(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 February 2025 (13.02.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/034619 A1

(51) International Patent Classification:
C07K 14/705 (2006.01) *G16B 30/20* (2019.01)
G06N 3/02 (2006.01) *G16B 15/30* (2019.01)
G16B 30/10 (2019.01) *G01N 33/68* (2006.01)

(21) International Application Number:
PCT/US2024/040882

(22) International Filing Date:
03 August 2024 (03.08.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/517,862 04 August 2023 (04.08.2023) US

(71) Applicant: **VCREATE, INC.** [US/US]; 930 Brittan Avenue, San Carlos, CA 94070 (US).

(72) Inventors: **CHEN, Binbin**; 1546 San Antonio Street, Menlo Park, CA 94025 (US). **FAST, Ethan**; 1546 San Antonio Street, Menlo Park, CA 94025 (US). **DHAR, Manjima**; 1 Lady Diana Circle, Marlton, NJ 08053 (US).

(74) Agent: **KUMAMOTO, Andrew**; HelixIP LLP, 1935 Belmont Avenue, San Carlos, CA 94070 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE,

(54) Title: METHODS FOR FINDING NOVEL T-CELL RECEPTORS

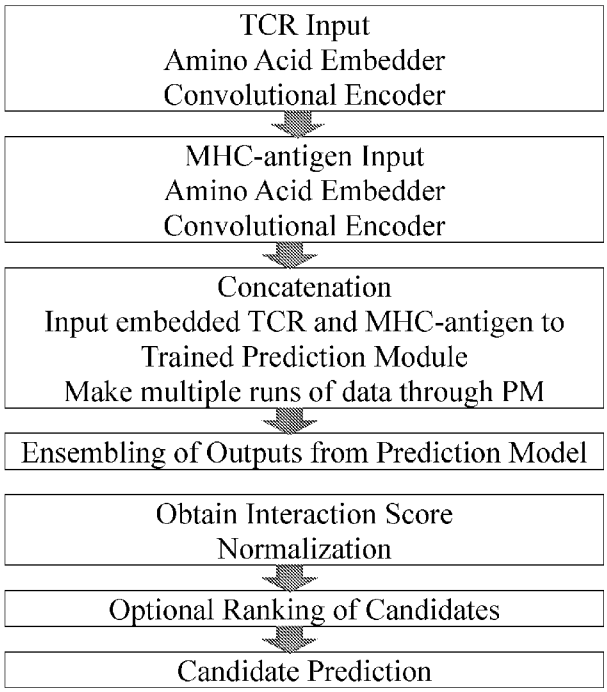

FIG. 1

(57) Abstract: The present disclosure provides methods, systems, and computer readable storage media for the identifying T-cell receptor and pMHC binding pairs. The methods, systems, and computer readable storage media disclosed herein are especially suited to finding TCR-pMHC binding pairs when the TCR and/or the pMHC present data that is not similar to the training data.

WO 2025/034619 A1

[Continued on next page]

SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*
— *with sequence listing part of description (Rule 5.2(a))*

METHODS FOR FINDING NOVEL T-CELL RECEPTORS

REFERENCE TO SEQUENCE LISTING, TABLE OR COMPUTER PROGRAM

[0001] The official copy of the Sequence Listing is submitted concurrently with the specification as an xml file, made with WIPO Sequence Version 2.1.0, via EFS-Web, with a file name of "VC007.xml", a creation date of August 4, 2023, and a size of 2 kilobytes. The Sequence Listing filed via EFS-Web is part of the specification and is incorporated in its entirety by reference herein.

BACKGROUND

[0002] A fundamental and unsolved question about T-cell receptors regards T-cell receptor (TCR) binding specificity for tumor associated antigens (TAA) presented by a certain classes of MHC proteins (pMHCs). The ability to link pMHCs to TCR sequences is essential for monitoring the interactions between the immune system and tumors. Additional insights into the interactions between pMHCs and TCR sequences could be used to enhance the design or implementation of various types of immunotherapies. For example, the selection of candidates for synthesizing TAA vaccines could be informed by whether there are any existing pairing detected between the antigen candidates and the patient's TCR repertoire.

[0003] Existing approaches to detecting T-cell receptor and pMHC pairs (e.g., tetramer analysis, TetTCR-seq, and T-scan) are time-consuming, technically challenging, and too costly to be clinically viable. Additionally, these approaches are experimental and have not been rigorously validated or even validated at all in clinical settings. Therefore, there exists a need for developing machine learning approaches to predict T-cell receptor binding specificity of target antigens. Data driven approaches to identifying T-cell receptor and pMHC pairs would significantly reduce the time and cost of identifying the pairings and could complement experimental approaches by streamlining the validation of existing techniques and facilitating the development of improved experimental approaches.

SUMMARY

[0004] Disclosed herein are methods, systems and computer-readable storage media for predicting T cell receptor (TCR) binding specificities. The methods, systems and computer-readable storage media can make a set of TCR embeddings, a set of MHC embeddings that encode an antigen and a major histocompatibility complex (MHC), use a trained predictive model (e.g., a convolutional model) multiple times to generate output from versions of the predictive model that are different, an ensemble combines these results and optionally performs post ensemble activities (e.g., normalizes the output), and an interaction score is obtained.

Optionally, the interaction score can be modified (e.g., normalized) and can be used to rank the candidate TCRs, and/or rank the pMHCs.

[0005] The methods, systems and computer-readable storage media disclosed herein has separate embedders for the TCR and the pMHC (polypeptide-MHC). The data input to these embedders is "abbreviated" sequence information that identifies the sequences important for the TCR-pMHC interaction. Sequences that have little to no role in this interaction are not included. When this TCR and pMHC data is made into a training data set, the methods, systems and computer-readable storage media disclosed herein can include a data augmentation step where each positive binding pair has certain sequence information (e.g., the J chain) removed and the "incomplete" data pair is placed into the training data set. This introduces uncertainty into the training data and produces a trained prediction model that can make better predictions with data that is different from the training data and so may have unknowns.

[0006] The disclosed methods, systems and computer-readable storage media may further produce candidate TCR-pMHC pairs that can be experimentally validated. The validation of the biological activity of the candidate TCR-pMHC pairs can also validate the prediction model by comparing the binding specificity prediction for the input TCR-pMHC pair to the experimentally determined activity of the candidate TCR-pMHC pair. The disclosed methods may further comprise determining a clonal expansion of a plurality of T cells, the clonal expansion including multiple TCR clones having known binding interactions with a set of pMHCs and a clone size for each of the multiple TCR clones; determining a prediction for binding specificity between each of the multiple TCR clones and each of the pMHCs included in the set of pMHCs based on the prediction model; and validating the prediction model by comparing the clone size for each of the TCR clones to the predicted binding specificity.

[0007] The methods, systems and computer-readable storage media may train a TCR embedding layer on a TCR training dataset including multiple training TCR protein sequences, the TCR training dataset including a structured data representation of one or more biochemical properties of multiple amino acids included in the training TCR protein sequences; and determining the vector representation of the one or more TCR protein sequences based on the TCR embedding layer.

[0008] Similarly, the methods, systems and computer-readable storage media may train a pMHC embedding layer on a pMHC training dataset including multiple training pMHC sequences, the pMHC training dataset including a structured data representation of one or more biochemical properties of multiple amino acids included in the training pMHC sequences; and

determining the vector representation of the one or more pMHC sequences based on the pMHC embedding layer.

[0009] The methods, systems and computer-readable storage media may further normalize the MHC embeddings and/or the TCR embeddings to enable the prediction model to be pre-trained on multiple classes of pMHCs. In various aspects, the prediction for binding specificity includes a variable that describes a percentile rank of a predicted binding strength between the input TCR-pMHC pair, with respect to a pool of randomly sampled TCRs (as a background distribution) against the pMHC included in the TCR-pMHC pair.

[0010] Systems disclosed herein for predicting T cell receptor (TCR) binding specificities may have a memory including executable instructions; and a processor that may be configured to execute the executable instructions and cause the system to: determine a set of TCR embeddings; determine a set of pMHC embeddings; pre-train a prediction model on the set of TCR embeddings and the set of pMHC embeddings; train the prediction model using a differential learning schema that feeds a binding TCR-pMHC pair and a non-binding TCR-pMHC pair into the prediction model; and determine a prediction for binding specificity of an input TCR-pMHC pair based on the prediction model.

BRIEF DESCRIPTION OF THE FIGURES

[0011] FIG. 1 illustrates an exemplary process for predicting TCR and pMHC interactions.

[0012] FIG. 2 is a flow chart showing the TAPIR method for predicting TCR and pMHC interactions.

[0013] FIG. 3 is a systems overview for the TAPIR method.

[0014] FIG. 4 illustrates making an ensemble of TAPIR models and combining the ensemble into an interaction score.

[0015] FIG. 5 is flow chart showing the training of a TAPIR model.

[0016] FIG. 6 illustrates the data augmentation step of the TAPIR model.

[0017] FIG. 7 shows a flow char including validation of the predicted TCR candidates.

[0018] FIG. 8 shows a detailed flow chart for a convolutional neural network.

DETAILED DESCRIPTION

[0019] Before the various embodiments are described, it is to be understood that the teachings of this disclosure are not limited to the particular embodiments described, and as such can, of course, vary. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to be limiting, since the scope of the present teachings will be limited only by the appended claims.

[0020] Aspects of the present disclosure provide systems, methods, and computer-readable storage media for automated screening of TCRs against MHC presented antigens using machine learning for use in pharmaceutical products such as drugs, medicine, remedies, cosmetics, and the like. The techniques described herein support automatic determination of uses (e.g., particular disease states or conditions for which the candidate pharmaceutical molecules or compounds may provide a therapeutic effect) for newly identified TCRs, and new uses for existing TCRs (e.g., previously identified TCRs), using artificial intelligence and machine learning techniques. As used in the present disclosure, new or newly-identified TCRs encompass TCRs that may have not previously been identified, tested, and/or studied for a given pharmaceutical application (e.g., identification of new molecules for existing or new uses, disease states, or conditions and/or identification of existing molecules for new uses, disease states, or conditions). The artificial intelligence and machine learning techniques described herein may be trained using a variety of binding data associated with known pharmaceutical targets (e.g., tumor associated antigens), such as data that indicates chemical properties of multiple different pharmaceutical targets, for example physiochemical structures of the pharmaceutical targets, molecular shapes of the pharmaceutical targets, molecular fingerprints of the pharmaceutical targets, and the like. The pharmaceutical targets may include proteins (e.g., protein receptors, enzymes, ion channels, etc.), nucleic acids, and the like. The binding data used for training may be obtained from a variety of sources, such as publicly available binding information from databases such as VDJDB (a public database of about 60,000 TCRs with their associated antigen targets), third-party databases (e.g., pharmaceutical company databases, university databases, government agency databases, and the like), proprietary databases, or a combination thereof. An embedder is trained to represent the polypeptide of the TCR and/or the polypeptides of the MHC-antigen as multidimensional vectors (e.g., 32-64 dimensions). This can generate training data which can be used to train machine learning model(s) to assign TCRs and MHC-antigen molecules to various clusters based on chemical properties. Such clustering may reduce the search space for identifying MHC-antigens that will (or are likely to) bind to the candidate TCRs, thereby improving efficiency and reducing resource usage and costs associated with the screening process. Using artificial intelligence and machine learning that are trained based on binding data associated with large quantities of MHC-antigens may result in identification of pharmaceutical uses for TCRs that would not be identified by a human (e.g., a chemist or biochemist) using existing screening processes. To illustrate, because the artificial intelligence and machine learning are able to determine underlying similarities between a wider variety of MHC-antigens and candidate TCRs, many of which may not be apparent to a human,

the systems and methods described herein may enable identification of previously unknown uses for existing TCRs and/or uses for newly-identified and unstudied TCRs. Additionally, automated identification of the pharmaceutical uses may be faster than identification performed by other systems that require substantial user interaction and decision making. By providing improved insight into pharmaceutical uses of new and existing TCRs in a shorter period of time, the systems and methods described herein may substantially reduce the costs and shorten the development cycle associated with discovering and launching new drugs (e.g., pharmaceuticals) or identifying new uses for existing drugs. Although described in the context of TCRs, the techniques of the present disclosure may be applied to identify uses for other types of products, such as health products and supplements, personal hygiene products, cosmetic products, biotech products, chemical products, and the like.

[0021] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure belongs. Although any methods and materials similar or equivalent to those described herein can also be used in the practice or testing of the present teachings, some exemplary methods and materials are now described.

[0022] As will be apparent to those of skill in the art upon reading this disclosure, each of the individual embodiments described and illustrated herein has discrete components and features which can be readily separated from or combined with the features of any of the other several embodiments without departing from the scope or spirit of the present teachings. Any recited method can be carried out in the order of events recited or in any other order which is logically possible.

[0023] As used in this specification and the appended claims, the singular forms “a”, “an” and “the” include plural referents unless the context clearly indicates otherwise. Thus, for example, reference to “a polypeptide” includes more than one polypeptide.

[0024] The section headings used herein are for organizational purposes only and not to be construed as limiting the subject matter described.

Definitions

[0025] As used herein, the terms “protein”, “polypeptide,” and “peptide” are used interchangeably and are defined to mean a polymer of at least two amino acids covalently linked by an amide bond, regardless of length or post-translational modification (*e.g.*, glycosylation, phosphorylation, lipidation, myristilation, ubiquitination, etc.). Included within this definition are D- and L-amino acids, and mixtures of D- and L-amino acids. In some embodiments of the descriptions of polypeptides, the standard single or three letter abbreviations are used for the

genetically encoded amino acids (see, *e.g.*, IUPAC-IUB Joint Commission on Biochemical Nomenclature, "Nomenclature and Symbolism for Amino Acids and Peptides," *Eur. J. Biochem.* 138:9-37, 1984).

[0026] As used herein, the terms "polynucleotide" or "nucleic acid' are used interchangeably and are defined to mean two or more nucleosides that are covalently linked together. The polynucleotide may be wholly comprised ribonucleosides (*i.e.*, an RNA), wholly comprised of 2' deoxyribonucleotides (*i.e.*, a DNA) or mixtures of ribo- and 2' deoxyribonucleosides. While the nucleosides will typically be linked together via standard phosphodiester linkages, the polynucleotides may include one or more non-standard linkages. The polynucleotide may be single-stranded or double-stranded, or may include both single-stranded regions and double-stranded regions. Moreover, while a polynucleotide will typically be composed of the naturally occurring encoding nucleobases (*i.e.*, adenine, guanine, uracil, thymine and cytosine), it may include one or more modified and/or synthetic nucleobases, such as, for example, inosine, xanthine, hypoxanthine, *etc*. Preferably, such modified or synthetic nucleobases will be encoding nucleobases.

[0027] As used herein, the term "coding sequence" is defined to mean a portion of a nucleic acid (*e.g.*, a gene) that encodes an amino acid sequence of a protein.

[0028] As used herein the term "logistic regression" is a regression model for binary data from statistics where the logit of the probability that the dependent variable is equal to one is modeled as a linear function of the dependent variables.

[0029] As used herein the term "neural network" is a machine learning model for classification or regression consisting of multiple layers of linear transformations followed by element wise nonlinearities typically trained via stochastic gradient descent and back-propagation.

[0030] As used herein, the terms "recombinant" or "engineered" or "non-naturally occurring" are used interchangeably and are defined to mean modified polypeptides or nucleic acids which polypeptides or nucleic acids are modified in a manner that would not otherwise exist in nature, or is produced or derived from synthetic materials and/or by manipulation using recombinant techniques. Non-limiting examples include, among others, recombinant cells expressing genes that are not found within the native (non-recombinant) form of the cell or express native genes that are otherwise expressed at a different level.

