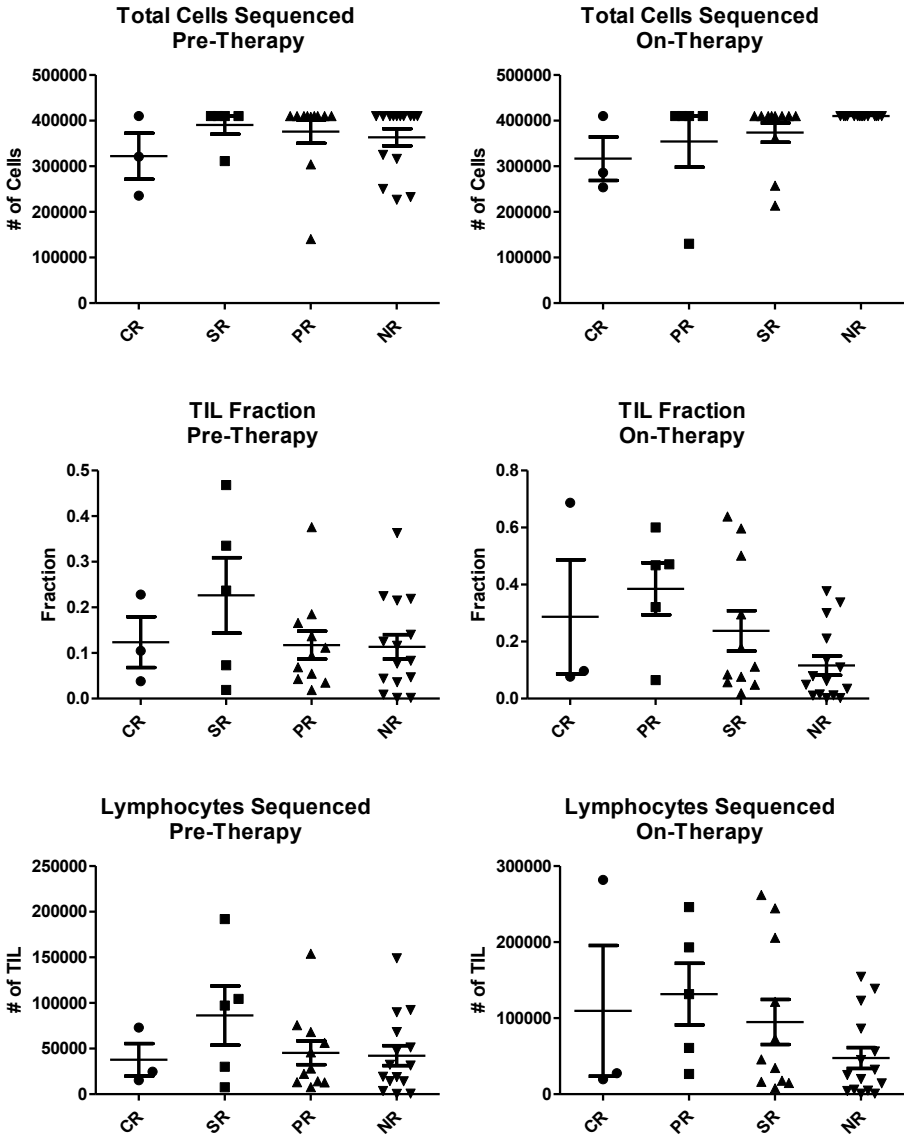
Supplementary Figure 2. A) FACS analysis of antigen-specific CD8 T cells on D7 in Naïve and Tumor-bearing lymphoid organs. B) ICS staining of antigen-specific CD8 T Cells confirming specificity and functionality. C) Comparison of Dimer+ and TNF+ CD8. D) Antigen-specific CD8 T cells staining from splenic CD8 T cells directly ex vivo compared to unloaded Kb-Ig staining. N = 3, Statistical 2-tailed T-test.



Supplementary Figure 3. V-beta usage for Naïve Kb-SIY & Kb-TRP2 Response



Supplementary Figure 4. V-beta usage for Tumor-Bearing Kb-SIY & Kb-TRP2 Response in Various Lymphoid Organs

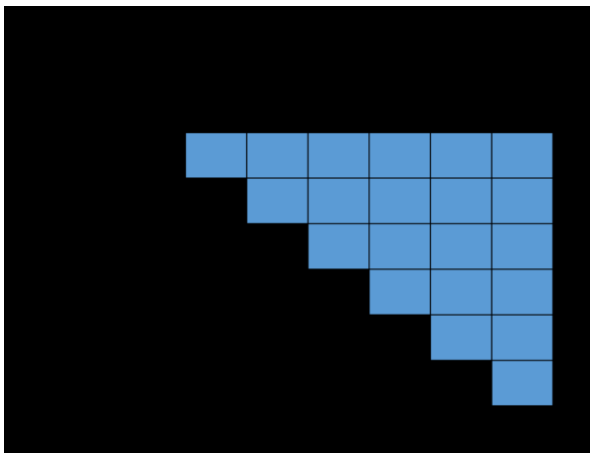


Supplementary Figure 5. In order to determine number of tumor-infiltrating lymphocytes that were sequenced for each patient, Adaptive reported amount of total DNA in nanograms that underwent sequencing and based on the assumption of 6.5pgDNA per cell, we were able to calculate the number of total cells that underwent sequencing. Furthermore, Adaptive calculated a %TIL metric based on number of non-recombined to recombined sequence reads. With this information, we were able to deduce the number of starting lymphocytes that were sequenced for each patient.

A)

$$Sequence\ Distance = (1 - \frac{Score12}{Score11})(1 - \frac{Score12}{Score22})$$

$$Mapped\ Sequence\ Distance = \frac{1}{1 + \left[(1 - \frac{Score12}{Score11})(1 - \frac{Score12}{Score22})\right]}$$