[0031] As used herein, the terms "percentage of sequence identity" and "percentage homology" are used interchangeably and are defined to mean comparisons among polynucleotides or polypeptides, and are determined by comparing two optimally aligned sequences over a comparison window, where the portion of the polynucleotide or polypeptide sequence in the

comparison window may comprise additions or deletions (*i.e.*, gaps) as compared to the reference sequence for optimal alignment of the two sequences. The percentage may be calculated by determining the number of positions at which the identical nucleic acid base or amino acid residue occurs in both sequences to yield the number of matched positions, dividing the number of matched positions by the total number of positions in the window of comparison and multiplying the result by 100 to yield the percentage of sequence identity. Alternatively, the percentage may be calculated by determining the number of positions at which either the identical nucleic acid base or amino acid residue occurs in both sequences or a nucleic acid base or amino acid residue is aligned with a gap to yield the number of matched positions, dividing the number of matched positions by the total number of positions in the window of comparison and multiplying the result by 100 to yield the percentage of sequence identity. Those of skill in the art appreciate that there are many established algorithms available to align two sequences. Optimal alignment of sequences for comparison can be conducted, *e.g.*, by the local homology algorithm of Smith and Waterman, *Adv Appl Math.* 2:482, 1981; by the homology alignment algorithm of Needleman and Wunsch, *J Mol Biol.* 48:443, 1970; by the search for similarity method of Pearson and Lipman, *Proc Natl Acad Sci. USA* 85:2444, 1988; by computerized implementations of these algorithms (GAP, BESTFIT, FASTA, and TFASTA in the GCG Wisconsin Software Package), or by visual inspection (see generally, Current Protocols in Molecular Biology, F. M. Ausubel et al., eds., Greene Publishing Associates, Inc. and John Wiley & Sons, Inc., (1995 Supplement). Examples of algorithms that are suitable for determining percent sequence identity and sequence similarity are the BLAST and BLAST 2.0 algorithms, which are described in Altschul et al., *J. Mol. Biol.* 215:403-410, 1990; and Altschul et al., *Nucleic Acids Res.* 25(17):3389-3402, 1977; respectively. Software for performing BLAST analyses is publicly available through the National Center for Biotechnology Information website. BLAST for nucleotide sequences can use the BLASTN program with default parameters, *e.g.*, a wordlength (W) of 11, an expectation (E) of 10, M=5, N=-4, and a comparison of both strands. BLAST for amino acid sequences can use the BLASTP program with default parameters, *e.g.*, a wordlength (W) of 3, an expectation (E) of 10, and the BLOSUM62 scoring matrix (see Henikoff and Henikoff, *Proc Natl Acad Sci. USA* 89:10915, 1989). Exemplary determination of sequence alignment and % sequence identity can also employ the BESTFIT or GAP programs in the GCG Wisconsin Software package (Accelrys, Madison WI), using default parameters provided.

[0032] As used herein, the term "reference sequence" is defined to mean a defined sequence used as a basis for a sequence comparison. A reference sequence may be a subset of a larger

sequence, for example, a segment of a full-length gene or polypeptide sequence. Generally, a reference sequence is at least 20 nucleotide or amino acid residues in length, at least 25 residues in length, at least 50 residues in length, or the full length of the nucleic acid or polypeptide. Since two polynucleotides or polypeptides may each (1) comprise a sequence (*i.e.*, a portion of the complete sequence) that is similar between the two sequences, and (2) may further comprise a sequence that is divergent between the two sequences, sequence comparisons between two (or more) polynucleotides or polypeptide are typically performed by comparing sequences of the two polynucleotides or polypeptides over a "comparison window" to identify and compare local regions of sequence similarity. In some embodiments, a "reference sequence" can be based on a primary amino acid sequence, where the reference sequence is a sequence that can have one or more changes to the primary sequence.

[0033] As used herein, the term "substantial identity" refers to a polynucleotide or polypeptide sequence that has at least 80 percent sequence identity, at least 85 percent identity and 89 to 95 percent sequence identity, more usually at least 99 percent sequence identity as compared to a reference sequence over a comparison window of at least 20 residue positions, frequently over a window of at least 30-50 residues, wherein the percentage of sequence identity is calculated by comparing the reference sequence to a sequence that includes deletions or additions which total 20 percent or less of the reference sequence over the window of comparison. In specific embodiments applied to polypeptides, the term "substantial identity" means that two polypeptide sequences, when optimally aligned, such as by the programs GAP or BESTFIT using standard parameters, *i.e.*, default parameters, share at least 80 percent sequence identity, preferably at least 89 percent sequence identity, at least 95 percent sequence identity or more (*e.g.*, 99 percent sequence identity). Preferably, residue positions which are not identical differ by conservative amino acid substitutions.

[0034] As used herein, the terms "corresponding to", "reference to" or "relative to" are used interchangeably when used in the context of the numbering of a given amino acid or polynucleotide sequence and are defined in this context to mean the numbering of the residues of a specified reference sequence when the given amino acid or polynucleotide sequence is compared to the reference sequence. In other words, the residue number or residue position of a given polymer is designated with respect to the reference sequence rather than by the actual numerical position of the residue within the given amino acid or polynucleotide sequence. For example, a given amino acid sequence can be aligned to a reference sequence by introducing gaps to optimize residue matches between the two sequences. In these cases, although the gaps are present, the numbering of the residue in the given amino acid or polynucleotide sequence is

made with respect to the reference sequence to which it has been aligned. As such, the term "corresponding to", "reference to" or "relative to" also refers to a residue that is analogous, homologous, or equivalent to an enumerated residue in a reference polypeptide. In addition, in some embodiments, crystal structure coordinates of a reference sequence may be used as an aid in determining a homologous polypeptide residue's three dimensional structure and location of equivalent residues.

[0035] As used herein, the terms "consensus sequence" and "canonical sequence" are defined to mean an archetypical amino acid sequence against which all variants of a particular protein or sequence of interest are compared. The terms also refer to a sequence that sets forth the nucleotides that are most often present in a DNA sequence of interest. For each position of a gene, the consensus sequence gives the amino acid that is most abundant in that position in a multiple sequence alignment (MSA).

[0036] As used herein, the terms "optimal alignment" or "optimally aligned" are defined to mean the alignment of two (or more) sequences giving the highest percent identity score. For example, optimal alignment of two polypeptide sequences can be achieved by aligning the sequences such that the maximum number of identical amino acid residues in each sequence are aligned together or by using software programs or procedures described herein or known in the art. Optimal alignment of two nucleic acid sequences can be achieved by aligning the sequences such that the maximum number of identical nucleotide residues in each sequence are aligned together. Two sequences (*e.g.*, polypeptide sequences) may be deemed "optimally aligned" when they are aligned using defined parameters, such as a defined amino acid substitution matrix, gap existence penalty (also termed gap open penalty), and gap extension penalty, so as to achieve the highest similarity score possible for that pair of sequences. Optimal alignment can be done manually or by using software programs or procedures described herein or known in the art. *e.g.*, the BLASTP program for amino acid sequences and the BLASTN program for nucleic acid sequences.

[0037] As used herein, the terms "amino acid substitution" or "amino acid difference" are defined to mean a change in the amino acid residue at a position of a polypeptide sequence relative to the amino acid residue at a corresponding position in a reference sequence. Unless the context dictates otherwise, the reference sequence is the primary translation product of a gene starting at the methionine initiation codon. The positions of amino acid differences generally are referred to herein as "Xn," where n refers to the corresponding position in the reference sequence upon which the residue difference is based. For example, a "residue difference at position X as compared to SEQ ID NO" refers to a change of the amino acid residue at the

polypeptide position corresponding to position X of the SEQ ID NO. Thus, if the reference polypeptide of the SEQ ID NO has a valine at position X, then an "amino acid substitution" or "residue difference at position X as compared to the SEQ ID NO" refers to an amino acid substitution of any residue other than valine at the position of the polypeptide corresponding to position X of the SEQ ID NO. In most instances herein, the specific amino acid substitution or amino acid residue difference at a position is indicated as "XnY" where "Xn" specifies the corresponding position as described above, and "Y" is the single letter identifier of the amino acid found in the engineered polypeptide (*i.e.*, the different residue than in the reference polypeptide). In some embodiments, where more than one amino acid can appear at a specified residue position, the alternative amino acids can be listed in the form XnY/Z, where Y and Z represent alternate amino acid residues. In some instances, the present disclosure also provides specific amino acid differences denoted by the conventional notation "AnB", where A is the single letter identifier of the residue in the reference sequence, "n" is the number of the residue position in the reference sequence, and B is the single letter identifier of the residue substitution in the sequence of the engineered polypeptide. Furthermore, in some instances, a polypeptide of the present disclosure can include one or more amino acid residue differences relative to a reference sequence, which is indicated by a list of the specified positions where changes are made relative to the reference sequence.

[0038] As used herein, the terms "conservative amino acid substitution" or "conservative amino acid difference" are defined to mean a change in the amino acid at a residue position to a different residue having a similar side chain, and thus typically involves substitution of the amino acid in the polypeptide with amino acids within the same or similar defined class of amino acids. By way of example and not limitation, an amino acid with an aliphatic side chain may be substituted with another aliphatic amino acid, *e.g.*, alanine, valine, leucine, and isoleucine; an amino acid with hydroxyl side chain is substituted with another amino acid with a hydroxyl side chain, *e.g.*, serine and threonine; an amino acid having aromatic side chains is substituted with another amino acid having an aromatic side chain, *e.g.*, phenylalanine, tyrosine, tryptophan, and histidine; an amino acid with a basic side chain is substituted with another amino acid with a basic side chain, *e.g.*, lysine and arginine; an amino acid with an acidic side chain is substituted with another amino acid with an acidic side chain, *e.g.*, aspartic acid or glutamic acid; and a hydrophobic or hydrophilic amino acid is replaced with another hydrophobic or hydrophilic amino acid, respectively. Exemplary conservative substitutions are provided in **Table 1** below.

Table 1

Residue	Possible Conservative Substitutions
A, L, V, I	Other aliphatic (A, L, V, I) Other non-polar (A, L, V, I, G, M)
G, M	Other non-polar (A, L, V, I, G, M)
D, E	Other acidic (D, E)
K, R	Other basic (K, R)
N, Q, S, T	Other polar
H, Y, W, F	Other aromatic (H, Y, W, F)
C, P	None

[0039] As used herein, the terms "non-conservative substitution" or "non-conservative amino acid difference" are defined to mean a change in the amino acid at a residue position to a different residue with significantly differing side chain properties. Non-conservative substitutions may use amino acids between, rather than within, the defined groups and affects (a) the structure of the peptide backbone in the area of the substitution (*e.g.*, proline for glycine), (b) the charge or hydrophobicity, or (c) the bulk of the side chain. By way of example and not limitation, an exemplary non-conservative substitution can be an acidic amino acid substituted with a basic or aliphatic amino acid; an aromatic amino acid substituted with a small amino acid; and a hydrophilic amino acid substituted with a hydrophobic amino acid.

[0040] As used herein, the term "deletion" is defined to mean a modification of a polypeptide by removal of one or more amino acids from the reference polypeptide or modification of a nucleic acid by removal of one or more nucleotides from the reference nucleic acid. For example, deletions can comprise removal of 1 or more amino acids, 2 or more amino acids, 5 or more amino acids, 10 or more amino acids, 15 or more amino acids, or 20 or more amino acids, up to 10% of the total number of amino acids, or up to 20% of the total number of amino acids making up the reference polypepytide. Deletions can be directed to the internal portions and/or terminal portions of the polypeptide. In various embodiments, the deletion can comprise a continuous segment or can be discontinuous.

[0041] As used herein, the term "insertion" is defined to mean a modification to a polypeptide by addition of one or more amino acids from the reference polypeptide, or modification of a nucleic acid by addition of one or more nucleic acids. Insertions can be in the internal portions of the polypeptide, or to the carboxy or amino terminus. Insertions as used herein include fusion proteins as is known in the art. The insertion can be a contiguous segment of amino acids or separated by one or more of the amino acids in the reference polypeptide.

[0042] As used herein, the term "gene" is defined to mean a polynucleotide (*e.g.*, a DNA segment) that encodes a polypeptide. The term includes regions preceding and following the

coding regions as well as any intervening sequences when present (*e.g.*, introns) between individual coding segments (exons).

[0043] As used herein, the term "homologous genes" is defined to mean a pair of genes which correspond to each other and which are identical or similar to each other. The term encompasses genes that are separated by speciation (*i.e.*, the development of new species) (*e.g.*, orthologous genes), as well as genes that have been separated by genetic duplication (*e.g.*, paralogous genes).

[0044] As used herein, the terms "ortholog" and "orthologous genes" are defined to mean genes in different species that have evolved from a common ancestral gene (*i.e.*, a homologous gene) by speciation. Typically, orthologs retain the same function during the course of evolution. Identification of orthologs finds use in the reliable prediction of gene function in newly sequenced genomes.

[0045] As used herein, the terms "paralog" and "paralogous genes" are defined to mean genes that are related by duplication within a genome. Generally, paralogs tend to evolve into new functions, even though some functions are often related to the original one.

[0046] As used herein, the term "chromosomal integration" is defined to mean the process whereby an incoming sequence is introduced into the chromosome of a host cell. The homologous regions of the transforming DNA align with homologous regions of the chromosome. Subsequently, the sequence between the homology boxes is replaced by the incoming sequence in a double crossover (*i.e.*, homologous recombination). In some embodiments, homologous sections of an inactivating chromosomal segment of a DNA construct align with the flanking homologous regions of the indigenous chromosomal region of a host cell chromosome. Subsequently, the indigenous chromosomal region is deleted by the DNA construct in a double crossover (*i.e.*, homologous recombination).

[0047] As used herein, the term "homologous recombination" is defined to mean the exchange of DNA fragments between two DNA molecules or paired chromosomes at the site of identical or nearly identical nucleotide sequences. In some embodiments, chromosomal integration is homologous recombination.

[0048] As used herein, the term "isolated polypeptide" is defined to mean a polypeptide which is substantially separated from other contaminants that naturally accompany it, *e.g.*, protein, lipids, and polynucleotides. The term embraces polypeptides which have been removed or purified from their naturally-occurring environment or expression system (*e.g.*, host cell or *in vitro* synthesis).

[0049] As used herein, the term "specificity" as used in reference to an biocatalyst or enzyme is defined to mean the discrimination of the biocatalyst for a substrate compound.

[0050] As used herein, the term "stringent hybridization conditions" is defined to mean hybridizing in 50% formamide at 5XSSC at a temperature of 42 °C and washing the filters in 0.2XSSC at 60 °C. (1XSSC is 0.15M NaCl, 0.015M sodium citrate.) Stringent hybridization conditions also encompasses low ionic strength and high temperature for washing, for example 0.015 M sodium chloride/0.0015 M sodium citrate/0.1% sodium dodecyl sulfate at 50 °C; hybridization with a denaturing agent, such as formamide, for example, 50% (v/v) formamide with 0.1% bovine serum albumin/0.1% Ficoll/0.1% polyvinylpyrrolidone/50 mM sodium phosphate buffer at pH 6.5 with 750 mM sodium chloride, 75 mM sodium citrate at 42 °C; or 50% formamide, 5XSSC (0.75 M NaCl, 0.075 M sodium citrate), 50 mM sodium phosphate (pH 6.8), 0.1% sodium pyrophosphate, 5X Denhardt's solution, sonicated salmon sperm DNA (50 µg/ml), 0.1% SDS, and 10% dextran sulfate at 42 °C, with washes at 42 °C in 0.2XSSC (sodium chloride/sodium citrate) and 50% formamide at 55 °C, followed by a high-stringency wash consisting of 0.1XSSC containing EDTA at 55 °C.

[0051] As defined herein, the term "heterologous" polynucleotide or polypeptide is defined to mean any polynucleotide or polypeptide that is not naturally found in a host cell. As such, the term includes polynucleotides that are removed from a host cell, subjected to laboratory manipulation, and then reintroduced into a host cell. In some embodiments, the introduced polynucleotide expresses the heterologous polypeptide.

[0052] As used herein, the term "codon optimized" is defined to mean changes in the codons of the polynucleotide encoding a protein to those preferentially used in a particular organism such that the encoded protein is efficiently expressed in the organism of interest. Although the genetic code is degenerate in that most amino acids are represented by several codons, called "synonyms" or "synonymous" codons, it is well known that codon usage by particular organisms is nonrandom and biased towards particular codon triplets. This codon usage bias may be higher in reference to a given gene, genes of common function or ancestral origin, highly expressed proteins versus low copy number proteins, and the aggregate protein coding regions of an organism's genome.

[0053] As used herein, the term "control sequence" is defined to include all components, which are necessary or advantageous for the expression of a polynucleotide and/or polypeptide of the present disclosure. Each control sequence may be native or foreign to the nucleic acid sequence encoding the polypeptide. Such control sequences include, but are not limited to, a leader, polyadenylation sequence, propeptide sequence, promoter, signal peptide sequence, and transcription terminator. At a minimum, the control sequences include a promoter, and transcriptional and where appropriate, translational stop signals. The control sequences may be

provided with linkers for the purpose of introducing specific restriction sites facilitating ligation of the control sequences with the coding region of the nucleic acid sequence encoding a polypeptide.

[0054] As used herein, the term "operably linked" is defined to mean a configuration in which a control sequence is appropriately placed (*i.e.*, in a functional relationship) at a position relative to a polynucleotide of interest such that the control sequence directs or regulates the expression of the polynucleotide and/or polypeptide of interest.

[0055] As used herein, the term "promoter sequence" is defined to mean a nucleic acid sequence that is recognized by a host cell for expression of a polynucleotide of interest, such as a coding sequence or gene. The promoter sequence contains transcriptional control sequences, which mediate the expression of a polynucleotide of interest. The promoter may be any nucleic acid sequence which shows transcriptional activity in the host cell of choice including mutant, truncated, and hybrid promoters, and may be obtained from genes encoding extracellular or intracellular polypeptides either homologous or heterologous to the host cell.

[0056] As used herein, the terms "microbial," "microbial organism" or "microorganism" are defined to mean any organism that exists as a microscopic cell that is included within the domains of archaea, bacteria or eukarya. Therefore, the term is intended to encompass prokaryotic or eukaryotic cells or organisms having a microscopic size and includes bacteria, archaea and eubacteria of all species as well as eukaryotic microorganisms such as yeast and fungi. The term also includes cell cultures of any species that can be cultured for the production of a biochemical.

[0057] As used herein, the terms "wild-type" is defined to mean the form found predominantly in nature. For example, a wild-type polypeptide or polynucleotide sequence is a sequence predominantly present in an organism that can be isolated from a source in nature and which has not been intentionally modified by human manipulation.

Methods for Identifying Binding Interactions

[0058] The disclosure provides methods, systems, and computer-readable storage media for identifying binding interactions between TCRs and MHC-antigens. The methods, systems, and computer-readable storage media are able to identify TCR and MHC-antigen binding interactions when the TCR and/or MHC-antigen are different from the training data. This is because the methods, systems, and computer-readable storage media are trained for uncertainty in the test data arising from information the methods, systems, and computer-readable storage media have not experienced. Generating accurate predictions with these uncertainties is made possible by several innovative algorithmic designs including, for example, identifying the

functional sequences of the TCRs and MHC-antigens, including in the dataset iterations of successful binding pairs where some of the functional sequences are removed (so the methods infer successful binding without all of the sequence information), running multiple iterations of the predictive model and then combining these results, training an embedder to produce vectors for each TCR and/or MHC-antigen which vector has multiple dimensions (e.g., 16-64), and having multiple embedders trained for TCRs or for MHC-antigens. The methods, systems, and computer-readable storage media herein have been used to identify novel TCRs for binding to a MHC-TAA (e.g., a mutant PIK3CA) that had not been previously discovered. This novel TCR has very different CDR sequences from the previously known TCRs that bind to the mutant PIK3CA.

[0059] FIG. 1 and FIG. 2 are block diagrams illustrating an exemplary process for using a machine learning model to predict TCR binding specificities. In the first step, a set of TCR embeddings are made, the input sequence information is embedded using a vector representation (e.g., 16-128 dimensions, or 32-64 dimensions, or 64 dimensions) learned by the model during training (Amino Acid Embedder). TCRs are then encoded via independent multi-layer convolutional components (Convolutional Encoder) into vectors of important features.

[0060] The TCR embeddings may include a vector representation of TCR sequences generated by an ensemble trained to generate a multidimensional vector for the TCR (e.g., 16-128 dimensions, or 32-64 dimensions, or 64 dimensions). Alternatively, the TCR embeddings may be made by an auto-encoder or other TCR numeric embedding layer. The TCR embedding layer may be trained on a TCR training dataset that includes multiple training TCR sequences. The TCR training dataset may include TCR data, for example, a matrix or other structured data representation of one or more biochemical properties of amino acids included in each of the training TCR protein sequences. The embedding layer may include a plurality of layers that encode the structured data representations into multidimensional vectors. The embedding layer may also include a plurality of decoder layers that generate a reconstruction of the structured data representations based on the vectors generated by the encoder layers. Accordingly, the TCR embeddings may be validated by comparing the structured data representations input into the encoder layers to the reconstruction of the structured data representations generated by the decoder layers. A high degree of similarity (i.e., % similar or any other measure of similarity that is at or above a pre-defined similarity threshold) between the input structured data representations and the reconstruction may indicate accurate TCR embeddings.

[0061] The TCR input into the embedder is a string that represents the alpha and the beta chain of a TCR. For example, the representation of the string can be: TRAV + TRA_CDR3 + TRAJ +

TRBV + TRB_CDR3 + TRB_J, where each component is separated by a "-" character. V and J genes are represented by short "pseudo-sequences" of amino acids that under prior computational analysis were determined as most important for interaction (e.g., TRAV12-1 is "NSASQ-VYSSG"). For each positive binding pair (TCR with an MHC-antigen) multiple iterations are made in which one of more of the above string are removed from the input data for the embedder (so each positive binding pair makes multiple sets of input data). This allows the model to make predictions about TCRs where any number of components are missing or changed from the training data (e.g., beta chain only, or cdr3 only). In these cases an "X" can be passed in place of the missing component. Examples of TCR inputs:

- NSASQ-VYSSG-CAVNPPDTGFQKLVF-DRGS-MDHE-SYDVK-CASSLSFRQGLREQY-SYNE (no missing information)
- NSASQ-VYSSG-CAVNPPDTGFQKLVF-X-X-X-X (alpha chain only)
- X-X-CAVNPPDTGFQKLVF-X-X-X-CASSLSFRQGLREQY-X (cdr3 only)

[0062] In the second step, a set of embeddings are made for the MHC-antigen for a plurality of pMHCs and antigens (e.g., TAAs). MHC-antigen input sequence information is embedded using a vector representation (e.g., 16-128 dimensions, or 32-64 dimensions, or 64 dimensions) learned by the model during training (Amino Acid Embedder). MHC-antigens are then encoded via independent multi-layer convolutional components (Convolutional Encoder) into vectors of important features. Each of the MHC-antigen embeddings may include a numeric representation of one or more pMHCs generated by a multi-layer neural network or other pMHC numeric embedding layer. For example, to generate the pMHC embeddings, the pMHC numeric embedding layer may be trained on a pMHC training dataset including textual representations of pMHCs. The pMHC numeric embedding layer may convert the sequence data for a group of input MHC-antigen sequences into multidimensional vectors.

[0063] The MHC-antigen input into the embedder is a string that represents a pMHC complex. It may consist of the amino acids for an antigen target sequence and a pseudo-sequence of amino acids for an MHC allele, separated by "-". The MHC allele inputs are the amino acids of the MHC involved in antigen presentation and interaction with the TCR. As with the TCR inputs, either component can be missing and replaced with an "X". Examples:

- YFAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY-KLVVVGACGVGK
- X-KLVVVGACGVGK (antigen only)
- YFAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY-X (allele only)

[0064] The amino acid embedder for the TCR or pMHC takes a string of amino acids separated by "-" (and with optional "X"s) and learns during training how to convert this into a vector which can be used by downstream model components. We convert each character in the vocabulary of amino acids (with the addition of "-" and "X") into a number (so each sequence becomes a list of numbers, e.g., where "A" becomes 1 and so on) and can use the pytorch nn.Embedding module to map each amino acid onto a learned embedding. The dimensionality of the embedding learned by the final model can be in the range of 2-2000, or 2-256, or 32-128, or 16-64, or 32-64, or 2, or 4, or 8, or 16, or 32, or 64, or 128.

[0065] The convolutional encoder for the TCR or the pMHC learn how to encode important representations of TCRs or pMHC that can be used for downstream prediction. This encoder can have a series of convolutional layers, or it is also possible to use other popular deep learning architectures such as Transformers and LSTMs. Concretely, this encoder consists of a series of 3 convolutional plus pooling layers. The output channels for the TCR encoder are [64, 128, 256]. And the output channels for the pMHC encoder are [32, 64, 128]. A kernel size of 3 and stride of 1 can be used for all convolutions. Independent Convolutional Encoders are trained for TCRs or pMHC sequences.

[0066] The TCR and pMHC vectors are inputted to a concatenator, and concatenated together for the Prediction Module or Dense Layer. The Prediction Module or Dense Layer or "fully connected" layer is capable of computing non-linear relationships between TCR and pMHC vectors. Input dimensionality is X (the size of the concatenated TCR and target vectors) and output size is 256 or 512, or 1024, or 2056 for the model. As an example, a linear layer or a convolutional layer or sub-network of any complexity can be used for this component. Following this layer the model will output a score between 0-1 representing the likelihood of interaction between a TCR and target.

[0067] FIG. 1 and FIG. 4 show ensembles of outputs from the predictive model/dense layer that offer better performance when predicting TCR interaction with common antigens, and dramatically improve prediction performance against rare or novel antigens never encountered by models during training (>+20% AUC). FIG. 4 refers to TAPIR1 to TAPIRN and the term TAPIR refers to machine learning system of FIG. 2. The predictive model/dense layer can be a convolutional neural network with 3 layers with each layer having a convolutional and pooling portion. To produce a prediction score for an ensemble, one can combine the data in numerous known ways, for example, an average score produced by averaging the outputs of each ensemble member (e.g., ensembles of 16, 32, 64, 128, or more members). Without wishing to be bound by theory, even when trained on completely the same data, predictive models can learn to

recognize slightly different patterns from the same data inputs based on randomness in the initializations of model weights and the order in which training examples are encountered during training. When a model is asked to make predictions about data it has never seen before, these small differences can result in quite a large variance in output scores. By averaging the scores across many models, this variance can be smoothed out making stronger overall predictions when the predictive model sees new information in the test data.

[0068] When converting a set of ensemble scores into a single output score the following approach can be used: compute mean, median, mode, minimum, or maximum; adjust the score by multiplying or dividing by a standard deviation or variance or another function that captures the agreement/disagreement of the ensemble (e.g., max(ensemble_score)-min(ensemble_scores)); use an ensemble agreement measure such as standard deviation or variance directly as the output score, or as a secondary output score to help user or application understand the certainty of the ensemble about the prediction; do any of the above on a subset of the ensemble models chosen for their performance characteristics on a related task (e.g., performance on a validation set); train a neural network to process ensemble member scores into a single score, and distill the ensemble by training a new neural network model to predict the ensemble member scores and then using that new model.

[0069] Normalizing or adjusting a single output score with respect to a specific TCR called T and a specific antigen called A can be done, for example, by obtaining a TCR percentile score that computes the percentile of T's score against A as compared to a background distribution of TCRs also scored against A. This background distribution of TCRs could be training data, validation data, new experimental data, and or any other data source. The background distribution of TCRs might have specific characteristics such as being known to interact with A, or known not to interact with A, or that are associated with another property such as a patient, MHC allele, or secondary antigen B. An antigen percentile score can also be obtained by computing the percentile of T's score against A as compared to a background distribution of antigens also scored against T. This background distribution of antigens might have properties such as being common, uncommon (e.g., neoantigens), a specified mix of common or uncommon, being presented by the same or different MHC alleles, or known not to present by MHC. A TCR norm score can be used instead of or in addition to the TCR percentile. The TCR norm score is obtained by taking the percentile w.r.t. some TCR background distribution, instead divide by the mean (or some other statistic) of that distribution. An antigen norm score can also be used instead of or in addition to the antigen percentile. The antigen norm score is

obtained by taking the percentile w.r.t. some antigen background distribution, instead divide by the mean (or some other statistic) of that distribution.

[0070] FIG. 5 shows training of the ML and neural network components of the methods and systems using data from VDJDB and data obtained from the methods disclosed in PCT/US2022/036841 filed in July 12, 2022, or USSN 17/863,200 filed July 12, 2022, each of which is incorporated by reference in its entirety for all purposes. These data sets include T-cell receptors paired (single chains or paired chains) with antigen targets. In the data augmentation step, the training data examples is "expanded" using a masking procedure that helps the models learn to make predictions about individual components of sequences (e.g., just CDR3) in isolation. After data augmentation, negative training examples can be created by randomly assigning new pMHC targets for positive example TCR to create false TCR-pMHC pairs. Next, all training data (including positives and negatives) is shuffled to randomize the order in which it is presented to the model, and models are trained for a number of epochs, shuffling the data again between epochs. A final model is saved after the last epoch.

[0071] VDJDB is a public database of ~60,000 TCRs associated with antigen targets collected from decades of previous studies. The methods and systems described herein use both single chain and paired chain examples for training the models, which has not been done with previously published models. In addition, the methods and systems described here are trained on data for all the targets in the database, even targets with very limited numbers of examples which also has not been done with prior published models. Training data obtained from the methods disclosed in PCT/US2022/036841 filed July 12, 2022, or USSN 17/863,200 filed July 12, 2022, adds about 3000 TCRs paired with antigens, and includes about 700 novel antigens that are not found in VDJDB.

[0072] Data augmentation can expand a set of positive training examples by generating additional positive examples with various components of TCRs and pMHC sequences masked. This trains the model to learn the importance of individual components of sequences (e.g CDR3) to predict interactivity, and allows the model to make predictions given incomplete information (e.g., just beta chain TCR), and/or uncertainty in the test data (e.g., novel antigen targets or novel TCRs). Specifically, for each (TCR, pMHC) pair in the dataset of positives, new examples are generated with the following components masked ('X' character indicates mask):

- Mask Beta-chain TCR: (NSASQ-VYSSG-CAVNPPDTGFQKLVF-DRGS-X-X-X-X, pMHC unchanged)

- Mask Alpha-chain TCR: (X-X-X-X-MDHE-SYDVK-CASSLSFRQGLREQY-SYNE, pMHC unchanged)
- Mask V/J Genes: (X-X-CAVNPPDTGFQKLVF-X-X-X-CASSLSFRQGLREQY-X, pMHC unchanged)
- Mask antigen target: (TCR unchanged, YFAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY-X)
- Mask HLA pseudosequence: (TCR unchanged, X-KLVVVGACGVGK)

[0073] Negative data can be added at the data augmentation step by repeatedly (e.g., 3x) shuffling the pMHC targets associated with TCRs and labeling the resulting pairs as negative. While it is possible that randomly shuffling pMHC targets could result in correct pairings this is a low probability event, and better than having different distributions of pMHC targets for the positive and negative examples, which could introduce bias into the model.

[0074] The shuffling step simply shuffles the order in which all training data (both positive and negative) will be presented to a model over the course of an epoch.

[0075] A training epoch is the period over which a model is presented with all data in the training set (so a model trained for 20 epochs will observe all data 20 times). Training consists of presenting the model with examples, computing the loss of the examples with respect to the correct labels, then backpropagating that loss to update model parameters. The process itself involves two important hyperparameters, batch size and learning rate. Batch size is the number of examples a model observes simultaneously when computing predictions, loss, and gradients for backprop. This can be 256 for the models in certain implementations. The batch size can also be 128, or 256, or 512, or 1024, or 2048. Learning rate describes the rate at which model will be updated based on the training loss (higher learning rate=faster, more aggressive updates). Learning rate can be 0.001 for models in certain implementations. The learning rate can also be 0.01, or 0.0001, or 0.00001, and may vary with the model type and number of parameters in the model.