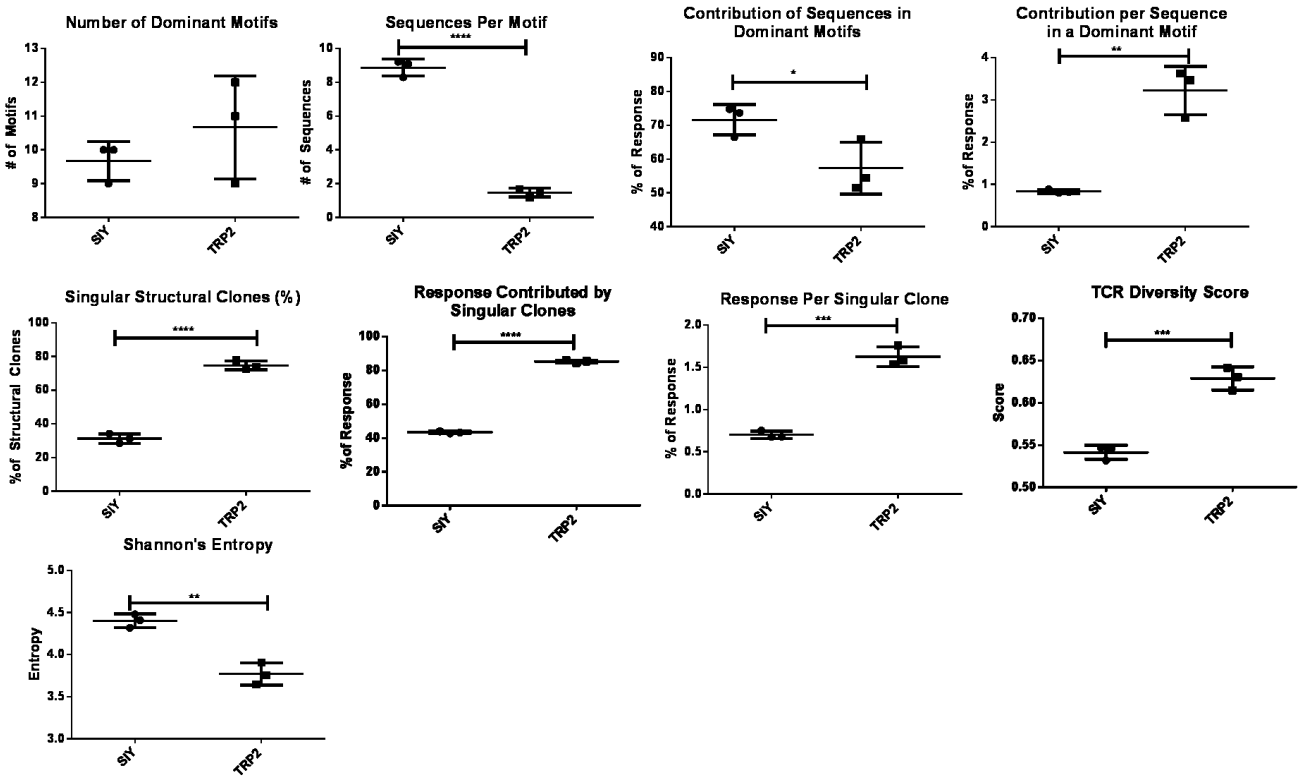


B)

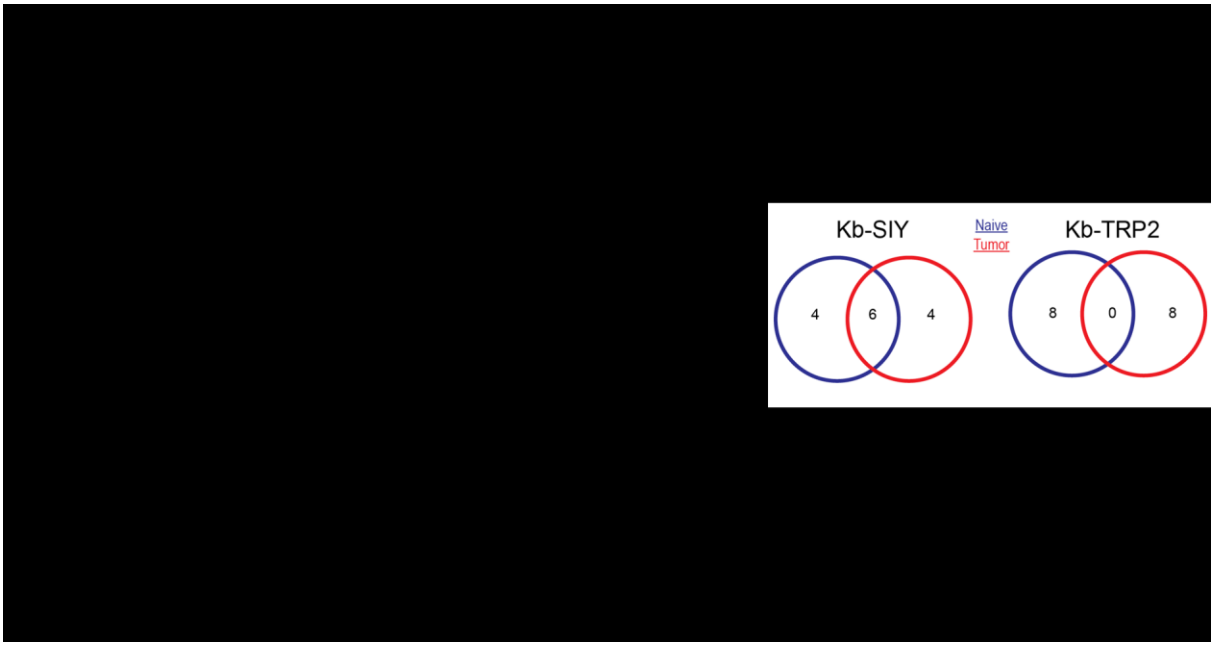
```
for i=1:size(DistanceMatrix3,1);
      for j=1:size(DistanceMatrix3,2);
         if i==j
            if Reads(i)==1
               ScorePreOut(i,j)=0;
            else
               ScorePreOut(i,j)=DistanceMatrix3(i,j)*combntns(Reads(i),2);
            end
         else
            ScorePreOut(i,j)=DistanceMatrix3(i,j)*Reads(i)*Reads(j);
         end
      end
   end

   TCRDiversityScore=1-(sum(sum(ScorePreOut))/combntns(sum(Reads),2));
```

Supplementary Figure 6. Calculation of TCR Diversity Score. A) Initially, all pair-wise distances are calculated with sequence distance based on global alignment scores. Sequence distance is then converted to a mapped sequence distance with values between 0 and 1. This Mapped Sequence Distance is defined as 0 being infinite sequence difference and 1 being identify. B) The distance matrix that is calculated is then weighted by the number of reads. The purpose of this step is to determine the average distance between every cell in the analysis. The piece of code iterates through the distance matrix calculated in A. If i==j (meaning that we are examining the same sequence against itself) and there exists only 1 read of that sequence, the new matrix calculation is 0, meaning there is no need to calculate a distance between a read and itself. If there is more than 1 read, then the new matrix entity calculation is the distance of the sequence and itself multiplied by the total number of all possible combinations of those reads. For example, if there are 10 reads of a given sequence, this new matrix entity is calculated as 1 (sequence distance) * 45 (all possible combinations of 10 reads). In all other cases, where the two sequences are different, the new matrix entity is calculated as the distance between those two sequences multiplied by the number of reads of the first sequence multiplied by the reads of the second sequence. Finally, this new matrix is summed and divided by the number of possible combinations of all reads. This number is then subtracted by 1 to give the final Mapped Sequence Distance where 0 represents identity and 1 represents infinite difference.