[0076] FIG. 6 shows a more detailed example of data augmentation described above.

[0077] FIG. 7 shows a computational screening process that filters and ranks hits from any kind of TCR screening assay. This increases the likelihood of discovering hits that validate with downstream in vitro functional and killing assays. The TCR screening assay can be any method that produces pairs of T-cell receptor and antigen target of interest. Such methods include, for example, those described in PCT/US2022/036841 filed July 12, 2022, or USSN 17/863,200 filed July 12, 2022, and tetramer/dextramer based methods, and sort-then-sequence methods on

various activation markers (e.g., CD69) can also be used. Other methods include any method that tests interactions between T-cell receptor and antigen targets. Methods often measure binding between T-cell receptors and the target or measure T-cell activation after a T-cell carrying the T-cell receptor of interest is stimulated by a target. The data can be generated using conventional TCR screening methods or the methods described in USSN 17/863,200 filed July 12, 2022. Conventional TCR screening methods include phage display, yeast display, hybridoma technology, tetramer staining, multimer staining, ELISpot assays, T-cell proliferation assays, T-cell killing assays, cytokine release assays, surface marker assays, trogocytosis assay (https://pubmed.ncbi.nlm.nih.gov/17406507/) , lentivirus entrance assay (https://pubmed.ncbi.nlm.nih.gov/35396484/), Tscan (https://pubmed.ncbi.nlm.nih.gov/31398327/), SABR (https://pubmed.ncbi.nlm.nih.gov/30700902/), and spatial genomics assays (https://pubmed.ncbi.nlm.nih.gov/35707680/).

[0078] T-cell activation assays can measure any suitable T-cell activation marker(s) can include, for example, CD25, CD28, CD40L, CD45RO, CD45RA, CD69, CD122, CD127, CD137, CD152, CD154, CD178, CD183, CD185, CD194, CD196, CD197, CD200R, CD223, CD244, CD279, CD278, CD314, CD366, CD107a, or HLA-DR. Cytokines can be measured to determine T-cell activation include IL-2, IL-4, IL-6, IL-7, IL-9, IL-10, IL-12, IL-15, IL-17, IL-21, IFN-γ (Interferon-gamma), TNF-α (Tumor Necrosis Factor-alpha), TGF-β (Transforming Growth Factor-beta), GM-CSF (Granulocyte-Macrophage Colony-Stimulating Factor), IL-5, IL-13, IL-22, IL-23, IL-27, IL-35, IL-39, IL-40, IL-41. All of these screening methods can be combined with sequencing and molecular cloning to determine the exact sequences of T-cell receptors and antigens.

[0079] The end result of this step is a list of T-cell receptor sequences (usually paired with antigen, but could be unpaired), along with standard sequencing metrics such as TCR clone counts, UMI and read counts for the TCR, as well as method specific metrics like tetramer/dextramer UMI/reads, etc. The list of T-cell receptor sequences can include, for example, alpha chain only, beta chain only, paired alpha beta chain, gamma chain only, delta chain only, or paired gamma delta chain, associated with a target antigen. The following standard sequencing metrics can be incorporated into training data for either data processing or modeling training: TCR clone counts, unique molecule index (UMI) count and read counts for the TCR, tetramer/dextramer UMI/reads, antigen/target UMI/reads, and sequencing quality metrics.

[0080] The TAPIR ensemble step uses an ensemble of TAPIR models as described above to label each TCR sequence with a score that describes the likelihood of a TCRs association with an antigen target of interest. In addition to this score, which we call the "raw score", the methods and systems described herein produce two other scores: a "TCR percentile score" and an "Antigen percentile score". The TCR percentile score is the percentile of a TCRs score with respect to a fixed background distribution of approximately 1000 TCRs. A high TCR percentile score like 99.5% would indicate that this TCR has a higher raw score against the target of interest than 99.5% of the TCRs in the fixed background distribution. The antigen percentile score is the percentile of a TCR's score against the antigen target of interest with respect to a fixed set of around 250 other antigens. A high antigen percentile score like 99% would indicate that, for a candidate TCR, the score for the antigen of interest is higher against this TCR than 99% of other antigens in the background set. These metrics provide a more human interpretable lens on the raw score, and in the case of the antigen percentile score, ensure that a selected TCR gives a score higher for the antigen of interest than most other possible antigen targets.

[0081] Filtering and ranking of candidates can be done by many methods that are known in the art for filtering and ranking candidates based on the computed and collected metrics. For tetramer/dextramer experiments in particular, hits can be filtered based on minimum UMI/read counts for both TCR sequences (>1 UMI, >10 reads for each chain), dextramers (>4 UMI), TCR percentile scores of 90%, and antigen percentile scores of 80%. The remaining candidates can then be ranked based on raw TAPIR scores.

[0082] Validation experiments can be done by many methods known in the art, including for example, candidate TCRs can be tested using binding assays, or in vitro or in vivo functional assays. For example, the T-cell receptor of interest can be engineered into human immune cells, such as T-cells, macrophages, NK cells, B-cells, and NKT cells, employing methods including viral transduction, plasmid transfection, gene editing, electroporation, nanoparticle-mediated delivery, mRNA introduction, or mini-circle systems. After the immune cells express this T-cell receptor and are mixed with target cells, activities of T-cell activation markers (e.g. CD25, CD69, CD137, CD134, HLA-DR, CD107, Granzyme B, trogocytosis, or morphology change) or killing properties can be compared to negative controls to confirm the function of the TCR candidate.

[0083] FIG. 8 shows a convolutional encoder for an amino acid sequence example. The convolutional encoder encodes an amino acid sequence into a vector through three convolutions with batch normalization and pooling. Each layer performs a 1D convolution over the sequence to produce a new, deeper output channel dimension (e.g., initial embedding dimension E is

transformed into a new dimension of D1). Then values are batch normalized over the output channel dimension and processed with MaxPool. The final output matrix is flattened across all D3 channels to produce a vector encoding for the sequence. For TCR sequences the output channel dimensions are D1=64, D2=128, D3=256. For antigen/MHC sequences the output channel dimensions are D1=32, D2=64, D3=128. Output channels as small as 8 or as large as 1280 would be feasible to use.

[0084] The methods and compositions disclosed herein are based on studies to determine better predictive limits of each parameter—ultimately resulting in a pattern that is more likely to predict TCRs that will bind to processed antigens, and bind those processed antigens in the context of certain HLA molecules. Thus, the methods combine multiple measurements and methods of calculation across a broad range of parameters gathered from a variety of curated database algorithms (resources). Although the idea of combining multiple resources (for example, Calis J J A Immunogenetics 67:85 (2015); Doytchinova I A et al. BMC Bioinformatics 7:131 (2016)) and the use of machine learning (reviewed by Luo H et al. Bioinformatics and Biology Insights S3:21 (2015)) are known to be ways to improve predictive accuracy, to our knowledge, prior to this invention, others had not embedded TCR, antigen and MHC information in the manner herein described, used separate encoders for the TCR and pMHC, augmented the training data by creating data that purposefully is missing information so that the model can be trained for uncertainty and missing data and/or to make predictions when the test data is different from the training data, or ensembling many models together to improve performance and reduce the variance in predictions especially against targets never observed by the model in training, resulting in a method that achieves markedly improved predictive performance.

[0085] In a particular aspect, a method for ranking TCR-pMHC pairs using machine learning includes providing, by one or more processors, input data indicating a TCR, and a pMHC to one or more machine learning (ML) models to identify TCR candidates which can bind to pMHC. The one or more ML models are configured to score the binding interaction between a TCR and a pMHC based on chemical or physical properties of the molecules. The method also includes, determining by the one or more processors, a respective composite score based on one or more predictive model runs for a TCR with a pMHC. The method further includes generating, by the one or more processors, an output based on the subset of candidate TCRs and the pMHC(s).

Systems

[0086] FIG. 3 illustrates a diagrammatic representation of computer system within which a set of instructions may be executed to cause the computer to perform any one or more of the

methods discussed herein. Specifically, FIG. 3 shows a diagrammatic representation of the computer system, within which instructions (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the computer to perform any one or more of the methods discussed herein may be executed. For example, the instructions may cause the computer to implement the methods described with respect to FIGS. 1, 2, and 4-8, respectively.

[0087] The instructions and/or training data transform the general, non-programmed machine (computer system) into a particular machine programmed to carry out the described and illustrated functions in the manner described. In alternative embodiments, the machine operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine may comprise, but not be limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (PDA), an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the instructions, sequentially or otherwise, that specify actions to be taken by the machine. Further, while only a single machine (computer system) is illustrated, the term “machine” shall also be taken to include a collection of machines that individually or jointly execute the instructions to perform any one or more of the methods discussed herein.

[0088] Examples of a computing device can include logic, one or more components, circuits (e.g., modules), or mechanisms. Circuits are tangible entities configured to perform certain operations. In an example, circuits can be arranged (e.g., internally or with respect to external entities such as other circuits) in a specified manner. In an example, one or more computer systems (e.g., a standalone, client or server computer system) or one or more hardware processors (processors) can be configured by software (e.g., instructions, an application portion, or an application) as a circuit that operates to perform certain operations as described herein. In an example, the software can reside (1) on a non-transitory machine readable medium or (2) in a transmission signal. In an example, the software, when executed by the underlying hardware of the circuit, causes the circuit to perform the certain operations.

[0089] In an example, a circuit can be implemented mechanically or electronically. For example, a circuit can comprise dedicated circuitry or logic that is specifically configured to perform one or more techniques such as discussed above, such as including a special-purpose

processor, a field programmable gate array (FPGA) or an application-specific integrated circuit (ASIC). In an example, a circuit can comprise programmable logic (e.g., circuitry, as encompassed within a general-purpose processor or other programmable processor) that can be temporarily configured (e.g., by software) to perform the certain operations. It will be appreciated that the decision to implement a circuit mechanically (e.g., in dedicated and permanently configured circuitry), or in temporarily configured circuitry (e.g., configured by software) can be driven by cost and time considerations.

[0090] Accordingly, the term "circuit" is understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily (e.g., transitorily) configured (e.g., programmed) to operate in a specified manner or to perform specified operations. In an example, given a plurality of temporarily configured circuits, each of the circuits need not be configured or instantiated at any one instance in time. For example, where the circuits comprise a general-purpose processor configured via software, the general-purpose processor can be configured as respective different circuits at different times. Software can accordingly configure a processor, for example, to constitute a particular circuit at one instance of time and to constitute a different circuit at a different instance of time.

[0091] In an example, circuits can provide information to, and receive information from, other circuits. In this example, the circuits can be regarded as being communicatively coupled to one or more other circuits. Where multiple of such circuits exist contemporaneously, communications can be achieved through signal transmission (e.g., over appropriate circuits and buses) that connect the circuits. In embodiments in which multiple circuits are configured or instantiated at different times, communications between such circuits can be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple circuits have access. For example, one circuit can perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further circuit can then, at a later time, access the memory device to retrieve and process the stored output. In an example, circuits can be configured to initiate or receive communications with input or output devices and can operate on a resource (e.g., a collection of information).

[0092] The various operations of method examples described herein can be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors can constitute processor-implemented circuits that operate to perform one or more operations or functions. In an example, the circuits referred to herein can comprise processor-implemented circuits.

[0093] Similarly, the methods described herein can be at least partially processor-implemented. For example, at least some of the operations of a method can be performed by one or processors or processor-implemented circuits. The performance of certain of the operations can be distributed among the one or more processors, not only residing within a single machine, but deployed across a number of machines. In an example, the processor or processors can be located in a single location (e.g., within a home environment, an office environment or as a server farm), while in other examples the processors can be distributed across a number of locations.

[0094] The one or more processors can also operate to support performance of the relevant operations in a "cloud computing" environment or as a "software as a service" (SaaS). For example, at least some of the operations can be performed by a group of computers (as examples of machines including processors), with these operations being accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., Application Program Interfaces (APIs).)

[0095] Example embodiments (e.g., apparatus, systems, or methods) can be implemented in digital electronic circuitry, in computer hardware, in firmware, in software, or in any combination thereof. Example embodiments can be implemented using a computer program product (e.g., a computer program, tangibly embodied in an information carrier or in a machine readable medium, for execution by, or to control the operation of, data processing apparatus such as a programmable processor, a computer, or multiple computers).

[0096] A computer program can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a software module, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

[0097] In an example, operations can be performed by one or more programmable processors executing a computer program to perform functions by operating on input data and generating output. Examples of method operations can also be performed by, and example apparatus can be implemented as, special purpose logic circuitry (e.g., a field programmable gate array (FPGA) or an application-specific integrated circuit (ASIC)).

[0098] The computing system can include clients and servers. A client and server are generally remote from each other and generally interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective

computers and having a client-server relationship to each other. In embodiments deploying a programmable computing system, it will be appreciated that both hardware and software architectures require consideration. Specifically, it will be appreciated that the choice of whether to implement certain functionality in permanently configured hardware (e.g., an ASIC), in temporarily configured hardware (e.g., a combination of software and a programmable processor), or a combination of permanently and temporarily configured hardware can be a design choice. Below are set out hardware (e.g., a computing device) and software architectures that can be deployed in example embodiments.

[0099] In a networked deployment, the computing device can operate in the capacity of either a server or a client machine in server-client network environments. In an example, computing device can act as a peer machine in peer-to-peer (or other distributed) network environments. The computing device can be a personal computer (PC), a tablet PC, a set-top box (STB), a Personal Digital Assistant (PDA), a mobile telephone, a web appliance, a network router, switch or bridge, or any machine capable of executing instructions (sequential or otherwise) specifying actions to be taken (e.g., performed) by the computing device. Further, while only a single computing device is illustrated, the term "computing device" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[00100] Example computing device can include a processor (e.g., a central processing unit CPU), a graphics processing unit (GPU) or both), a main memory and a static memory, some or all of which can communicate with each other via a bus. The computing device can further include a display unit, an alphanumeric input device (e.g., a keyboard), and a user interface (UI) navigation device (e.g., a mouse). In an example, the display unit, input device and UI navigation device can be a touch screen display. The computing device can additionally include a storage device (e.g., drive unit), a signal generation device (e.g., a speaker), a network interface device, and one or more sensors, such as a global positioning system (GPS) sensor, compass, accelerometer, or other sensor.

[00101] The storage device can include a machine readable medium on which is stored one or more sets of data structures or instructions (e.g., software) embodying or utilized by any one or more of the methodologies or functions described herein. The instructions can also reside, completely or at least partially, within the main memory, within static memory, or within the processor during execution thereof by the computing device. In an example, one or any combination of the processor, the main memory, the static memory, or the storage device can constitute machine readable media.

[00102] While the machine readable medium is illustrated as a single medium, the term "machine readable medium" can include a single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that configured to store the one or more instructions. The term "machine readable medium" can also be taken to include any tangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of the present disclosure or that is capable of storing, encoding or carrying data structures utilized by or associated with such instructions. The term "machine readable medium" can accordingly be taken to include, but not be limited to, solid-state memories, and optical and magnetic media. Specific examples of machine-readable media can include non-volatile memory, including, by way of example, semiconductor memory devices (e.g., Electrically Programmable Read-Only Memory (EPROM), Electrically Erasable Programmable Read-Only Memory (EEPROM)) and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks.

[00103] The instructions can further be transmitted or received over a communications network using a transmission medium via the network interface device utilizing any one of a number of transfer protocols (e.g., frame relay, IP, TCP, UDP, HTTP, etc.). Example communication networks can include a local area network (LAN), a wide area network (WAN), a packet data network (e.g., the Internet), mobile telephone networks (e.g., cellular networks), Plain Old Telephone (POTS) networks, and wireless data networks (e.g., IEEE 802.11 standards family known as Wi-Fi®, IEEE 802.16 standards family known as WiMax®), peer-to-peer (P2P) networks, among others. The term "transmission medium" shall be taken to include any intangible medium that is capable of storing, encoding or carrying instructions for execution by the machine, and includes digital or analog communications signals or other intangible medium to facilitate communication of such software.

[00104] A presentation identification system can be one or more computer models, embodied in a computing system, that receives data associated with a set of MHC alleles and may determine the likelihood that an antigen peptide sequence can be presented by one or more MHC alleles. This is useful in a variety of contexts. One specific use case for a presentation identification system is that it is able to receive nucleotide sequences of candidate antigens associated with a set of MHC alleles from tumor cells of a patient and determine likelihoods that the candidate antigens will be presented by one or more of the associated MHC alleles of the tumor and/or induce immunogenic responses in the immune system of the patient. Those candidate antigens

with high likelihoods as determined by the system can be used to find TCRs that will bind these antigens.

[00105] The computing device may generate an output that indicates the TCR candidates paired with antigens of interest for testing and potential trial. For example, the computing device may initiate display of a graphical user interface (GUI) that includes text, images, graphics, or a combination thereof, that indicate the TCR candidate and the pMHC, rank the TCR candidate in relation to other TCR candidates, predicted properties of the TCR candidates, disease states or conditions that are predicted to be treated using the TCRs, and the like. In some implementations, the computing device may operate as a training device that trains the machine learning models and provides the machine learning models (or data indicative of the configuration of the machine learning models) to client devices for screening at the client devices.

[00106] In another particular aspect, a system for TCR candidate screening using machine learning includes a memory and one or more processors communicatively coupled to the memory. The one or more processors can be configured to provide input data indicating a candidate TCR to one or more ML models to identify a cluster of multiple clusters to which the candidate TCR can be assigned. The one or more ML models can be configured to assign TCR candidates to one of the multiple clusters based on chemical or physical properties of the TCR candidates. Each cluster of the multiple clusters can be associated with a respective plurality of TCR candidates assigned to the cluster based on chemical or physical properties of the respective plurality of pMHCs. The one or more processors can be configured to determine, for each pMHC of a plurality of pMHCs associated with the cluster, a respective composite score based on one or more comparisons between the candidate TCR and the pMHC. The one or more processors can be configured to identify a subset of pMHCs from the plurality of pMHCs associated with the cluster based on the composite scores. The one or more processors can be further configured to generate an output based on the subset of PMHCs and the candidate TCRs.

[00107] In another particular aspect, a non-transitory computer readable storage medium stores instructions that, when executed by one or more processors, cause the one or more processors to perform operations for TCR candidates screening using machine learning. The operations can include providing input data indicating a candidate TCR to one or more ML models to identify a cluster of multiple clusters to which the candidate TCR is assigned. The one or more ML models can be configured to assign TCRs to one of the multiple clusters based on chemical or physical properties of the TCRs. Each cluster of the multiple clusters can be associated with a respective plurality of pMHCs assigned to the cluster based on chemical or physical properties of the

respective plurality of pMHCs. The operations can also include for each pMHC of a plurality of pMHCs associated with the cluster, determining a respective composite score based on one or more comparisons between the candidate TCR and the pMHC. The operations can include identifying a subset of pMHCs from the plurality of pMHCs associated with the cluster based on the composite scores. The operations further can include generating an output based on the subset of pMHCs and the candidate TCRs.

Machine Learning Tools

[00108] The machine learning model can be a deep neural network (e.g., a convolutional neural network) comprising multiple processing layers, which each layer's outputs serving as the next layer's inputs. The architecture can enable the model to learn directly from 3D structures and to learn effectively given a very small amount of experimental data. Certain embodiments use other machine learning algorithms such as, without limitation, SVMs, random forests, decision trees, linear and logistic regressions, and other deep neural networks. Certain embodiments can augment the neural network such as, without limitation, the use of attention-based mechanisms (e.g., transformers), residual layers, hierarchical coarse-graining, regularization, and other activation and normalization layers.

[00109] Certain aspects can use multiple different prediction models such as, without limitation, in the generation of candidates, which can be used to make different final predictions. Additionally, some aspects can use multiple different templates such as in the generation of candidate models. Additional embodiments can use coarser-grained and finer-grained models as input and/or output.

[00110] Various embodiments do not incorporate any assumptions about what features of a model are relevant to assessing its accuracy. It should be noted that embodiments are not restricted and several embodiments are applicable to any type of molecular system.

[00111] In some aspects, the initial layers of networks of various embodiments can be designed to recognize certain patterns, whose identities are learned during the training process rather than specified in advance. In such aspects, each of these layers can compute several features for each pattern based on the arrangement of relationship and the features computed by the previous layer. In certain aspects, the first layer's only inputs are the three-dimensional coordinates and chemical element type of each atom. Such a strategy allows various embodiments to predict a global property (e.g., accuracy of the structural model) while capturing local structural motifs and interatomic interactions in detail.

[00112] In some aspects, the architecture of these initial network layers can recognize that instances of a given structural motif are typically oriented and positioned differently from one

another, and that coarser-scale motifs (e.g., helices) often comprise particular arrangements of finer-scale motifs (e.g., base pairs). In some aspects, each layer can be rotationally and translationally equivariant-that is, rotation or translation of its input leads to a corresponding transformation of its output. Equivariance can capture the invariance of physics to rotation or translation of the frame of reference but ensures that orientation and position of an identified motif (or structure) are passed on to the network's next layer, which can use this information to recognize coarser-scale motifs. Equivariance can allow a single filter to learn to recognize a pattern in any orientation (as the rotated pattern corresponds to multiplying the output of the filter by a square matrix), and then for those patterns to be themselves combined together in rotation-independent ways, while still being able to reason about the rotation of the subunits.

[00113] The design of these initial layers builds on recently developed machine learning techniques that capture rotational as well as translational symmetries, particularly Tensor Field Networks. (See.g., D. E. Worrall, S. J. Garbin, D. Turmukhambetov, G. J. Brostow, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2017), ρρ. 7168-7177; B. Anderson, T. Hy, R. Kondor, Advances in Neural Information Processing Systems (2019), ρρ. 14537-14546; M. Weiler, M. Geiger, M. Welling, W. Boomsma, T. Cohen, Advances in Neural Information Processing Systems (2018), ρρ. 10381-10392; N. Thomas et al., arXiv 1802.08219 [cs.LG] (2018); and S. Eismann et al., Proteins. 89, 493-501 (2020); the disclosures of which are hereby incorporated by reference in their entireties.) In some aspects, one of the primary equivariant layers is the equivariant convolution.

Model Training

[00114] To train various embodiments, a library of TCRs and their pMHC partners can be obtained. For example, VDJDB is a public database with tens of thousands of TCR and antigen pairs, which can be used as a training set. As discussed above, training data sets can be augmented to include positive binding examples in which certain information is missing. This trains the model to handle uncertainty and missing data, producing a trained model that can accurately work with new TCR or antigen sequences.

[00115] Various embodiments can utilize a sampling method, such as the Rosetta FARFAR2 sampling method, without making any use of the known structure. (See e.g., A. M. Watkins, R. Rangan, R. Das, Structure. 28, 963-976.66 (2020); the disclosure of which is hereby incorporated by reference in its entirety.) Additional embodiments can utilize other sampling methods, such as Monte Carlo sampling. Further embodiments can then optimize the parameters of the model (e.g., neural network) such that its output matches as closely as possible the RMSD of each predicted structure from the corresponding experimentally derived structure.

[00116] After generating the training data, the computing system may train one or more machine learning models based on the training data. Such training may configure the machine learning models to group molecules into clusters based on chemical properties of the molecules. For example, the machine learning models may group the pharmaceutical targets into multiple clusters such that members of each cluster have similar chemical properties. In some implementations, the machine learning models may be configured to perform sparse subspace clustering to group the molecules into clusters. After training the machine learning models, the computing device may use the machine learning models to assign a candidate TCR to one of the clusters based on chemical properties of the candidate TCR. Members (e.g., multiple pMHCs) of the cluster to which the candidate TCR is assigned may be scored based on comparisons between the pMHCs and the candidate TCRs, and a subset of the pMHCs may be selected based on the scores. In some implementations, for each pMHC that is a member of the cluster, comparisons between molecular fingerprints of the pMHC and molecular fingerprints of the candidate TCR may be performed to generate multiple similarity scores, such as a Tanimoto coefficient, a cosine similarity, a largest common string (LCS) similarity, a Library for the Enumeration of Modular Natural Structures (LEMONS)-based similarity, a retrobiosynthesis and alignment (GRAPE) similarity, and the like, and a composite score for the pharmaceutical target can be generated by averaging the multiple similarity scores. In such implementations, pMHC having composite scores that satisfy one or more thresholds may be selected as the subset of pMHCs. In some implementations, the subset of pMHCs may be ranked using conjoint analysis and additional machine learning models to indicate the predicted commercial success, or another metric, of drugs formed by the candidate TCR and the respective pMHC.

[00117] The present disclosure describes systems that may train machine learning models to automatically identify a group of pMHC that are at least somewhat chemically or structurally related to a candidate TCRs (e.g., via clustering of pMHC based on chemical properties or structural properties). In some implementations, the clustering may include density-based spatial clustering of applications with noise (DBSCAN), K-means clustering, K-means for largescale clustering (LSC-K), longest common subsequence (LCS) clustering, longest common cyclic subsequence (LCCS) clustering, or the like. Identifying a group of pMHC that are more likely to bind to the candidate TCRs or compound from a much larger group of pMHC limits the search space for screening the candidate TCR, which may enable scoring, filtering, and identification of a selected subset of pMHC faster, and using less processing resources, than systems that implement “brute force” techniques to analyze and score an entirety of the larger group of pMHC. Additionally or alternatively, the machine learning models may be trained using a large

amount of binding data as training data, which enables the machine learning models to learn underlying relationships between chemical or physical properties of pMHC that would not be apparent to human drug experts due to their limited experience and knowledge, or their focus on finding a relationship with a particular pMHC for treating a particular disease or condition. Automatically ranking the selected subset of pMHC may also reduce the cycle time and cost associated with selecting the most commercially successful drugs to test. By screening candidate TCRs against pMHC in a shorter period of time and using fewer resources, the systems and methods described herein may substantially reduce the costs and shorten the development cycle associated with discovering and launching new drugs (e.g., pharmaceuticals) or discovering new pharmaceutical uses for existing TCRs.

[00118] Methods and systems disclosed herein can assess the ability of models to identify accurate structural models of previously unseen TCRs or pMHCs. In doing so, the methods and systems can utilize a benchmark set of training data for which experimentally determined structures have been published, but are not used in the training set. Various embodiments utilize a set of structural models for each TCRs in the benchmark set (e.g., 100 structural models, 250 structural models, 500 structural models, 1000 structural models, 1,500 structural models, or more).

Polynucleotides and Expression Vectors

[0102] In another aspect, polynucleotides can encode any of the candidate TCR polypeptides described herein. The polynucleotides may be operatively linked to one or more control sequences that control gene expression to create a recombinant polynucleotide capable of expressing the polypeptide. Expression constructs containing a heterologous polynucleotide encoding the candidate TCR can be introduced into appropriate host cells to express the corresponding TCR polypeptide.

[0103] Accordingly, in some embodiments, the polynucleotide encodes candidate TCR polypeptide having at least 60%, 65%, 70%, 75%, 80%, 85%, 90%, 91%, 92%, 93%, 94%, 95%, 96%, 97%, 98%, 99% or more sequence identity to a reference amino acid sequence identified by the methods and systems herein.

[0104] The polynucleotides encoding the candidate TCRs can be capable of hybridizing under highly stringent conditions to a reference polynucleotide sequence that encodes the candidate TCR, or a complement thereof.

[0105] In some embodiments, the polynucleotides are codon optimized to fit the host cell in which the protein is being produced. For example, preferred codons used in bacteria are used to express the gene in bacteria; preferred codons used in yeast are used for expression in yeast; and

preferred codons used in mammals are used for expression in mammalian cells. In some embodiments, all codons need not be replaced to optimize the codon usage since the natural sequence will comprise preferred codons and because use of preferred codons may not be required for all amino acid residues.

[0106] In another aspect, the polynucleotide encoding a candidate TCR polypeptide may be manipulated in a variety of ways to provide for expression of the polypeptide. The polynucleotides encoding the polypeptides can be provided as expression vectors where one or more control sequences are present to regulate the expression of the polynucleotides and/or polypeptides. Manipulation of the isolated polynucleotide prior to its insertion into a vector may be desirable or necessary depending on the expression vector. The techniques for modifying polynucleotides utilizing recombinant DNA methods are well known in the art. Guidance is provided in Sambrook et al., Molecular Cloning: A Laboratory Manual, 3rd Ed., Cold Spring Harbor Laboratory Press (2001); and Current Protocols in Molecular Biology, Ausubel. F. ed., Greene Pub. Associates, (1998), with updates to 2006.

[0107] In some embodiments, the control sequences include among others, promoters, enhancers, leader sequences, polyadenylation sequences, propeptide sequences, signal peptide sequences, and transcription terminators. Other control sequences will be apparent to the person of skill in the art.

[0108] Suitable promoters can be selected based on the host cells used. For bacterial host cells, suitable promoters for directing transcription of the nucleic acid constructs of the present disclosure, include the promoters obtained from the *E. coli* lac operon, *Streptomyces coelicolor* agarase gene (dagA), *Bacillus subtilis* levansucrase gene (sacB), *Bacillus licheniformis* alpha-amylase gene (amyL), *Bacillus stearothermophilus* maltogenic amylase gene (amyM), *Bacillus amyloliquefaciens* alpha-amylase gene (amyQ), *Bacillus licheniformis* penicillinase gene (penP), *Bacillus subtilis* xylA and xylB genes, and prokaryotic beta-lactamase gene, the *tac* promoter, or the T7 promoter.

[0109] Exemplary promoters for filamentous fungal host cells, include promoters obtained from the genes for *Aspergillus oryzae* TAKA amylase, *Rhizomucor miehei* aspartic proteinase, *Aspergillus niger* neutral alpha-amylase, *Aspergillus niger* acid stable alpha-amylase, *Aspergillus niger* or *Aspergillus awamori* glucoamylase (glaA), *Rhizomucor miehei* lipase, *Aspergillus oryzae* alkaline protease, *Aspergillus oryzae* triose phosphate isomerase, *Aspergillus nidulans* acetamidase, and *Fusarium oxysporum* trypsin-like protease (WO 96/00787), as well as the NA2-tpi promoter (a hybrid of the promoters from the genes for *Aspergillus niger* neutral alpha-amylase and *Aspergillus oryzae* triose phosphate isomerase), and mutant, truncated, and

hybrid promoters thereof. Exemplary yeast cell promoters can be from the genes can be from the genes for *Saccharomyces cerevisiae* enolase (ENO-1), Saccharomyces *cerevisiae* galactokinase (GAL1), *Saccharomyces cerevisiae* alcohol dehydrogenase/glyceraldehyde-3-phosphate dehydrogenase (ADH2/GAP), and *Saccharomyces cerevisiae* 3-phosphoglycerate kinase.

[0110] Exemplary promoters for insect cells include, among others, those based on polyhedron, PCNA, OplE2, OplE1, Drosophila metallothionein, and Drosophila actin 5C. In some embodiments, insect cell promoters can be used with Baculoviral vectors.

[0111] Exemplary promoters for plant cells include, among others, those based on cauliflower mosaic virus (CaMV) 35S, polyubiquitin gene (PvUbi1 and PvUbi2), rice (*Oryza sativa*) actin 1 (OsAct1) and actin 2 (OsAct2) promoters, the maize ubiquitin 1 (ZmUbi1) promoter, and multiple rice ubiquitin (RUBQ1, RUBQ2, rubi3) promoters.

[0112] Exemplary promoters for mammalian cells include, among others, CMV IE promoter, elongation factor 1α-subunit promoter, ubiquitin C promoter, Simian Virus 40 promoter, and phosphoglycerate Kinase-1 promoter.

[0113] The control sequence may also be a suitable leader sequence, a nontranslated region of an mRNA that is important for translation by the host cell. The leader sequence is operably linked to the 5' terminus of the nucleic acid sequence encoding the polypeptide. Any leader sequence that is functional in the host cell of choice may be used.