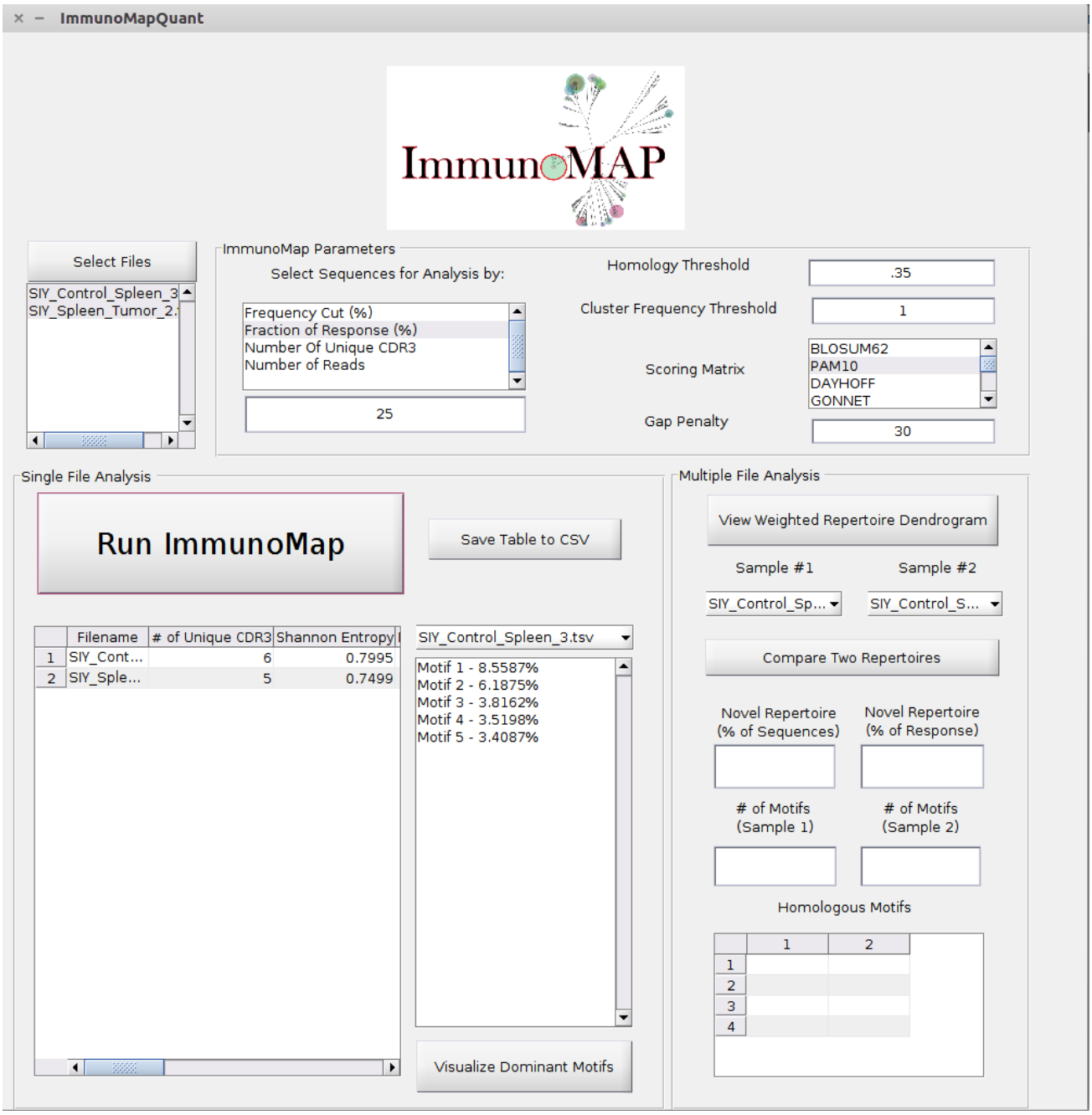
Naïve SIY vs. TRP2 Repertoire



Naïve vs. Tumor-Bearing SIY & TRP2 Repertoire



Supplementary Figure 7. Duplicates of Murine Experiments. Corresponding duplicate figures to Figures 2&3 in main manuscript showing differences in SIY/TRP2 repertoire in naïve and tumor-bearing setting.


ImmunoMapQuant
ImmunoMAP
Select Files
SIY_Control_Spleen_3
SIY_Spleen_Tumor_2
ImmunoMap Parameters
Select Sequences for Analysis by:
Frequency Cut (%)
Fraction of Response (%)
Number Of Unique CDR3
Number of Reads
25
Homology Threshold
.35
Cluster Frequency Threshold
1
Scoring Matrix
BLOSUM62
PAM10
DAYHOFF
GONNET
Gap Penalty
30
Single File Analysis
Run ImmunoMap
Save Table to CSV
Filename
# of Unique CDR3
Shannon Entropy
1 SIY_Cont... 6 0.7995
2 SIY_Sple... 5 0.7499
SIY_Control_Spleen_3.tsv
Motif 1 - 8.5587%
Motif 2 - 6.1875%
Motif 3 - 3.8162%
Motif 4 - 3.5198%
Motif 5 - 3.4087%
Visualize Dominant Motifs
Multiple File Analysis
View Weighted Repertoire Dendrogram
Sample #1
Sample #2
SIY_Control_Sp...
SIY_Control_S...
Compare Two Repertoires
Novel Repertoire (% of Sequences)
Novel Repertoire (% of Response)
# of Motifs (Sample 1)
# of Motifs (Sample 2)
Homologous Motifs
1
2
3
4

Supplementary Figure 8. ImmunoMap Graphical User Interface & Instructions for Use.

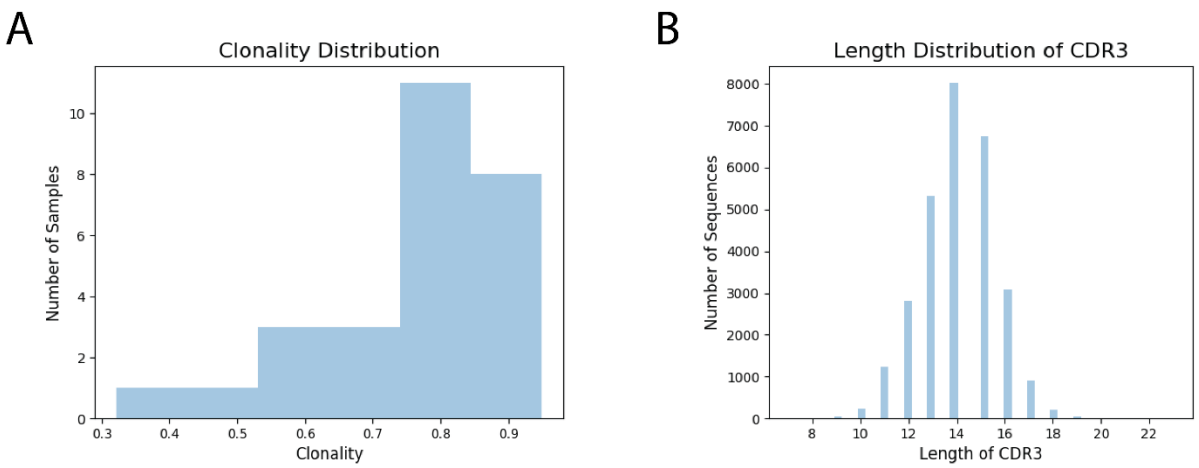
1) User selects files for analysis by pressing the 'Select Files' button and importing tsv files of TCRSeq Data exported by Adaptive Biotechnologies. If one has a different source of TCRSeq, one can submit a tsv file with the first column being nucleotide sequences, the second column being amino acid sequence, and the third column being number of counts.
2) After importing the files, one can set a variety of ImmunoMap parameters including how much of the file to use, structural homology thresholds, cluster frequency thresholds as well as parameters such as the scoring matrix used and the gap penalty.
3) After parameters have been set, one highlights the files under the 'Select Files' button they want to analyze at once. At this point, the user can press 'Run Immunomap' and get a table of all relevant ImmunoMap metrics for their respective files. For each file, they can view the multiple alignments of their dominant motifs by selecting the file and the motif in the window to the right and pressing 'Visualize Dominant Motifs.' Finally, one can save a csv file with all the summarized ImmunoMap metrics by pressing the 'Save Table to CSV' button. This file can be opened by Microsoft Excel using a comma as the delimiter.
4) If one desire to compare multiple files by visualizing them by the Weighted Repertoire Dendrograms, they can highlight the desired files for visualization in the window beneath the 'Select Files' button and press 'View Weighted Repertoire Dendrogram.'
5) Finally, if one desires to compare two repertoires by seeing the percent of structural overlap as well shared dominant motifs, one should select the two files they want to compare in the drop menu's provided and press the 'Compare Two Repertoires' button.