[0114] The control sequence may also be a polyadenylation sequence, a sequence operably linked to the 3' terminus of the nucleic acid sequence and which, when transcribed, is recognized by the host cell as a signal to add polyadenosine residues to transcribed mRNA. Any polyadenylation sequence which is functional in the host cell of choice may be used in the present invention.

[0115] The control sequence may also be a signal peptide coding region that codes for an amino acid sequence linked to the amino terminus of a polypeptide and directs the encoded polypeptide into the cell's secretory pathway. The 5' end of the coding sequence of the nucleic acid sequence may inherently contain a signal peptide coding region naturally linked in translation reading frame with the segment of the coding region that encodes the secreted polypeptide. Alternatively, the 5' end of the coding sequence may contain a signal peptide coding region that is foreign to the coding sequence. Any signal peptide coding region which directs the expressed polypeptide into the secretory pathway of a host cell of choice may be used in the present disclosure.

[0116] The control sequence may also be a propeptide coding region that codes for an amino acid sequence positioned at the amino terminus of a polypeptide. The resultant polypeptide is

known as a proenzyme or propolypeptide (or a zymogen in some cases). A propolypeptide can be converted to a mature active polypeptide by catalytic or autocatalytic cleavage of the propeptide from the propolypeptide. Where both signal peptide and propeptide regions are present at the amino terminus of a polypeptide, the propeptide region is positioned next to the amino terminus of a polypeptide and the signal peptide region is positioned next to the amino terminus of the propeptide region.

[0117] The control sequence may also be a suitable transcription terminator sequence, a sequence recognized by a host cell to terminate transcription. The terminator sequence is operably linked to the 3' terminus of the nucleic acid sequence encoding the polypeptide. Any terminator which is functional in the host cell of choice may be used.

[0118] It may also be desirable to add regulatory sequences, which allow the regulation of the expression of the polypeptide relative to the growth of the host cell. Examples of regulatory systems are those which cause the expression of the gene to be turned on or off in response to a chemical or physical stimulus, including the presence of a regulatory compound. In prokaryotic host cells, suitable regulatory sequences include the lac, tac, and trp operator systems. In yeast host cells, suitable regulatory systems include, as examples, the ADH2 system or GAL1 system. In filamentous fungi, suitable regulatory sequences include the TAKA alpha-amylase promoter, *Aspergillus niger* glucoamylase promoter, and *Aspergillus oryzae* glucoamylase promoter.

[0119] In another aspect, the present disclosure is also directed to a recombinant expression vector comprising a polynucleotide encoding a candidate TCR polypeptide, and one or more expression regulating regions such as a promoter and a terminator, a replication origin, etc., depending on the type of hosts into which they are to be introduced.

[0120] The expression vector may be an autonomously replicating vector, *i.e.*, a vector that exists as an extrachromosomal entity, the replication of which is independent of chromosomal replication, *e.g.*, a plasmid, an extrachromosomal element, a minichromosome, or an artificial chromosome. The vector may contain any means for assuring self-replication. Alternatively, the vector may be one which, when introduced into the host cell, is integrated into the genome and replicated together with the chromosome(s) into which it has been integrated. Furthermore, a single vector or plasmid or two or more vectors or plasmids which together contain the total DNA to be introduced into the genome of the host cell, or a transposon may be used. The expression vector can exist as a single copy in the host cell, or maintained at higher copy numbers, *e.g.*, up to 4 for low copy number and 50 or more for high copy number.

[0121] In some embodiments, the expression vector contains one or more selectable markers, which permit selection of transformed cells. A selectable marker is a gene the product of which

provides for biocide or viral resistance, resistance to heavy metals, prototrophy to auxotrophs, and the like. Examples of bacterial selectable markers are the *dal* genes from *Bacillus subtilis* or *Bacillus licheniformis*, or markers, which confer antibiotic resistance such as ampicillin, kanamycin, chloramphenicol (Example 1) or tetracycline resistance. Suitable markers for yeast host cells are ADE2, HIS3, LEU2, LYS2, MET3, TRP1, and URA3. Selectable markers for use in a filamentous fungal host cell include, but are not limited to, amdS (acetamidase), argB (ornithine carbamoyltransferase), bar (phosphinothricin acetyltransferase), hph (hygromycin phosphotransferase), niaD (nitrate reductase), pyrG (orotidine-5'-phosphate decarboxylase), sC (sulfate adenyltransferase), and trpC (anthranilate synthase), as well as equivalents thereof. Embodiments for use in an Aspergillus cell include the amdS and pyrG genes of *Aspergillus nidulans* or *Aspergillus oryzae* and the bar gene of *Streptomyces hygroscopicus*.

Host Cells

[0122] In another aspect, the present disclosure provides a host cell comprising a polynucleotide encoding a candidate TCR polypeptide of the present disclosure. Host cells can be prokaryotic or eukaryotic cells. Prokaryotic host cells include eubacteria such as, for example, Bacillus, such as B. lichenformis or B. subtilis; Pantoea, such as P. citrea; Pseudomonas, such as P. alcaligenes; Streptomyces, such as S. lividans or S. rubiginosus; Escherichia, such as E. coli; Enterobacter; Streptococcus; Archaea, such as Methanosarcina mazei; or Corynebacterium, such as C. glutamicum. The host cell can be a gram-positive bacterium such as, for example, strains of Streptomyces (*e.g.*, s. lividans, S. coelicolor, or S. griseus) and Bacillus. The host cell can also be a gram-negative bacterium, such as, for example, E. coli or Pseudomonas sp.

[0123] Eukaryotic host cells can include, for example, fungi, algal, plant, or mammalian cells. Fungal host cells include, for example, Saccharomyces cerevisiae, Schizosaccharomyces pombe, Kluyveromyces lactis, Kluyveromyces marxianus, Aspergillus terreus, Aspergillus niger, Pichia pastoris, Rhizopus arrhizus, Rhizobus oryzae, Yarrowia lipolytica, and the like.

[0124]] Examples of algal host cells include, for example, green algae, red algae, glaucophytes, chlorarachniophytes, euglenids, chromista, dinoflagellates, Chlorella, Chlamydomonas, Scenedesmus, Isochrysis, Dunaliella, Tetraselmis, Nannochloropsis, or Prototheca.

[0125] Plant host cells include, for example, cells of monocotyledonous or dicotyledonous plants including, but not limited to, maize, wheat, barley, rye, oat, rice, soybean, peanut, pea, lentil and alfalfa, cotton, rapeseed, canola, pepper, sunflower, potato, tobacco, tomato, eggplant, eucalyptus, a tree, an ornamental plant, a perennial grass, or a forage crop. In other embodiments the host cells are algal, including but not limited to algae of the genera,

[0126] Polynucleotides for expression of the candidate TCR may be introduced into cells by various methods known in the art. Techniques include among others, electroporation, biolistic particle bombardment, liposome mediated transfection, calcium chloride transfection, microinjection, recombinant viral transfection, and protoplast fusion. The introduced nucleic acids may be integrated into chromosomal DNA or maintained as extrachromosomal replicating sequences. General transformation techniques are known in the art (see, *e.g.*, Current Protocols in Molecular Biology, F. M. Ausubel et al. eds, Chapter 9 (1987); Sambrook et al., Molecular Cloning: A Laboratory Manual, 3rd Ed., Cold Spring Harbor Laboratory Press, N.Y. (2001); and Campbell et al., *Curr Genet.* 16:53-56, 1989; each publication incorporated herein by reference).

[0127] Various features and embodiments of the disclosure are illustrated in the following representative examples, which are intended to be illustrative, and not limiting. However, one skilled in the art will readily appreciate that the specific methods and results discussed are merely illustrative of the inventions as described more fully in the claims which follow thereafter. Unless otherwise indicated, the disclosure is not limited to specific procedures, materials, or the like, as such may vary. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting.

EXAMPLES

Example 1: TAPIR – Method for Finding TCR-Antigen Interactions

[0128] TAPIR (T-cell receptor and Peptide Interaction Recognizer) is a deep neural network (e.g., a convolutional neural network) architecture that leverages convolutional neural network based encoders to process TCR and target sequences across a variety of representations (e.g., a, B, or paired aB-chain TCRs; MHC alleles or pMHC complexes) and learns to predict interactivity between TCR and pMHC sequences. The resulting TAPIR models can make predictions over any combination of V gene, J gene, CDR3 gene, and MHC allele and target sequences. This flexibility allows TAPIR to learn from larger and more diverse datasets than prior work, including a dataset of 23,353 paired and 67,578 unpaired TCRs against 1063 targets from VDJDB, as well as a proprietary dataset of 2008 paired TCRs against an additional 706 targets.

[0129] TAPIR is robust and versatile, and produced novel results when presented with TCRs or pMHCs that were different from the training data. Validation of some of these TCR-pMHC pairs showed that TAPIR could accurately find novel TCRs sequences that were different from the training data for a pMHC target.

Example 2: Comparison of TAPIR to Published Models

[0130] TAPIR was compared to three previously published models from the literature: TCRAI, DeepTCR, and NetTCR. These prior models only reported results for common antigen targets, so all models were benchmarked on the 14 most common antigens in VDJB. All models were retrained on the same snapshot of VDJDB.

[0131] Overall, the area under the curve (AUC) metric indicates that TAPIR, DeepTCR, and TCRAI all offer strong performance on common targets, with TAPIR reporting higher and more consistent scores. Notably, NetTCR, like TAPIR, is a general model that can take any antigen target as an input, whereas DeepTCR and TCRAI adopt categorical classes for each antigen and cannot be applied to targets outside of the training set.

[0132] TAPIR and NetTCR were then challenged with test data that that does not appear within >2 Levenshtein distance from any target in the training set, leaving 46 novel targets. A second set of test data was made from a smaller group of 10 targets with >0 evidence score in VDJDB. TAPIR showed stronger performance on both on the full set of novel targets (0.73 vs 0.65 AUC) and also on the smaller set of targets with non-zero evidence scores (0.78 vs 0.68). TAPIR significantly outperforms NetTCR ($p < 0.001$) despite training on the same data. DeepTCR and TCRAI adopt categorical classes for each antigen and so cannot be applied to targets outside of the training set and were not usable in this challenge.

Example 3: TAPIR Discovered a Novel TCR for PIK3CA

[0133] PIK3CA is a gene frequently mutated in various cancers. PIK3CA encodes the catalytic subunit of the phosphatidylinositol 3-kinase enzyme, involved in cell signaling pathways. PIK3CA H1047L mutation is a well characterized cancer driver mutation and is present in 10-18% of breast cancer patients, and its peptide fragment ALHGGWTTK (SEQ ID NO: 1) is shown to present well by HLA-A*03:01 allele.

[0134] A population of TCRs sequenced from three cancer patients, and these clones enriched for binders to ALHGGWTTK (SEQ ID NO: 1). This population of TCRs were sequenced and TAPIR was used to rank the 67 TCRs with more than 10 clones in the population. The TCRs were ranked by raw score, the direct output score of the TAPIR model, and antigen percentile, a measure of how a TCR's score against ALHGGWTTK compares to scores for 200 other common, rare, and novel antigens.

[0135] A novel T-cell Receptor was found in the top 3 ranked TCRs that binds to the mutated PIK3CA antigen ALHGGWTTK (SEQ ID NO: 1), and the activity of this TCT was biologically validated.

[0136] All publications, patents, patent applications and other documents cited in this application are hereby incorporated by reference in their entireties for all purposes to the same

extent as if each individual publication, patent, patent application or other document were individually indicated to be incorporated by reference for all purposes.

[0137] While various specific embodiments have been illustrated and described, it will be appreciated that various changes can be made without departing from the scope of the invention(s) of the disclosure.

What is claimed is:

1. A method, comprising, embedding an amino acid sequence of a T-cell receptor using a first convolutional encoder; embedding an amino acid sequence of an antigen and an MHC using a second convolutional encoder; combining the embedded amino acid sequence of the T-cell receptor with the embedded amino acid sequence of the antigen and the MHC by concatenation to make a concatenated data; inputting the concatenated data to a first trained prediction model; and obtaining an interaction score from the prediction model.

2. The method of claim 1, further comprising the steps of inputting the concatenated data into a plurality of other trained prediction models, wherein the first trained prediction model and the plurality of other prediction models use a common prediction model and are trained with the same data, but each model represents a different training epoch and an order of a plurality of training data is different for each predictive model; and making an ensemble of the outputs from each predictive model to produce an ensemble interaction score.

3. The method of claim 1, wherein the interaction score is used to rank a plurality of TCRs.

4. The method of claim 1, wherein the prediction model is a convolutional neural network.

5. The method of claim 4, wherein the convolutional neural network has 3 layers.

6. The method of claim 5, wherein the prediction model is trained on a set of data that has been augmented to make an augmented set of data.

7. The method of claim 6, wherein the augmented set of data includes pairs of T-cell receptors and antigen-MHC that are missing some sequence information for the embedded amino acid sequence of the T-cell receptor.

8. The method of claim 6, wherein the augmented set of data includes pairs of T-cell receptors and antigen-MHC that have mismatched to make a nonbinding pair.

9. A system, comprising a memory including executable instructions; and a processor configured to execute the executable instructions and cause the system to embed an amino acid sequence of a T-cell receptor; a processor configured to execute the executable instructions and cause the system to embed an amino acid sequence of an antigen and an MHC using a second convolutional encoder; a processor configured to execute the executable instructions and cause the system to embed combine the embedded amino acid sequence of the T-cell receptor with the embedded amino acid sequence of the antigen and the MHC by

concatenation to make a concatenated data; and a processor configured to execute the executable instructions to make a first trained prediction model that produces an interaction score.

10. The system of claim 9, further comprising a plurality of processors configured to execute the executable instructions to make a plurality of other trained prediction models.

11. The system of claim 10, further comprising a processor configured to execute the executable instructions to make an ensemble of the outputs from each predictive model to produce an ensemble interaction score.

12. The system of claim 9, further comprising a plurality of processors configured to execute the executable instructions to perform an additional data analysis on the interaction score.

13. The system of claim 9, wherein the prediction model is a convolutional neural network.

14. The system of claim 13, wherein the convolutional neural network has 3 layers.

15. The system of claim 9, wherein the prediction model is trained on a set of data that has been augmented to make an augmented set of data.

16. The system of claim 15, wherein the augmented set of data includes pairs of T-cell receptors and antigen-MHC that are missing some sequence information for the embedded amino acid sequence of the T-cell receptor.

17. The system of claim 15, wherein the augmented set of data includes pairs of T-cell receptors and antigen-MHC that have mismatched to make a nonbinding pair.

18. A computer readable storage media with a plurality of instructions, comprising, embedding an amino acid sequence of a T-cell receptor using a first convolutional encoder; embedding an amino acid sequence of an antigen and an MHC using a second convolutional encoder; combining the embedded amino acid sequence of the T-cell receptor with the embedded amino acid sequence of the antigen and the MHC by concatenation to make a concatenated data; inputting the concatenated data to a first trained prediction model; and obtaining an interaction score from the prediction model.

19. The computer readable storage media of claim 18, further comprising instructions for the steps of inputting the concatenated data into a plurality of other trained prediction models, wherein the first trained prediction model and the plurality of other prediction models use a common prediction model and are trained with the same data, but each model represents a different training epoch and an order of a plurality of training data is

different for each predictive model; and making an ensemble of the outputs from each predictive model to produce an ensemble interaction score.

20. The computer readable storage media of claim 19, further comprising instructions for the step of subjecting the interaction score to an additional data analysis.

21. The computer readable storage media of claim 19, further comprising instructions for the step of ranking a plurality of TCRs using the interaction score.

22. The computer readable storage media of claim 19, wherein the prediction model is a convolutional neural network.

23. The computer readable storage media of claim 22, wherein the convolutional neural network has 3 layers.

24. The computer readable storage media of claim 19, further comprising a set of data that has been augmented to make an augmented set of data for training the prediction model.

25. The computer readable storage media of claim 24, wherein the augmented set of data includes pairs of T-cell receptors and antigen-MHC that are missing some sequence information for the embedded amino acid sequence of the T-cell receptor.

26. The computer readable storage media of claim 24, wherein the augmented set of data includes pairs of T-cell receptors and antigen-MHC that have mismatched to make a nonbinding pair.

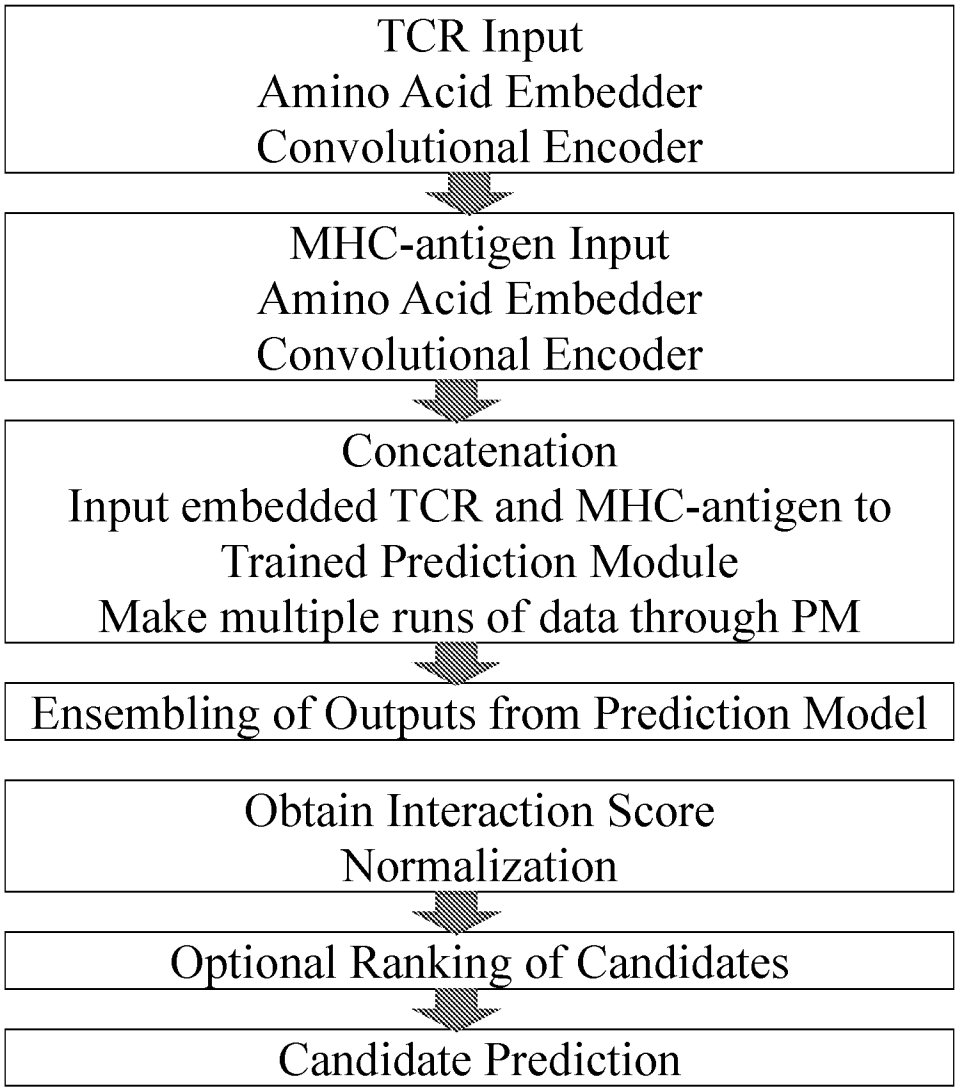
TCR Input
Amino Acid Embedder
Convolutional Encoder
MHC-antigen Input
Amino Acid Embedder
Convolutional Encoder
Concatenation
Input embedded TCR and MHC-antigen to
Trained Prediction Module
Make multiple runs of data through PM
Ensembling of Outputs from Prediction Model
Obtain Interaction Score
Normalization
Optional Ranking of Candidates
Candidate Prediction

FIG. 1

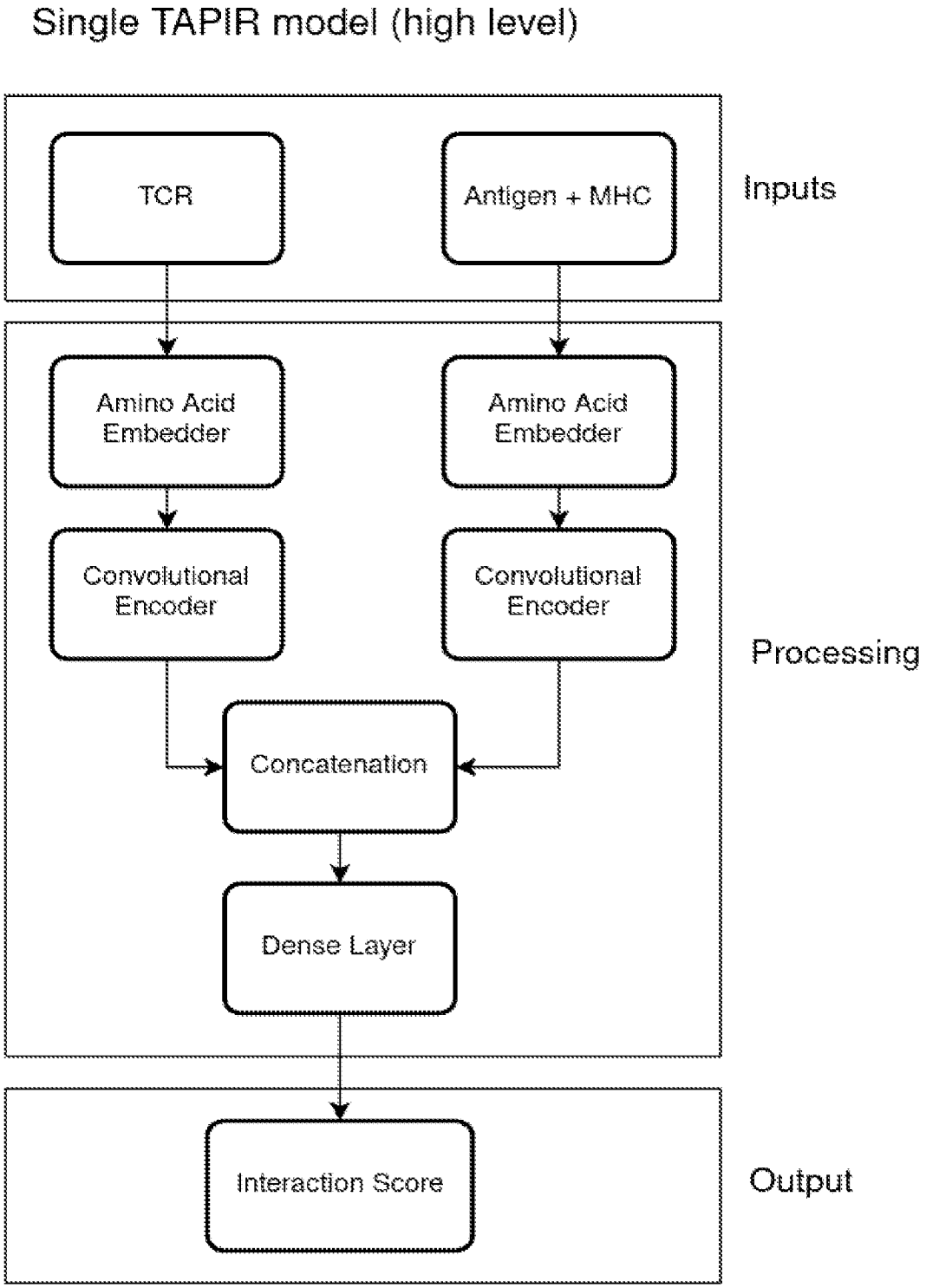
Single TAPIR model (high level)
TCR
Antigen + MHC
Inputs
Amino Acid Embedder
Amino Acid Embedder
Convolutional Encoder
Convolutional Encoder
Processing
Concatenation
Dense Layer
Interaction Score
Output

FIG. 2

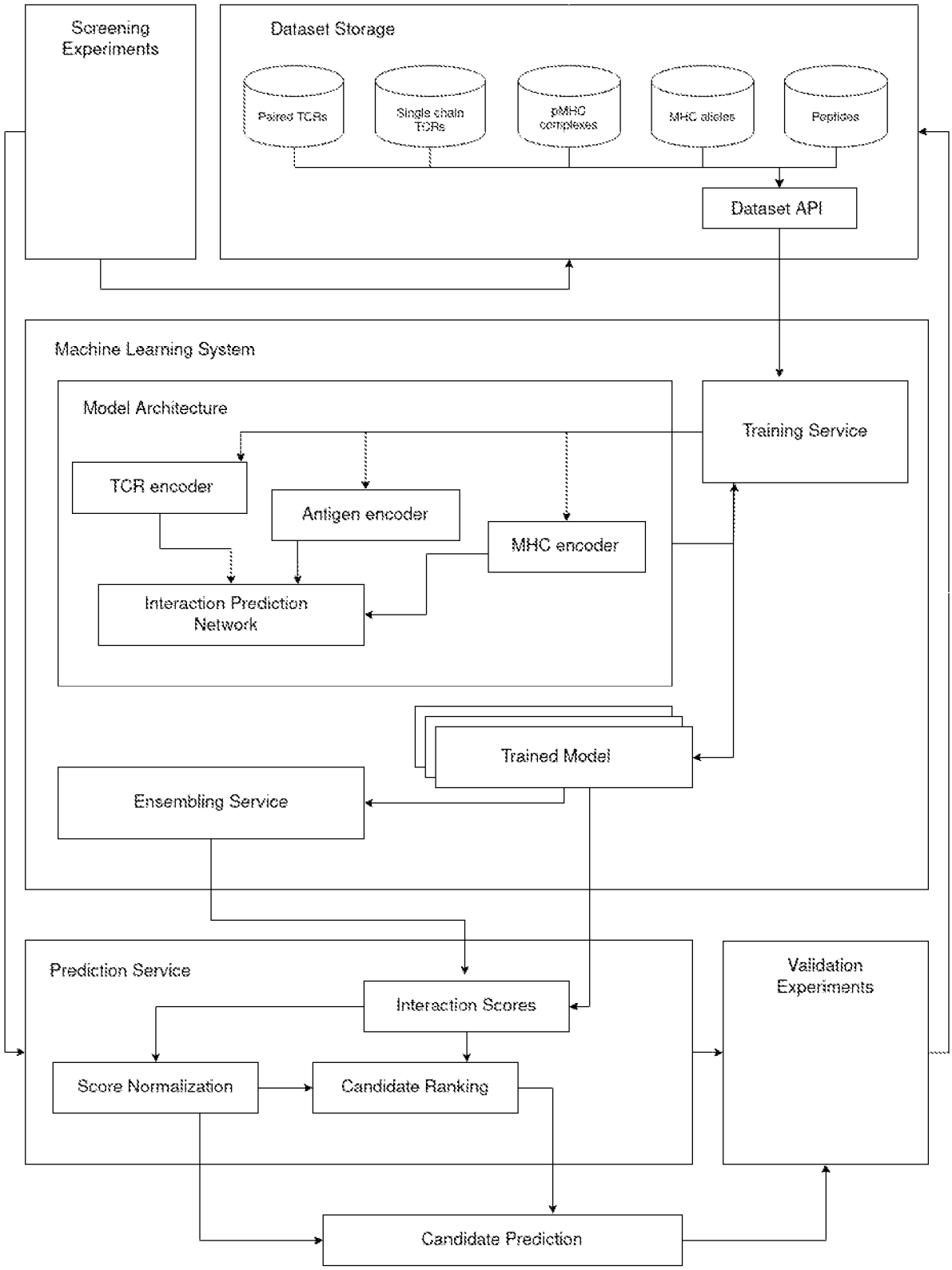
Screening Experiments
Dataset Storage
Paired TCRs
Single chain TCRs
pMHC complexes
MHC alleles
Peptides
Dataset API
Machine Learning System
Model Architecture
Training Service
TCR encoder
Antigen encoder
MHC encoder
Interaction Prediction Network
Trained Model
Ensembling Service
Prediction Service
Interaction Scores
Score Normalization
Candidate Ranking
Validation Experiments
Candidate Prediction

FIG. 3

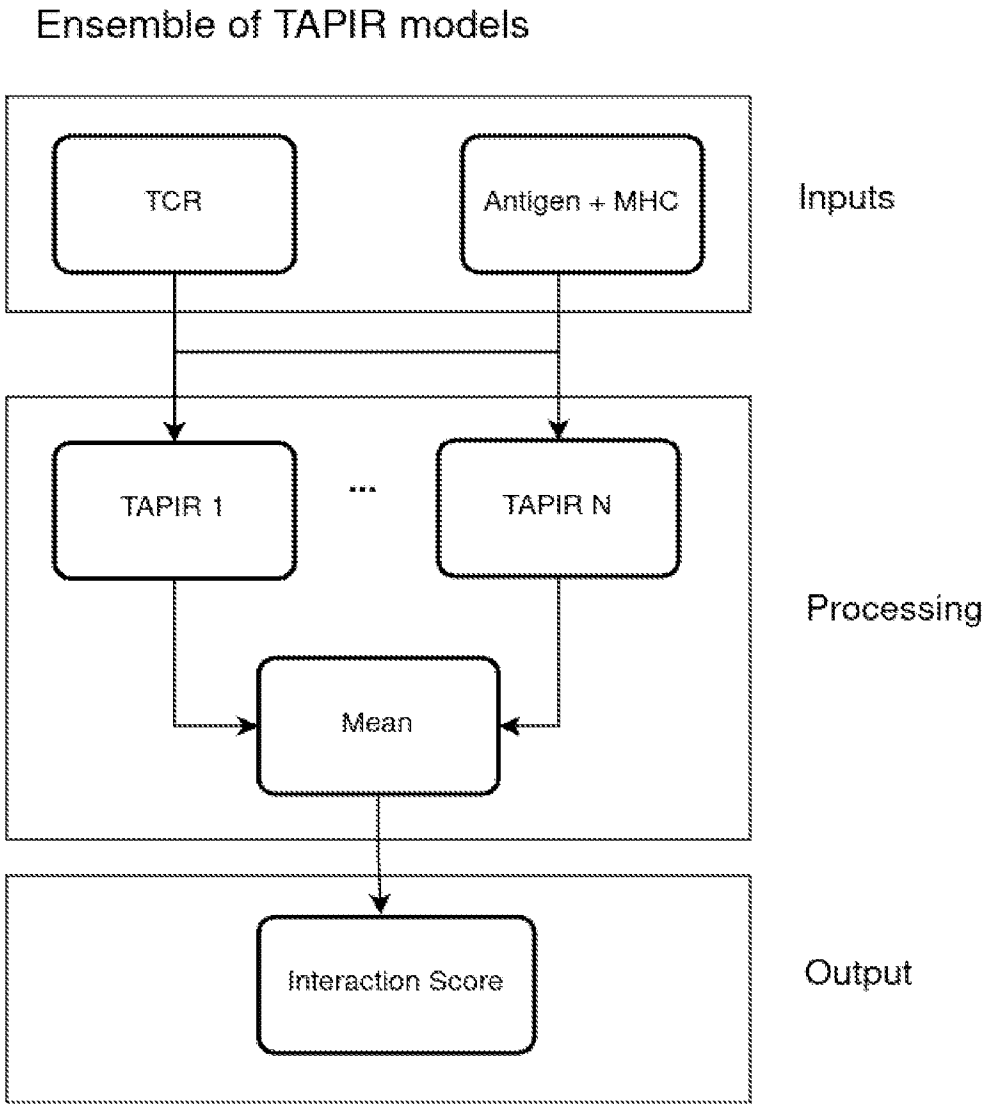
Ensemble of TAPIR models
TCR
Antigen + MHC
Inputs
TAPIR 1
...
TAPIR N
Processing
Mean
Interaction Score
Output