nucleotide	aminoAcid	count (templates/reads)	frequencyCount (%)
TTCTCCCTCATTCTGGAG	CASGTGDNQAPLF	25	0.030194623
GAGAACTTCTCCCTCATT	CASGDPDTQYF	22	0.027358645
CTCACTGTGACATCTGC	CASSIKGQGFDEQYF	15	0.021143412
TTCTCTCTCATTCTGGAG	CASSAQGAGEQYF	12	0.01495893
GAGAACTTCTCCCTCATT	CASGTGDTQYF	11	0.013469187
CTGCTGGAATTGGCTTC	CASSVTGGANTGQLYF	10	0.01300108
TATTTCACTCTGAAAATC	CASSLGQYEQYF	10	0.013209507
TTCCTCCTGCTGGAATTC	CASSDSYNNQAPLF	10	0.013209507
AATCTTCGAATCAAGTC	CASSPPGLGETLYF	10	0.014521574
TTCTCCCTCATTCTGGAG	CASGAGDYAEQFF	9	0.012857572
CTCCTGCTGGAATTGGC	CASSDGGASTGQLYF	9	0.010387196
CTCCTGCTGGAATTGGC	CASSDVGGPYAEQFF	9	0.00955007
GAGATGAACATGAGTGC	CASSPTGGAPEQYF	8	0.011439583
CACTCTGAAGATTCAACCTACAGAACCCAAGGACT		8	0.008689026
CAGCCTAGAAATTCAGT	LCQQSVTGGAEQYF	8	0.012119535
TTCCTCCTGCTGGAATTC	CASSDSYNNQAPLF	8	0.010732297

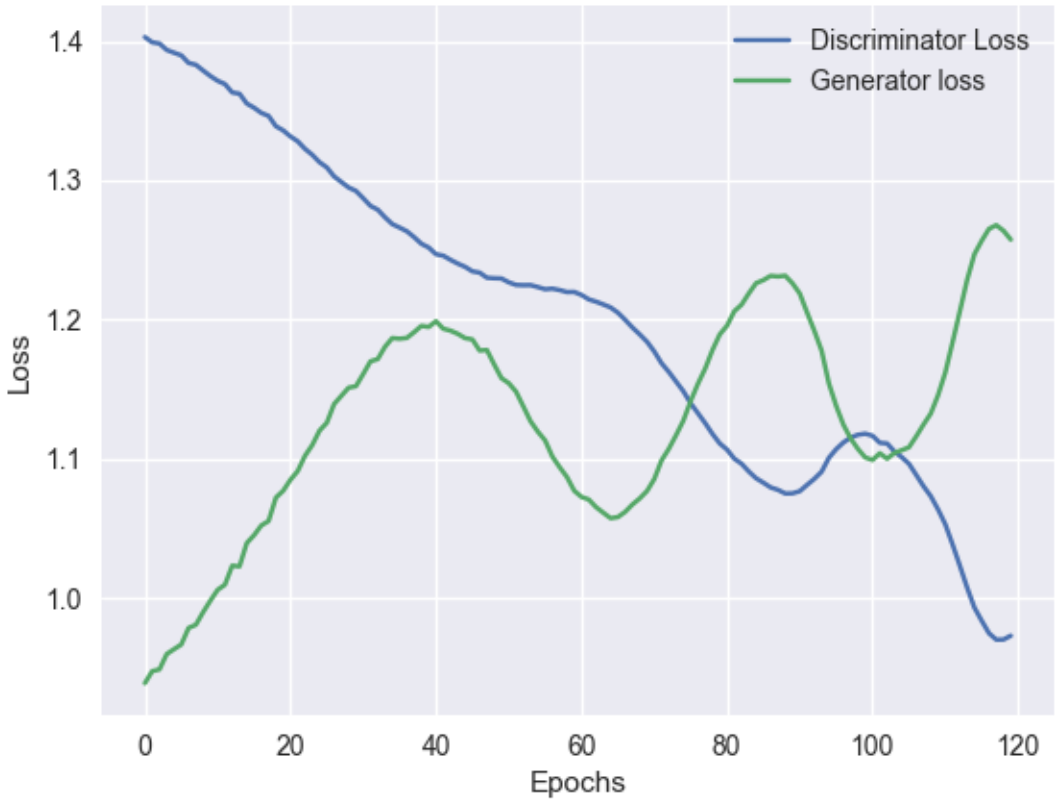
Supplementary Figure 9. TCRSeq File Structure. Following processing by Adaptive Biotechnologies, we received data in this general format. The first column is the nucleotide level sequence of the T-cell clones. The second column is translated from the first column. The counts and frequency columns are used to determine the abundance of a given clone. During processing of the data, we collapse all nucleotide clones that share the same amino acid sequence and sum the counts to determine their relative abundance.

Dataset	Host	Pathology	Description
Glanville_2017	Human	None	T-cells were tetramer sorted for 7 Class I specificities.
Sidhom_2017	Murine	None	CD8 T-cells were stimulated and expanded against SIY and TRP2 antigens before being sorted and sequenced.
Rudqvist_2017	Murine	Cancer	Tumor-infiltrating lymphocytes were collected from mice who either received no treatment, radiation therapy, anti-CTLA4 (9H10), or combo therapy.

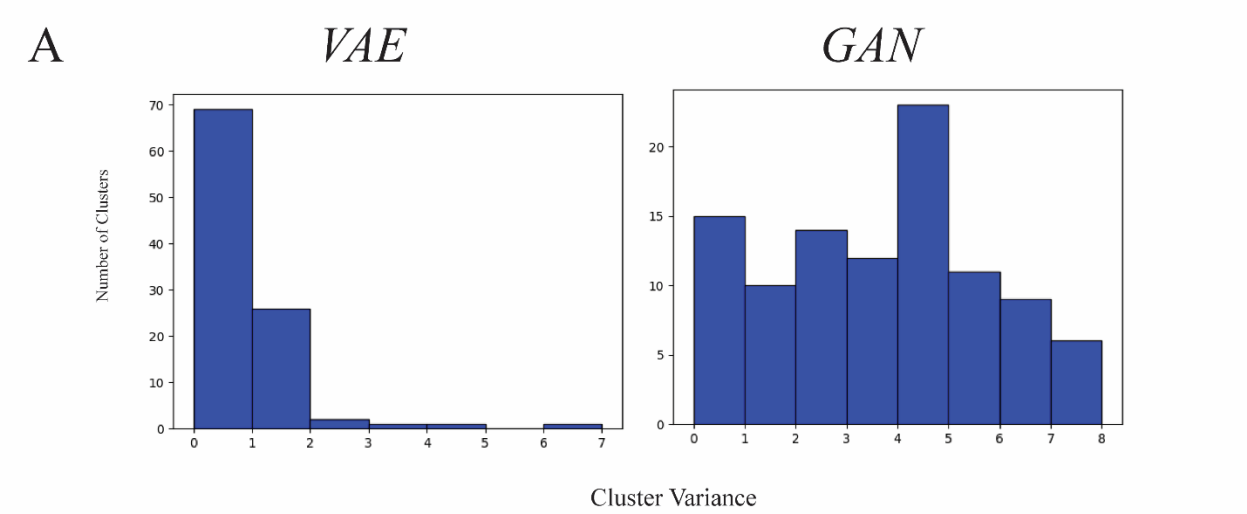
Supplementary Figure 10. Dataset Descriptions. DeepTCR was piloted on three sources of data that covered both human and mouse TCR's including samples taken from normal and cancer pathology.



Supplementary Figure 11. Quantitative TCRSeq Metrics. (a) Distribution of clonalities across all samples used in this manuscript. We note the majority of samples have a high clonality. **(b)** Distribution of length of sequences analyzed in this manuscript.



Supplementary Figure 12. GAN Losses. Prototypical discriminator and generator losses during training of Generative Adversarial Network.



B

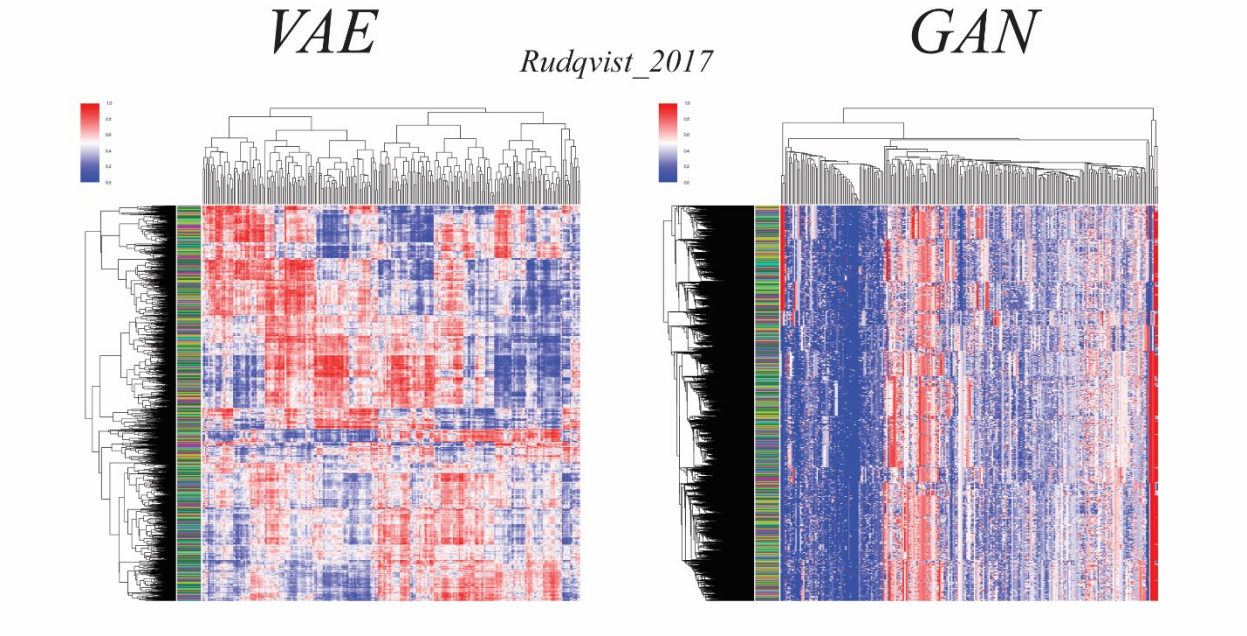
VAE Clusters

CASSDSYNNQAPLF	GNVRILTPNTGQLYF	IRPTWNLPNTGQLYF
CASGDSYNNQAPLF	RYHDCSNANTGQLYF	EFPTFDLANTGQLYF
CASSDAYNNQAPLF	RMKAILEPNTGQLYF	AVRTWNLPNTGQLYF
CASSDSYNNQAPLF	ELRSRHPPNTGQLYF	YDTIGSAPNTGQLYF
CASSDAYSNQAPLF	LCQQFRGEDTGQLYF	RNDQSKLPNTGQLYF
CASSDSYNNQAPLF	TFVTILTPNTGQLYF	RVPTWNLPNTGQLYF
CASSDAYSNQAPLF	DMGAMIWTNTGQLYF	LIGQSKLPNTGQLYF
CASSQDWGSAETLYF	DTSMPNSANTGQLYF	RATNGRCANTGQLYF
CASSDSYNNQAPLF	EGQNYEVANTGQLYF	VSDQSKLPNTGQLYF
CASSDSHNNQAPLF	CASGDGWSNTGQLYF	SGPTFDLANTGQLYF
CASSSDWGSAETLYF	SRQSRHLPNTGQLYF	DCINGRCANTGQLYF
CASSEDRNNQAPLF	VDPNYEVANTGQLYF	ISAYWFCANTGQLYF
	GSTGNMQPNTGQLYF	ARLTWNLPNTGQLYF
	DLRDCSNANTGQLYF	VFPTFDLANTGQLYF
	LNRDCSNANTGQLYF	RGLTFDLANTGQLYF
	QAIIGPVPNTGQLYF	VEPLHSTANTGQLYF
	RGLNYEVANTGQLYF	STALLFCANTGQLYF
	FCSYAHSANTGQLYF	RSLTWNLPNTGQLYF
	SGQNYEVANTGQLYF	WYLHRYSTNTGQLYF
	CAPTWNLPNTGQLYF	RAGYAHSANTGQLYF
	KGALHSTANTGQLYF	PNDQSKLPNTGQLYF
	GTAYLFCANTGQLYF	PGPTWNLPNTGQLYF
	VYPNYEVANTGQLYF	LSALLFCANTGQLYF
	CALTFDLANTGQLYF	RFAYCFCANTGQLYF
	VGRDCSNANTGQLYF	KRLRSAATNTGQLYF
	PDTGNMQPNTGQLYF	
	QFRSRRHPNTGQLYF	
	CLRGYLPTNTGQLYF	
	RSVTILTPNTGQLYF	

GAN Clusters

CASSETGSNTGQLYF	CASSERQGGSSAETLYF	CASSPQDSTGQLYF
CASSDTGSNTGQLYF	CSSRRGGLAETLYF	CASSTGDTGQLYF
CASSTHWASNTGQLYF	CASSIRWGGGAETLYF	CASSRGTGGARSTGQLYF
CASSETGSNTGQLYF	CASSRDRGGETLYF	CASSPGTGGSGQLYF
CASSDMANTGQLYF	CASSDRLAAETLYF	CASSTGDTGQLYF
CASSLSRTGANTGQLYF	CASSERQGGSSAETLYF	CASSLLGGGTGQLYF
CASSVTGGANTGQLYF	CASSRSRDNSAETLYF	CASSEGTGQLYF
CASSQRLGTNTGQLYF	CASGRDRGRAAETLYF	CASSGTGGSTGQLYF
CASSDTGSNTGQLYF	SPIRGLSAETLYF	CASSLGGHTGQLYF
CASGDGWSNTGQLYF	CASGRDRGRAAKTLYF	CASSDGGASTGQLYF
CASSPPGQGNTGQLYF	CASSERQGGSSAETLYF	CASGGTGGASTGQLYF
SASASPNTGQLYF	SPIRELSAETLYF	CASRGTGENTGQLYF
NGPSASPNTGQLYF		CASSDTGQLYF
CASSPPWTGNTGQLYF		CASSGGFTGQLYF
CASSLDRRQANTGQLYF		CASSSSRGVDTGQLYF
CASSVGNTGQLYF		CASSPSSGGDTGQLYF
CASSLRTANTGQLYF		CASSYRGGGGQLYF
CASSTNTGQLYF		CASSYGGGGGQLYF
CASSLRDWGGSNTGQLYF		CASSQEGGTGQLYF
CASSRAANTGQLYF		
CASSPWTGENTGQLYF		
CASSLTGINTGQLYF		
CASSITGVNTGQLYF		

Supplementary Figure 13. VAE vs. GAN Clusters. We used the *Sidhom_2017* dataset to do hierarchical clustering on the latent feature space for both the VAE and the GAN to note differences in the clustering solutions. Since the objective function of the VAE is to reconstruct the sequence, we note the latent space contains knowledge about the length of the sequences and thus, clusters use sequences of similar length. In contrast, since the GAN's objective functions do not rely on reconstruction of the sequences, the clusters use sequences of variable length. **(a)** Hierarchical clustering was performed on SIY-specific sequences, creating 100 flat clusters following dimensionality reduction via VAE or GAN. The distribution of the variance of sequence length in all clusters is shown. **(b)** Prototypical clusters from both unsupervised methods.



Supplementary Figure 14: Rudqvist_2017 Sequence Features. A subset of sequences is shown by their learned unsupervised features from either the variational autoencoder or the generative adversarial network for the Rudqvist_2017 dataset.

Supplementary Table 1. Supertype Classification of Class I Alleles as determined by *Sidney et. al.* [XV]. U = unclassified.

Alleles	Supertype
A0101	A01
A0201	A02
A0202	A02
A0203	A02
A0205	A02
A0206	A02
A0207	A02
A0210	U
A0211	A02
A0212	A02
A0216	A02
A0217	A02
A0219	A02
A0250	A02
A0301	A03
A0302	A03
A0319	U
A1101	A03
A1102	A03
A2301	A24
A2402	A24
A2403	A24
A2501	A01
A2601	A01
A2602	A01
A2603	A01
A2902	A01A24
A3001	A01A03
A3002	A01
A3101	A03
A3201	A01
A3207	A01
A3215	U
A3301	A03
A6601	A03
A6801	A03
A6802	A02
A6823	A03
A6901	A02
A8001	A01
B0702	B07
B0801	B08
B0802	B08
B0803	B08
B1402	B27
B1501	B62
B1502	B62
B1503	B27
B1509	B27
B1517	B58
B1542	B62
B1801	B44
B2701	B27
B2702	B27
B2703	B27
B2705	B27
B2720	B27
B3501	B07
B3503	B07
B3508	B07
B3701	B44
B3801	B27
B3901	B27
B3906	B27
B4001	B44
B4002	B44
B4013	U
B4201	B07
B4402	B44
B4403	B44
B4501	B44
B4506	U
B4601	B62
B4801	B27
B5101	B07
B5301	B07
B5401	B07
B5701	B58
B5801	B58
B5802	B58
B7301	B27
B8301	U
C0401	U
C0602	U
C1402	U
E0101	U

Supplementary Table 2. All Class I alleles trained with Counts of Peptide/Allele

Alleles	Counts
A0101	4609
A0201	12324
A0202	4077
A0203	6244
A0205	88
A0206	5561
A0207	69
A0210	18
A0211	1084
A0212	1183
A0216	919
A0217	332
A0219	1244
A0250	135
A0301	7195
A0302	10
A0319	29
A1101	6248
A1102	14
A2301	2416
A2402	3191
A2403	1227
A2501	960
A2601	4307
A2602	631
A2603	522
A2902	2548
A3001	2717
A3002	1847
A3101	5621
A3201	1089
A3207	87
A3215	74
A3301	3510
A6601	52
A6801	3708
A6802	5499
A6823	80
A6901	2565
A8001	1164
B0702	4513
B0801	3298
B0802	1000
B0803	455
B1402	80
B1501	4178
B1502	165
B1503	594
B1509	816
B1517	1446
B1542	357
B1801	2594
B2701	2
B2702	4
B2703	875
B2705	3433
B2720	87
B3501	3198
B3503	13
B3508	1
B3701	30
B3801	492
B3901	1623
B3906	33
B4001	3199
B4002	964
B4013	56
B4201	8
B4402	2117
B4403	1382
B4501	953
B4506	355
B4601	1806
B4801	882
B5101	2718
B5301	1616
B5401	1110
B5701	2781
B5801	3118
B5802	42
B7301	121
B8301	333
C0401	352
C0602	54
C1402	87
E0101	1

Supplementary Table 3. Detailed assessment IEDB Class I ic50 benchmarks. Shown are AI-MHC along with 11 provided algorithms on the website. AUC values are reported for a given benchmark (identified by IEDB reference, Allele Name, Peptide Length, Peptide Count, and Measurement Type). Bold indicates the algorithm that performed the best on a given benchmark.

IEDB reference	Allele name	Peptide length	Peptide count	Measurement type	TrainingCount	InternalAUC	AI-MHC	ARB	IEDBConsensus	NetMHC 3.4(ANN)	NetMHC 4.0(ANN)	NetMHCcons	NetMHCpan 2.8	NetMHCpan 3.0	PickPocket	SMM	SMMPMBEC	mhcflurry	Best score
1024516	A0201	10	31	ic50	12154	0.956	**0.738**	0.679	0.667	0.667		0.667	0.679		0.655	0.679	0.655		0.738
1024516	A0201	9	47	ic50	12154	0.956	0.600	**0.656**	0.633	0.633		0.622	0.622		0.644	0.644	0.633		0.656
1024516	A0202	9	50	ic50	4022	0.966	0.716	**0.754**	0.714	0.741		0.742	0.705		0.691	0.676	0.723		0.754
1024516	A0203	10	32	ic50	6152	0.951	**0.880**	0.731	0.743	0.794		0.806	0.760		0.714	0.720	0.703		0.880
1024516	A0203	9	50	ic50	6152	0.951	0.607	0.693	0.643	0.644		0.673	0.686		**0.753**	0.649	0.655		0.753
1024516	A0206	10	29	ic50	5469	0.935	0.731	0.752	0.777	0.784		**0.801**	0.796		0.724	0.761	0.754		0.801
1024516	A0206	9	24	ic50	5469	0.935	0.863	0.879		0.853		0.863	0.874		0.779	0.868	**0.884**		0.884
1024516	A6802	10	24	ic50	5407	0.957	**0.824**	0.759	0.792	0.796		0.790	0.694		0.769	0.796	0.796		0.824
1024516	A6802	9	45	ic50	5407	0.957	0.822	0.791	0.834	0.858		0.858	0.847		**0.868**	0.831	0.829		0.868
1024516	A0201	9	24	ic50	12154	0.956	**0.764**	0.693		0.593			0.671			0.636			0.764
1024840	A2402	9	20	ic50	3190	0.863	0.542	0.500		0.635			0.667			**0.771**			0.771
1024840	A3002	9	56	ic50	1847	0.830	0.498	**0.657**		0.601			0.483			0.569			0.657
1024840	A6801	9	35	ic50	3708	0.898	0.808	0.771					**0.843**			0.794			0.843
1024840	B5801	9	35	ic50	3117	0.993	0.667	0.546		0.650			**0.716**			0.665			0.716
1026041	A0201	9	10	ic50	12154	0.956	0.833	0.917		**1.000**			0.917			0.958			1.000
1028228	C0303	9	163	ic50					**0.950**	0.935		0.931	0.905		0.904	0.946	0.936		0.950
1028229	C0401	9	153	ic50			0.449		0.993	**1.000**		0.998	0.904		0.852	0.972	0.996		1.000
1028230	C0501	9	172	ic50					0.996	**0.999**		0.997	0.968		0.946	0.996	0.990		0.999
1028231	C0602	9	183	ic50			0.455		**0.983**	0.980		0.971	0.947		0.930	0.982	0.976		0.983
1028232	C0702	9	140	ic50					0.963	0.969		**0.972**	0.951		0.908	0.961	0.965		0.973
1028233	C0802	9	87	ic50					0.929	**0.963**		0.957	0.905		0.831	0.915	0.924		0.963
1028234	C1402	9	218	ic50			0.621		0.884	**0.926**		0.915	0.865		0.829	0.874	0.879		0.926
1028235	C1502	9	181	ic50					0.948	**0.967**		0.965	0.918		0.867	0.939	0.941		0.967
1028286	C1203	9	172	ic50					0.586	0.687		**0.711**	0.706		0.647	0.574	0.610		0.711
1028553	A0201	9	22	ic50	12154	0.956	0.850	0.900	0.837	0.913		**0.925**	**0.925**		0.825	0.850	0.825		0.925
1028553	B0702	9	22	ic50	4426	0.980	0.877	0.877	**0.921**	**0.921**		0.912	0.895		0.895	0.895	0.877		0.921
1028554	A0201	9	44	ic50	12154	0.956	0.761	0.761	0.842	0.824		0.888	0.888		0.880	**0.890**	0.861		0.890
1028554	B0702	9	52	ic50	4426	0.980	0.812	0.757	0.880	**0.882**		0.862	0.772		0.743	0.853	0.855		0.882
1028554	B3501	9	56	ic50	3142	0.960	0.522	0.642	0.651	0.566		0.642	**0.679**		0.503	0.591	0.528		0.679
1028554	B4403	9	46	ic50	1336	0.930	0.667	0.558	0.640	0.651		0.643	0.612		0.643	0.752	**0.760**		0.760
1028554	B5701	9	53	ic50	2703	0.957	0.844	0.628	0.871	**0.944**		0.916	0.863		0.849	0.765	0.772		0.944
1028790	A0201	9	55	ic50	12154	0.956	0.598	0.566		0.565		0.566	0.574		**0.660**	0.564	0.553		0.660
1028790	A0201	10	35	ic50	12154	0.956	**0.798**	0.637	0.661	0.669		0.677	0.677		0.645	0.653	0.593		0.798
1028790	A0202	9	55	ic50	4022	0.966	0.693	0.733	0.734	0.744		**0.747**	0.713		0.721	0.714	0.742		0.747
1028790	A0203	9	55	ic50	6152	0.951	0.678	0.684	0.675	0.668		0.694	0.696		**0.706**	0.678	0.686		0.766
1028790	A0203	10	35	ic50	6152	0.951	**0.888**	0.684	0.724	0.781		0.791	0.750		0.704	0.730	0.699		0.888
1028790	A0206	9	55	ic50	5469	0.935	0.737	0.725	0.752	0.757		**0.773**	0.770		0.749	0.732	0.732		0.773
1028790	A0206	10	35	ic50	5469	0.935	**0.833**	0.764	0.783	0.754		0.761	0.768		0.656	0.790	0.801		0.833
1028790	A6802	10	35	ic50	5407	0.957	**0.727**	0.697	0.690	0.699		0.655	0.620		0.662	0.718	0.685		0.727
1028790	A6802	9	55	ic50	5407	0.957	0.792	0.761	0.813	**0.835**		0.825	0.806		0.797	0.811	0.801		0.835
1029061	B5701	9	26	ic50	2703	0.957	0.886	0.568	0.875	0.875		0.886	0.943		0.909	0.909	0.886	**0.966**	0.966
1029957	B3801	9	28	ic50	465	1.000	0.674	0.647	0.997	**1.000**		**1.000**	0.963		0.957	0.984	0.989	**1.000**	1.000
1029957	B3906	9	32	ic50			0.294					**0.814**	**0.814**		0.749				0.814
1031253	A0301	9	14	ic50	7181	0.966	0.733		0.911			0.911		0.933	**0.956**	0.867	0.911	0.756	0.956
1031253	B0702	9	13	ic50	4426	0.980	**1.000**		**1.000**			**1.000**		**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
1031253	B2705	9	12	ic50	3421	0.986	0.629		0.629		**0.986**	0.657		0.629	0.600	0.629	0.600	0.571	0.686

Supplementary Table 4. All Class II alleles trained with Counts of Peptide/Allele

Alleles	Counts
DPA10103-DPB10201	787
DPA10103-DPB10301	1563
DPA10103-DPB10401	2725
DPA10103-DPB10402	45
DPA10103-DPB10601	584
DPA10201-DPB10101	2447
DPA10201-DPB10501	2470
DPA10201-DPB11401	2302
DPA10301-DPB10402	2641
DQA10101-DQB10501	2946
DQA10102-DQB10501	833
DQA10102-DQB10502	800
DQA10102-DQB10602	2747
DQA10102-DQB10604	61
DQA10103-DQB10302	6
DQA10103-DQB10603	462
DQA10104-DQB10503	883
DQA10201-DQB10201	23
DQA10201-DQB10202	944
DQA10201-DQB10301	827
DQA10201-DQB10303	761
DQA10201-DQB10402	768
DQA10301-DQB10201	4
DQA10301-DQB10301	207
DQA10301-DQB10302	3111
DQA10302-DQB10303	6
DQA10302-DQB10401	27
DQA10303-DQB10402	567
DQA10401-DQB10402	2890
DQA10501-DQB10201	2897
DQA10501-DQB10301	3585
DQA10501-DQB10302	847
DQA10501-DQB10303	564
DQA10501-DQB10402	749
DQA10505-DQB10301	1
DQA10601-DQB10402	565
DRB10101	10412
DRB10102	8
DRB10103	42
DRB10301	5352
DRB10302	37
DRB10401	6317
DRB10402	53
DRB10403	59
DRB10404	3657
DRB10405	3962
DRB10406	14
DRB10411	2
DRB10701	6325
DRB10801	937
DRB10802	4465
DRB10803	8
DRB10804	3
DRB10901	4318
DRB11001	2066
DRB11101	6045
DRB11104	27
DRB11201	2384
DRB11301	1034
DRB11302	4477
DRB11402	1
DRB11501	4850
DRB11502	23
DRB11503	1
DRB11602	1699
DRB30101	4633
DRB30202	3334
DRB30301	884
DRB40101	3961
DRB40103	846
DRB50101	5125
DRB50102	2
H-2-IAb	1794
H-2-IAd	774
H-2-IAk	115
H-2-IAq	31
H-2-IAs	190
H-2-IAu	56
H-2-IEd	245
H-2-IEk	68

Supplementary Table 5. Detailed assessment IEDB Class II ic50 benchmarks. Shown are AI-MHC along with 11 provided algorithms on the website. AUC values are reported for a given benchmark (identified by IEDB reference, Allele Name, Peptide Length, Peptide Count, and Measurement Type). Bold indicates the algorithm that performed the best on a given benchmark.

IEDB reference	Allele name	Peptide length	Peptide count	Measurement type	Training Count	Internal AUC	AI-MHC	Comblib matrices	Consensus IEDB method	NN-align	NetMHCIIpan-3.1	SMM-align	Tepitope (Sturniolo)	Best score
1027578	DRB10301	0	14	ic50	4355	0.894	0.950		0.950	0.975	**1.000**	0.900	0.075	1.000
1027578	DRB10701	0	19	ic50	5262	0.913	0.762	0.964	**1.000**	0.952	0.929	0.976	0.101	1.000
1027578	DRB30101	0	20	ic50	3790	0.919	**0.956**	0.703	0.846	0.901	0.945	0.879		0.956
1027578	DRB40101	0	14	ic50	3954	0.887	0.600	0.725	0.650	0.600	**0.800**	0.725		0.800
1027578	DRB40103	0	18	ic50			**0.754**				**0.800**			0.754
1028057	DRB10101	0	29	ic50	10257	0.848	0.838	0.851	0.851	0.851	**0.890**	0.753	0.260	0.890
1028057	DRB10401	0	29	ic50	6146	0.796	0.700		0.760	0.870	**0.920**	0.770	0.475	0.920
1028057	DRB10701	0	29	ic50	5262	0.913	0.796	0.778	**0.963**	0.907	0.889	0.796	0.278	0.963
1028057	DRB11301	0	29	ic50	171	1.000	0.808		**0.842**	0.744	0.483		0.158	0.842
1028057	DRB11501	0	29	ic50	4684	0.877	**0.808**				0.679	0.705	0.301	0.808
1028237	DQA10103-DQB10603	0	339	ic50	462	0.864	**0.964**			0.692	0.806			0.964
1028238	DQA10201-DQB10303	0	759	ic50	761	0.857	**0.943**				0.760			0.943
1028239	DQA10501-DQB10302	0	834	ic50	847	0.828	**0.896**				0.771			0.896
1028241	DRB10101	0	885	ic50	10257	0.848	0.847	0.789	0.864	0.876	**0.890**	0.849	0.185	0.890
1028242	DRB10301	0	863	ic50	4355	0.894	0.766		0.789	0.777	**0.856**	0.777	0.251	0.856
1028243	DRB10404	0	861	ic50	2796	0.893	0.828		0.802	0.797	**0.861**	0.783	0.173	0.861
1028244	DRB10701	0	857	ic50	5262	0.913	0.864	0.784	0.867	0.869	**0.876**	0.862	0.188	0.876
1028245	DRB10801	0	889	ic50			0.757		0.798		**0.863**		0.201	0.863
1028246	DRB10901	0	855	ic50	3445	0.927	0.851	0.600	0.820	0.857	**0.880**	0.803		0.880
1028247	DRB11101	0	868	ic50	5039	0.868	0.819		0.841	0.859	**0.880**	0.839	0.208	0.880
1028248	DRB11301	0	837	ic50	171	1.000	**0.830**		0.786		0.772		0.214	0.830
1028249	DRB11454	0	854	ic50							**0.889**			0.889
1028250	DRB30101	0	832	ic50	3790	0.919	0.770	0.677	0.799	0.825	**0.835**	0.806		0.835
1028251	DRB30202	0	771	ic50	2563	0.928	**0.745**				0.740			0.745
1028252	DRB30301	0	854	ic50			0.672				**0.781**			0.781
1028253	DRB40103	0	821	ic50			0.777				**0.788**			0.788
1028254	DRB50101	0	762	ic50	4363	0.909	0.778		0.775	0.806	**0.843**	0.778	0.260	0.843
1028711	DQA10102-DQB10501	0	825	ic50	833	0.759	**0.926**				0.596			0.936
1028712	DQA10601-DQB10402	0	565	ic50	565	0.964	**0.963**				0.497			0.963
1028713	DQA10201-DQB10402	0	765	ic50	768	0.816	**0.962**				0.519			0.962
1028714	DQA10501-DQB10402	0	747	ic50	749	0.763	**0.937**				0.577			0.937
1028715	DQA10303-DQB10402	0	567	ic50	567	0.944	**0.921**				0.482			0.921
1028716	DQA10501-DQB10303	0	564	ic50	564	0.866	**0.919**				0.812			0.919
1028717	DQA10201-DQB10301	0	818	ic50	827	0.815	**0.950**				0.814			0.950
1030032	DRB10101	0	108	ic50	10257	0.848	0.603	0.416	0.577	**0.818**	0.593	0.593	0.532	0.818
1030032	DRB10301	0	124	ic50	4355	0.894	0.679		0.639	0.697	**0.744**	0.613	0.435	0.744
1030032	DRB10401	0	132	ic50	6146	0.796	0.722		0.603	0.803	**0.830**	0.589	0.557	0.830
1030032	DRB10701	0	148	ic50	5262	0.913	0.756	0.433	0.549	0.729	0.778	0.632	**0.843**	0.843
1030032	DRB10802	0	142	ic50	4323	0.859	0.703		0.315	0.705	**0.736**	0.385	0.721	0.736
1030032	DRB11101	0	121	ic50	5039	0.868	0.899		0.292	0.931	**0.945**	0.866	0.847	0.945
1030032	DRB11302	0	134	ic50	4343	0.934	0.701		0.765	**0.910**	0.903	0.756	0.720	0.910
1030032	DRB11501	0	120	ic50	4684	0.877	0.586		0.384	0.757	**0.770**	0.470	0.750	0.770
1030414	DQA10103-DQB10603	0	18	ic50	462	0.864	**1.000**				0.875			1.000
1030414	DQA10301-DQB10302	0	18	ic50	3111	0.900	0.911	0.830	**0.991**	0.964	0.973	0.946		0.991
1030414	DRB10101	0	18	ic50	10257	0.848	**0.923**	0.377	0.738	0.862	0.785	0.738	0.285	0.923
1030414	DRB10401	0	18	ic50	6146	0.796	0.861		0.681	0.722	**0.889**	0.625	0.458	0.889
1030414	DRB10701	0	18	ic50	5262	0.913	0.775	0.725	**0.931**	0.825	0.825	0.838	0.238	0.931
1030414	DRB10901	0	18	ic50	3445	0.927	0.857	0.339	**1.000**	**1.000**	0.946	0.911		1.000
1030414	DRB11101	0	17	ic50	5039	0.868	**1.000**		0.869	0.929	**1.000**	**1.000**	0.238	1.000
1030414	DRB11501	0	18	ic50	4684	0.877	**0.929**		0.714	0.821	0.804	0.732	0.455	0.929
1030414	DRB11502	0	18	ic50			0.511		0.544		**0.667**		0.444	0.667
1030865	DRB10101	0	14	ic50	10257	0.848	0.778	0.556	0.822	0.822	**0.911**	0.500	0.478	0.911
1031250	DQA10102-DQB10602	0	10	ic50	2747	0.921	0.625	0.250	0.750	0.813	0.813	**1.000**		1.000
1032311	DRB10101	0	16	ic50	10257	0.848	0.893	0.589	0.964	0.964	**1.000**	**1.000**	0.000	1.000

CURRICULUM VITAE

John-William Sidhom

2809 Boston Street, Apt. 328

Baltimore, MD. 21224

Jsidhom1@jhmi.edu

Date and Place of Birth

August 17th, 1989. Providence, Rhode Island.

Education

Present Enrollment

MD/PhD Candidate at the Johns Hopkins University School of Medicine, Baltimore, Maryland

Earned Degrees/Certificates

Masters of Bioengineering, Innovation, and Design at the Johns Hopkins University School of Medicine, Baltimore, Maryland

Bachelors of Biomedical Engineering at the University of Michigan, Ann Arbor, Michigan

- Minor in Mathematics, Certificate of Entrepreneurship

Awards & Fellowships

- Paul & Daisy Soros Fellowship Finalist
- Medtronic Fellow – Graduate Fellowship

Select Publications

- Riaz, N., Havel, J. J., Makarov, V., Desrichard, A., Urba, W. J., Sims, J. S., Hodi, F.S., Martin-Algarra. S., Mandal, R., Sharfman, W.H., Bhatia, S., Hwu, W.J., Gajewski, T.F., Slingluff Jr, C.L., Chowell, D., Kendall, S.M., Chang, H., Shah, R., Kuo, F., Morris, L.G.T., **Sidhom, J.W**., Schneck, J.P., Horak, C.E., Weinhold, N., Chan, T.A. (2017). Tumor and Microenvironment Evolution during Immunotherapy with Nivolumab. *Cell*.
- Rudqvist, N.P., Pilones, K.A., Lhuillier, C., Wennerberg, E., **Sidhom, J.W**., Emerson, R.O., Robins, H.S., Schneck, J.P., Formenti, S.C., Demaria, S. (2017). Radiotherapy and CTLA-4 blockade shape the TCR repertoire of tumor-infiltrating T cells. *Cancer Immunology Research*
- **Sidhom, J.W.,** Bessell, C.A., Havel, J.J., Kosmides, A., Chan, T.A., Schneck, J.P. (2017). ImmunoMap: A Novel Bioinformatics Tool for T-Cell Repertoire Analysis. *Cancer Immunology Research*