FIG. 4

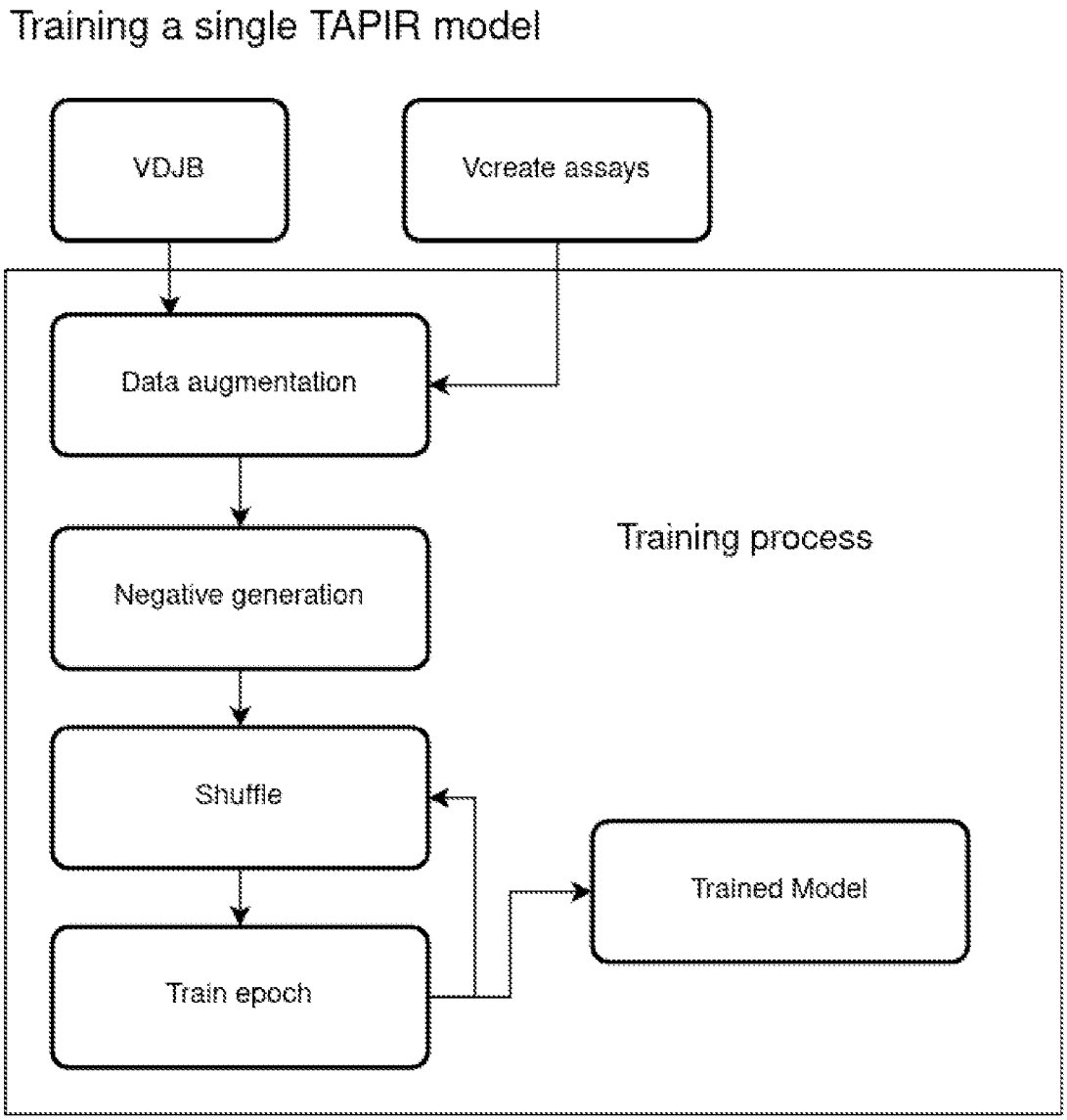
Training a single TAPIR model
VDJB
Vcreate assays
Data augmentation
Training process
Negative generation
Shuffle
Trained Model
Train epoch

FIG. 5

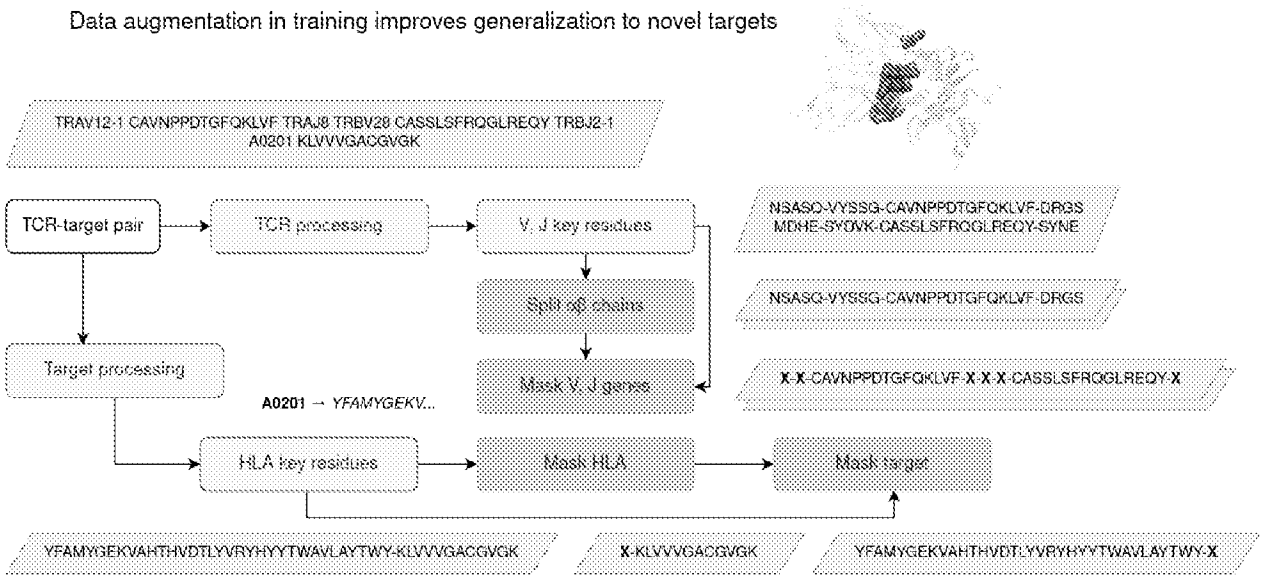
Data augmentation in training improves generalization to novel targets
TRAV12-1 CAVNPPDTGFQKLVF TRAJ8 TRBV28 CASSLSFRQGLREQY TRBJ2-1
A0201 KLVVVGACGVGK
TCR-target pair
TCR processing
V, J key residues
Split αβ chains
Mask V, J genes
Target processing
A0201 → YFAMYGEKV...
HLA key residues
Mask HLA
Mask target
NSASQ-VYSSG-CAVNPPDTGFQKLVF-DRGS
MDHE-SYQVK-CASSLSFRQGLREQY-SYNE
NSASQ-VYSSG-CAVNPPDTGFQKLVF-DRGS
X-X-CAVNPPDTGFQKLVF-X-X-X-CASSLSFRQGLREQY-X
YFAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY-KLVVVGACGVGK
X-KLVVVGACGVGK
YFAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY-X

FIG. 6

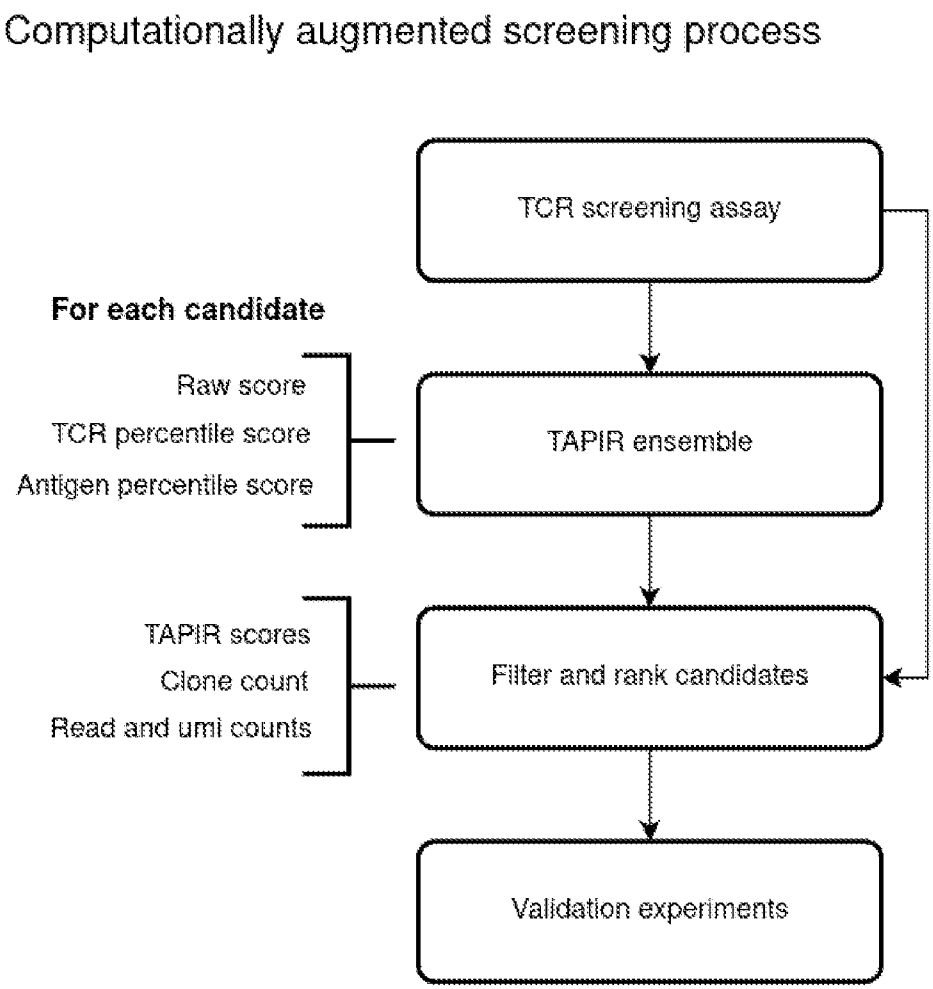
Computationally augmented screening process
TCR screening assay
For each candidate
Raw score
TCR percentile score
Antigen percentile score
TAPIR ensemble
TAPIR scores
Clone count
Read and umi counts
Filter and rank candidates
Validation experiments

FIG. 7

E = embedding dimension
D1, D2, D3 = convolution output channel dimensions

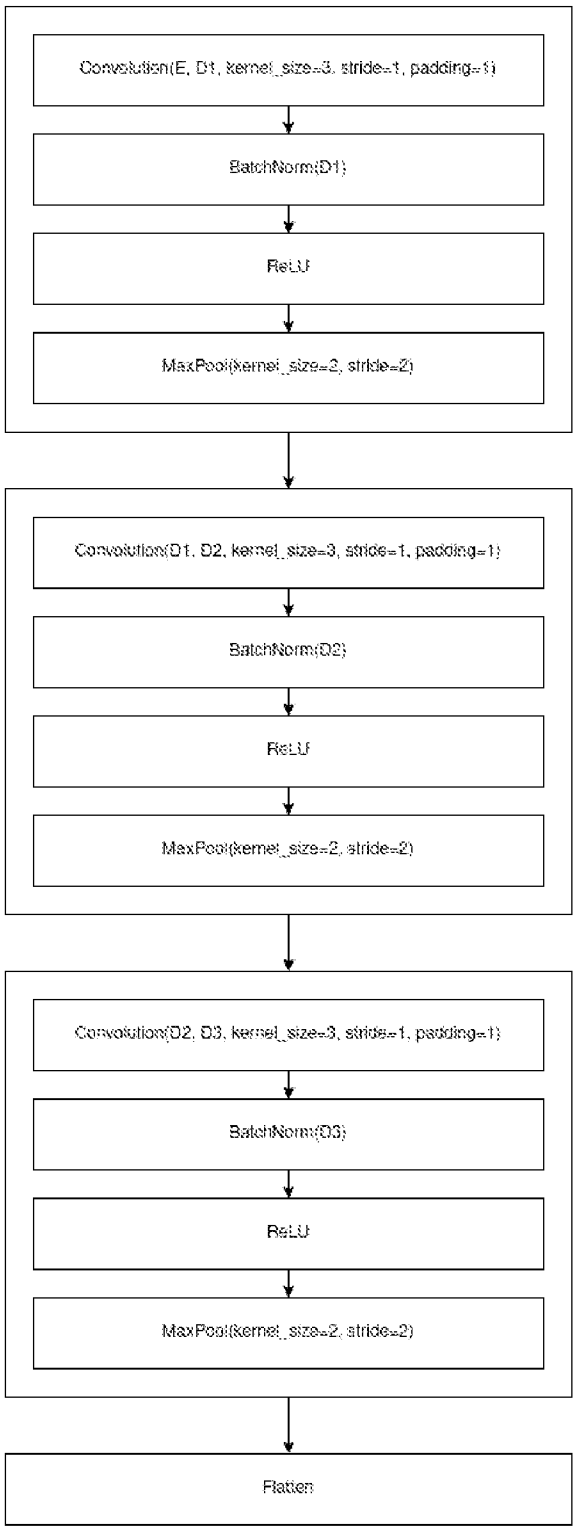

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2024/040882

A. CLASSIFICATION OF SUBJECT MATTER

IPC: ***C07K 14/705*** (2024.01); ***G06N 3/02*** (2024.01); ***G16B 30/10*** (2024.01); ***G16B 30/20*** (2024.01); *G16B 15/30* (2024.01); *G01N 33/68* (2024.01)

CPC: ***C07K 14/7051; G06N 3/02; G16B 30/10; G16B 30/20;*** *G01N 33/6818; G16B 15/30*

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

See Search History Document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

See Search History Document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History Document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	WO 2022/072722 A1 (THE BOARD OF REGENTS OF THE UNIVERSITY OF TEXAS SYSTEM) 07 April 2022 (07.04.2022)	
X	entire document	1, 3-10, 12-18
Y	entire document	2, 11, 19-26
	WO 2022/192699 A1 (THE BOARD OF REGENTS OF THE UNIVERSITY OF TEXAS SYSTEM) 15 September 2022 (15.09.2022)	
Y	entire document	2, 11, 19-26
	KR 2558549 B1 (NEOGENTC) 24 July 2023 (24.07.2023)	
A	see machine translation	1-26
	WO 2023/028595 A1 (THE REGENTS OF THE UNIVERSITY OF CALIFORNIA) 02 March 2023 (02.03.2023)	
A	entire document	1-26

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 September 2024 (17.09.2024)	**27 September 2024 (27.09.2024)**

Name and mailing address of the ISA/US	Authorized officer
Mail Stop PCT, Attn: ISA/US **Commissioner for Patents** **P.O. Box 1450, Alexandria, VA 22313-1450**	**MATOS** **TAINA**
Facsimile No. **571-273-8300**	Telephone No. **571-272-4300**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2024/040882

Box No. I Nucleotide and/or amino acid sequence(s) (Continuation of item 1.c of the first sheet)

1. With regard to any nucleotide and/or amino acid sequence disclosed in the international application, the international search was carried out on the basis of a sequence listing:
   a. ☑ forming part of the international application as filed.
   b. ☐ furnished subsequent to the international filing date for the purposes of international search (Rule 13*ter*.1(a)),
      ☐ accompanied by a statement to the effect that the sequence listing does not go beyond the disclosure in the international application as filed.

2. ☐ With regard to any nucleotide and/or amino acid sequence disclosed in the international application, this report has been established to the extent that a meaningful search could be carried out without a WIPO Standard ST.26 compliant sequence listing.

3. Additional comments: