OLGA: fast computation of generation probabilities of B- and T-cell receptor amino acid sequences and motifs

Zachary Sethna[1], Yuval Elhanati[1], Curtis G. Callan Jr.[1,2], Aleksandra M. Walczak[2*], Thierry Mora[3*]

[1] *Joseph Henry Laboratories, Princeton University,*
Princeton, New Jersey 08544 USA

[2] *Laboratoire de physique théorique,*
CNRS, Sorbonne Université,
and École Normale Supérieure (PSL University),
24, rue Lhomond, 75005 Paris, France

[3] *Laboratoire de physique statistique, CNRS,*
Sorbonne Université, Université Paris-Diderot,
and École Normale Supérieure (PSL University),
24, rue Lhomond, 75005 Paris, France

[*] *These authors contributed equally.*

Motivation: High-throughput sequencing of large immune repertoires has enabled the development of methods to predict the probability of generation by V(D)J recombination of T- and B-cell receptors of any specific nucleotide sequence. These generation probabilities are very non-homogeneous, ranging over 20 orders of magnitude in real repertoires. Since the function of a receptor really depends on its protein sequence, it is important to be able to predict this probability of generation at the amino acid level. However, brute-force summation over all the nucleotide sequences with the correct amino acid translation is computationally intractable. The purpose of this paper is to present a solution to this problem.

Results: We use dynamic programming to construct an efficient and flexible algorithm, called OLGA (Optimized Likelihood estimate of immunoGlobulin Amino-acid sequences), for calculating the probability of generating a given CDR3 amino acid sequence or motif, with or without V/J restriction, as a result of V(D)J recombination in B or T cells. We apply it to databases of epitope-specific T-cell receptors to evaluate the probability that a typical human subject will possess T cells responsive to specific disease-associated epitopes. The model prediction shows an excellent agreement with published data. We suggest that OLGA may be a useful tool to guide vaccine design.

Availability: Source code is available at `https://github.com/zsethna/OLGA`

I. INTRODUCTION

The ability of the adaptive immune system to recognize foreign peptides, while avoiding self peptides, depends crucially on the specificity of receptor-antigen binding and the diversity of the receptor repertoire. Immune repertoire sequencing (Repseq) of B- and T-cell receptors (BCR and TCR) [16, 21, 39, 46] offers an efficient experimental tool to probe the diversity of full repertoires in healthy individuals [11, 18, 26, 28, 32, 33, 45], in cohorts with specific conditions [4, 9, 10, 17, 19, 20, 29, 43] and evaluate the response to specific fluorescent MHC-multimers [2, 14]. Recent work has shown that responding clonotypes often form disjoint clusters of similar amino acid sequences, which has lead to the identification of responsive amino acid motifs [2, 14]. In order for these techniques to have practical applications in therapy and vaccine design, one needs a fast and efficient algorithm to evaluate which specific amino acid sequences and sequence motifs are likely to be generated and found in repertoires. We present a solution to this problem in the form of an algorithm and computational tool, called OLGA, which implements an exact computation of the generation probability of any BCR or TCR sequence (nucleotide or amino acid), or motif.

BCR and TCR are stochastically generated by choosing a germline genetic template in each of several cassettes of alternates (V, (D), or J) and then splicing them together with random nucleotide deletions and insertions at the junctions. Given a generative model, one can define the generation probability of any nucleotide sequence as the sum of the probabilities of all the generative events that can produce that sequence [6, 7, 25, 27]. However, computing the generation probability of amino acid sequences by summing over all consistent nucleotide sequences is impractical: because of codon degeneracy, the number of nucleotide sequences to be summed grows exponentially with sequence length. OLGA is powered by an efficient dynamic programming method to exactly sum over generative events and obtain net probabilities of amino acid sequences and motifs.

We validate our algorithm by comparing its results and performance to Monte-Carlo sampling estimates. We present results using publicly available data for both TCR α (TRA, Pogorelyy *et al.* [28]) and β (TRB, Robins *et al.* [33]) chains and BCR heavy chains (IGH, DeWitt *et al.* [3] of humans), and TRB of mice [35]. We applied OLGA to a TCR database that catalogs the different CDR3 amino acid sequences responding to a variety of different epitopes associated with disease [37]. We computed the generation probability of particular CDR3 amino acid sequences, as well as the net generation probability of all the TCR that respond to a particular epitope. Finally, we discuss OLGA's applications in vaccine design and other therapeutic contexts.

II. METHODS

A. Stochastic model of VDJ recombination

V(D)J recombination is a stochastic process involving several events (gene template selection, terminal deletions from the templates, random insertions at the junctions), each of which has a set of possible outcomes chosen according to a discrete probability distribution. The probability $P_{\text{gen}}^{\text{rec}}(E)$ of any generation event E, defined as a combination of the above-mentioned processes is, for the TRB locus:

$$\begin{aligned} P_{\text{gen}}^{\text{rec}}(E) &= P_{\text{V}}(V)P_{\text{DJ}}(D,J)P_{\text{delV}}(d_V|V)P_{\text{delJ}}(d_J|J) \\ &\times P_{\text{delD}}(d_D,d'_D|D)P_{\text{insVJ}}(\ell_{\text{VD}})p_0(m_1)\left[\prod_{i=2}^{\ell_{VD}} S_{\text{VD}}(m_i|m_{i-1})\right] \\ &\times P_{\text{insDJ}}(\ell_{\text{DJ}})q_0(n_{\ell_{DJ}})\left[\prod_{i=1}^{\ell_{DJ}-1} S_{\text{DJ}}(n_i|n_{i+1})\right], \end{aligned} \tag{1}$$

where (V,D,J) identify the choices of gene templates, (d_V,d_D,d'_D,d_J) are the numbers of deletions at each end of the segments, and $(m_1,\ldots,m_{\ell_{VD}})$ and $(n_1,\ldots,n_{\ell_{DJ}})$ are the untemplated inserted nucleotide sequences at the VD and DJ junctions. These variables specify the recombination event E, and are drawn according to the probability distributions (P_{V}, P_{DJ}, P_{delV}, P_{delD}, P_{delJ}, P_{insVJ}, P_{insDJ}, p_0, q_0, S_{VD}, S_{DJ}). The inserted segments are drawn according to a Markov process starting with the nucleotide distribution p_0 and with the transition matrix R, and running from the 5' side (left to right) for the VD segment, and from the 3' side (right to left) from the DJ segment. Similar models can be defined for the α chain or for BCR chains. Although here we describe the method for TRB only, it is also implemented for other chains in the software.

Since the same nucleotide sequence can be created by more than one specific recombination event, the generation probability of a nucleotide sequence is the sum of the probabilities of all possible events that generate the sequence: $P_{\text{gen}}^{\text{nt}}(\boldsymbol{\sigma}) = \sum_{E\to\boldsymbol{\sigma}} P_{\text{gen}}^{\text{rec}}(E)$, where the sum is over all recombination events E that produce the sequence $\boldsymbol{\sigma} = (\sigma_1,\ldots,\sigma_n)$. The probability of generation of an amino acid sequence, $\mathbf{a} = (a_1,\ldots,a_L)$ is the sum of the probabilities of all nucleotide sequences that translate into the amino acid sequence:

$$P_{\text{gen}}^{\text{aa}}(a_1,\ldots,a_L) = \sum_{\boldsymbol{\sigma}\sim\mathbf{a}} P_{\text{gen}}^{\text{nt}}(\sigma_1,.,\sigma_{3L}) = \sum_{E\to\boldsymbol{\sigma}\sim\mathbf{a}} P_{\text{gen}}^{\text{rec}}(E), \tag{2}$$

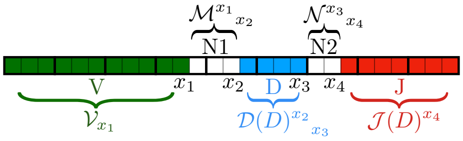
$\mathcal{M}^{x_1}{}_{x_2}$
$\mathcal{N}^{x_3}{}_{x_4}$
N1
N2
V
x_1
x_2
D
x_3
x_4
J
$\mathcal{V}_{x_1}$
$\mathcal{D}(D)^{x_2}{}_{x_3}$
$\mathcal{J}(D)^{x_4}$

FIG. 1: Partitioning a CDR3 sequence: boxes correspond to nucleotides and are indexed by integers. Each group of three boxes (identified by heavier boundary lines) corresponds to an amino acid. The nucleotide positions $x_1,\ldots,x_4$ identify the boundaries between different elements of the partition. The $\mathcal{V}$, $\mathcal{M}$, $\mathcal{D}(D)$, $\mathcal{N}$ and $\mathcal{J}(D)$ matrices define cumulated weights corresponding to each of the 5 elements.

where the $\sim$ sign indicates that $\boldsymbol{\sigma}$ translates into $\mathbf{a}$. We can generalize this approach to any scheme that groups nucleotide triplets, or codons, into arbitrary classes, which we still denote by $\boldsymbol{\sigma}\sim\mathbf{a}$. In the formulation above, these classes simply group together codons with the same translation according to the standard genetic code. In an example of generalization, all codons that code for amino acids with a common chemical property, e.g. hydrophobicity or charge, could be grouped into a single class. In that formulation, $(a_1,\ldots,a_L)$ would correspond to a sequence of symbols denoting that property. More generally, any grouping of amino acids can be chosen (including one where any amino acid is acceptable), and the partition can be position dependent. Thus, the generation probability of arbitrary "motifs" can be queried. In the following, for ease of exposition, we restrict our attention to the case where $\mathbf{a}$ is an amino acid sequence.

B. Dynamic programming computation of the generation probability of amino acid sequences

We now give an overview of how OLGA computes Eq. 2 without performing the sum explicitly, using dynamic programming. Fig. S1-S2 give a graphical overview of the method, and details of the method implementation can be found in SI Secs. I and II and in the code manual. Given the genomic nucleotide sequences of the possible gene templates, together with a specific model of the type described in Eq. A1, the algorithm computes the net probability of generating a recombined gene with a given CDR3 amino acid sequence under a given set of V and J gene choices.

Each recombination event implies an annotation of the CDR3 sequence, assigning a different origin to each nucleotide (V, N1, D, N2, or J, where N1 and N2 are the VD and DJ insertion segments, respectively) that parses the sequence into 5 contiguous segments (see schematic in Fig. 1). The principle of the method is to sum over the probabilities of all choices of nucleotides consistent with the known amino acid sequence, over the possible locations of the 4 boundaries (x_1, x_2, x_3, and x_4) between the 5 segments, and over the possible V, D, and J genomic templates (Fig. 1). We do this in a recursive way using matrix operations by defining weights that accumulate the probabilities of events from the left of a position x (i.e. up to x), and weights that accumulate events from the right of x (i.e. from $x+1$ on). Specifically, we define the following index notation: $\mathcal{X}_x$ with a subscript called left index, accumulates weights from the left of x; $\mathcal{Y}^x$, with a superscript called right index, accumulates weights from the right of x; a matrix $\mathcal{X}^x{}_y$ corresponds to accumulated weights from position $x+1$ to y (as will be explained shortly, these objects may have suppressed nucleotide indices as well). $P^{\rm aa}_{\rm gen}$ is calculated recursively by matrix-like multiplications as:

$$P^{\rm aa}_{\rm gen}(\mathbf{a}) = \sum_{x_1,x_2,x_3,x_4} \mathcal{V}_{x_1}\mathcal{M}^{x_1}{}_{x_2} \sum_D \left[\mathcal{D}(D)^{x_2}{}_{x_3}\mathcal{N}^{x_3}{}_{x_4}\mathcal{J}(D)^{x_4}\right]. \tag{3}$$

The vector $\mathcal{V}_x$ corresponds to a cumulated probability of the V segment finishing at position x; $\mathcal{M}^x{}_y$ is the probability of the VD insertion extending from $x+1$ to y; $\mathcal{N}^x{}_y$ is the same for DJ insertions; $\mathcal{D}^x{}_y(D)$ corresponds to weights of the D segment extending from $x+1$ to y, conditioned on the D germline choice being D; $\mathcal{J}^x(D)$ gives the weight of J segments starting at position $x+1$ conditioned on the D germline being D. This D dependency is necessary to account for the dependence between the D and J germline segment choices [27]. All the defined vectors and matrices depend implicitly on the amino acid sequence $(a_1,\ldots,a_L)$, but we leave this dependency implicit to avoid making the notation too cumbersome.

Because we are dealing with amino acid sequences encoded by triplet nucleotide codons, we need to keep track of the identity of the nucleotide at the beginning or the end of a codon. Depending on the position of the index x in the codon, the objects defined above may be vectors of size 4 (or 4×4 matrices) in the suppressed nucleotide index. We use conventions that depend on whether we are considering left or right indices, as follows.

If x is a multiple of 3, i.e. $x = 0 \pmod 3$, then we do not keep nucleotide information and both $\mathcal{X}_x$ and $\mathcal{Y}^x$ are scalars (whether x is a left or a right index). If $x = 1 \pmod 3$, then $\mathcal{X}_x$ must be interpreted as a row vector of 4 numbers, $\mathcal{X}_x(\sigma)$, $\sigma = A, T, G, C$, corresponding to the cumulated probability weight that the nucleotide at position x (first position of the codon) takes value σ. If $x = 2 \pmod 3$, then $\mathcal{X}_x$ is also a row vector of 4 numbers, $\mathcal{X}_x(\sigma)$, but with a different interpretation: it corresponds to the cumulated probability up to position x, with the additional constraint that the nucleotide at position $x+1$ (the last position in the codon) *can* take value σ (the value is 0 otherwise). For right indices, the interpretation is reversed and the entries are column vectors: when $x = 1 \pmod 3$ the $\mathcal{Y}^x$ is a column vector containing the cumulated weights from $x+1$ onwards, with the constraint that the nucleotide at x *can* be σ, and when $x = 2 \pmod 3$, it is the probability weight that the nucleotide at position $x+1$ *is* σ. Generalizing to matrices, $\mathcal{X}^x{}_y$ is a 4x4, 4x1, 1x4, or 1x1 matrix depending on whether the x and y positions are multiples of 3 or not, with the same rules as for vectors for each type of index.

Entries with left indices are interpreted as row vectors, and entries with right indices as column vectors. Thus, in Eq. B2 contractions between left and right indices correspond to dot products over the 4 nucleotides when the index is not a multiple of 3, and simply a product of scalars when it is.

The entries of the matrices corresponding to the germline segments, $\mathcal{V}$, $\mathcal{D}(D)$, and $\mathcal{J}(D)$, can be calculated by simply summing over the probabilities of different germline nucleotide segments compatible with the amino acid sequence $(a_1,\ldots,a_L)$ with conditions on deletions to achieve the required segment length. For instance, the $\mathcal{V}$ matrix elements are given by:

$$\mathcal{V}_x(\sigma) = \sum_V P_{\rm V}(V)P_{\rm delV}(l_V - x)\mathbb{I}(s^V_x = \sigma)\mathbb{I}(\mathbf{s}^V_{1:x}\sim\mathbf{a}_{1:i}) \text{ if } u=1$$

$$\mathcal{V}_x(\sigma) = \sum_V P_{\rm V}(V)P_{\rm delV}(l_V - x)\mathbb{I}((\mathbf{s}^V_{1:x},\sigma)\sim\mathbf{a}_{1:i}) \text{ if } u=2,$$

$$\mathcal{V}_x = \sum_V P_{\rm V}(V)P_{\rm delV}(l_V - x)\mathbb{I}(\mathbf{s}^V_{1:x}\sim\mathbf{a}_{1:i}) \text{ if } u=3, \tag{4}$$

where $x = 3(i-1)+u$, i.e. x is the $u^{\rm th}$ nucleotide of the $i^{\rm th}$ codon, $\mathbf{s}^V$ the sequence of the V germline gene, and $\mathbb{I}$ the indicator function. The $\sim$ sign is generalized to incomplete codons so that it returns a true value if there exists a codon completion that agrees with the motif $\mathbf{a}$. Detailed formulas for the other segments are derived using the same principles and are given in the SI Appendix. The sums in Eq. 4 (and equivalent expressions for J) can be restricted to particular germline genes to compute the generation probability of particular VJ-CDR3 combinations.

The entries of the insertion segment N1 are calculated using the following formula:

$$\mathcal{M}^x{}_y = P_{\rm insVD}(y-x)L^u_{a_i}T_{a_{i+1}}\ldots T_{a_{j-1}}R^v_{a_j}, \tag{5}$$

with $y = 3(j-1)+v$ (and $x = 3(i-1)+u$ as in Eq. 4). The transfer matrix

$$T_a(\tau,\sigma) = \sum_{(n_1,n_2,\sigma)\sim a} S_{\rm VD}(\sigma|n_2)S_{\rm VD}(n_2|n_1)S_{\rm VD}(n_1|\tau) \tag{6}$$

corresponds to the probability of inserting a codon coding for a and ending with nucleotide σ, knowing that the previous codon ended with nucleotide τ. L^u_a and R^v_a are vectors or matrices with different definitions depending on the values of x and y modulo 3, corresponding to the probabilities of inserting incomplete codons on the left and right ends of the insertion segment. Eq. 5 is only valid for $j > i$, but similar formulas describe the case $i = j$. The precise definitions of L and R, the $i = j$ case, and the formulas for $\mathcal{N}$ and the N2 insertion segment, which is exactly equivalent, are all given in detail in the SI Appendix.

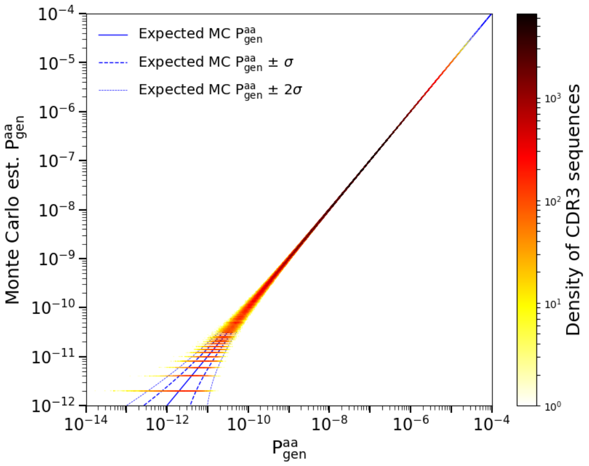

FIG. 2: Monte Carlo estimate of the generation probability of amino acid CDR3 sequences, $P^{\text{aa}}_{\text{gen}}$, versus OLGA's predictions (mouse TRB). The horizontal lines at the lower left of the plot represent CDR3s that were generated once, twice, etc, in the MC sample. The one- and two-sigma curves display the deviations from exact equality between simulated and computed P_{gen} to be expected on the basis of Poisson statistics.

The matrix product of Eq. 5 can be calculated recursively, requiring only 4×4 matrix multiplications. Thus, all $\mathcal{M}^x{}_y$ elements can be calculated in $\mathcal{O}(L^2)$ operations, instead of the exponential time that would be required using brute-force summation over nucleotides in degenerate codons. Finally, since the sums of Eq. B2 can also be done recursively through $L \times L$ matrix operations, the whole procedure has $\mathcal{O}(L^2)$ computational complexity.

III. RESULTS

A. Method validation

To verify the correctness of the OLGA code, we compared its predictions for generation probabilities to those estimated by Monte Carlo (MC) sequence generation [29]. MC estimation is done by drawing events from a given generative model, binning according to the resulting CDR3 amino acid sequence, and normalizing by the total number of recombination events. The scatter plot of the estimated generation probabilities for these sequences against the values predicted by OLGA gives a direct test of the algorithm. As MC estimation is susceptible to Poisson sampling noise, it is important to ensure that enough events are drawn to accurately assess the generative probabilities of individual CDR3 sequences. For this reason, we made the comparison using a generative model inferred from a mouse, rather than human, T cell repertoire, because of the significantly lower entropy of mouse repertoires [35]. The specific model was inferred by IGoR [25] using $\sim$ 70000 out-of-frame TRB sequences from a mature mouse thymus. MC estimation was done by generating 5×10^{11} recombination events, from which the first 10^6 unique CDR3 amino acid sequences are counted to serve as a sample for the comparison. This procedure provided good sequence coverage, with $> 98\%$ of sequences generated at least twice and $> 95\%$ of sequences generated at least 10 times. As Fig. 2 shows for mouse TRB (see Fig. S3 for human TRA), MC estimation and OLGA calculation are in agreement (up to Poisson noise in the MC estimate). The Kullback-Leibler divergence between the two distributions, a formal measure of their agreement, is a mere 4.82×10^{-7} bits.

B. Comparison of performance with existing methods

We compared the performance of OLGA to other methods. Direct calculation of amino acid sequence generation probability using OLGA is orders of magnitude faster than the two possible alternative methods: MC estimation (as described above), or exhaustive enumeration of the generative events giving rise to a given amino acid sequence. OLGA took 6 CPU hrs to compute the generation probabilities of the 10^6 amino acid sequences, i.e. 47 seqs/CPU/sec for mouse TRB (see SI Sec. III and Table S1 for runtimes of other loci). By comparison, MC estimation required 4313 CPU hrs. The scaling for the MC estimation does not depend on the number of queried sequences, but instead is determined by the number of recombinations needed to control the Poisson noise, which scales inversely with generation probability. In practice, to determine the generation probability of a typical sequence (which can be as low 10^{-20}, see Fig. 3 and below), one needs to generate very large datasets, and thus the generation probability of many sequences cannot be calculated by the MC method.

Alternatively, one could list all possible nucleotide sequences that translate to a particular amino acid CDR3 and sum the generation probabilities of each nucleotide sequence, using the IGoR algorithm [25]. Each amino acid sequence in the mouse validation sample is, on average, coded for by 1.84 billion nucleotide sequences (and much more for human TRB). Since IGoR computes generation probabilities of nucleotide sequences at the rate of $\sim$ 60 seqs/CPU/sec, it would take $\sim$ 8500 CPU hrs to compute the generation probability of a *single* amino acid sequence. A systematic comparison of OLGA with IGoR (Fig. S4) and MC estimation (Figs. S4 and S5) as a function of the number of analysed sequences and their CDR3 lengths shows that OLGA is faster than both other methods for all practical purposes (see Sec. IV for details).

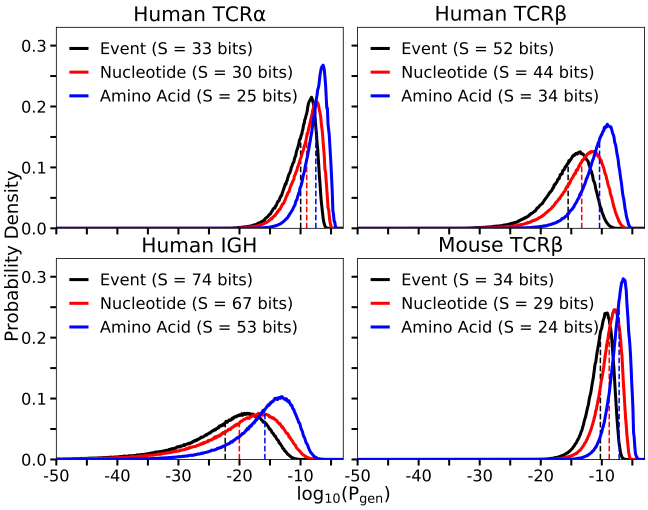

FIG. 3: Distributions of probabilities of recombination events ($P_{\rm gen}^{\rm rec}$), nucleotide CDR3 sequences ($P_{\rm gen}^{\rm nt}$), and CDR3 amino acid sequences ($P_{\rm gen}^{\rm aa}$) in different contexts. Each curve is determined by Monte Carlo sampling of 10^6 productive sequences for the indicated locus, and computing its generation probabilities at the three different levels. Entropies in bits (S) are, up to a $\ln(2)/\ln(10)$ factor, the negative of the mean of each distributions, indicated by dotted lines.

C. Distribution of generation probabilities and diversity

V(D)J recombination produces very diverse repertoires of nucleotide sequences, with a very broad distribution of generation probabilities spanning up to 20 orders of magnitude [6, 27]. This distribution gives a comprehensive picture of the diversity of the process, and can be used to recapitulate many classical diversity measures [26], and to predict the overlap between the repertoires of different individuals [8]. In particular, the opposite of the mean logarithm of the generation probability, $-\langle \log_2 P_{\rm gen}\rangle$, is equal to the entropy of the process. While previous work focused on nucleotide sequence generation, OLGA allows us to compute this distribution for amino acid sequences.

Fig. 3 shows the distribution of $P_{\rm gen}^{\rm aa}$ for 4 loci: human and mouse TRB, human TRA, and human IGH, and compares it to the distributions of nucleotide sequence generation probabilities, $P_{\rm gen}^{\rm nt}$, and recombination event probabilities, $P_{\rm gen}^{\rm rec}$. While all these datasets are based on DNA RepSeq, we checked that the generation probability distribution was robust to the choice of protocol by computing the TRB distribution for independent datasets generated by RNA RepSeq [38, 44, 47] (Figs. S6 and S7, and SI Sec. V). The generation models used here and elsewhere in this paper were taken from Marcou *et al.* [25], except for the human TRB model which was relearned using IGoR from one individual in Emerson *et al.* [9] as a check. Going from recombination events to nucleotide sequences to amino acid sequences leads to substantial shifts in the distribution, and corresponding drops in entropies, as the distribution is progressively coarse-grained. Higher generation probability of a given receptor sequence leads to higher chance of finding it in any given individual. Generation probabilities may be constrasted to the scale set by the inverse of the number of independent recombination events (estimated between 10^8 [31] and 10^{10} [22] for human TCR). Generation probabilities above this limit (10^{-10} to 10^{-8} for human TCR) can be considered “large” as the corresponding receptor will almost surely exist in each individual [8]. Another relevant scale to distinguish small from large generation probabilities is given by their geometric mean (dashed lines in Fig. 3).

D. Cross-species generation probabilities

While distinct species differ in their generation mechanisms, they may yet be able to generate the same CDR3s. Using OLGA, we computed the probabilities of producing human TRB CDR3s by the mouse recombination model, and vice versa (details in SI Sec. VI). An impressive 72.6% of human CDR3s can theoretically be produced by mice, and 100% of mouse CDR3s can be produced by humans. While cross-species generation probabilities are lower than intra-species ones (Fig. S8), they are correlated (Fig. S9). These results suggest that CDR3s observed in the repertoires of humanized mouse models of human diseases could be relevant for predicting their presence in human repertoires as well. OLGA allows for evaluating this potential, and could be used to inform clinical trials.

E. Generation probability of specific TCR

We can use OLGA to assess the total fraction of the generated repertoire that is specific to any given epitope, simply by summing the generation probabilities of all TRB sequences known to bind specifically to that epitope:

$$P_{\rm gen}^{\rm func}(\text{epitope}) = \sum_{\mathbf{a}\,|\,\text{epitope}} P_{\rm gen}^{\rm aa}(\mathbf{a}), \qquad (7)$$

where ”$\mathbf{a}\,|\,$epitope” means that the amino acid sequence **a** recognizes the epitope. Many experiments, based *e.g.* on multimer sorting assays [2, 14] or T-cell culture assays, have established lists of epitope-specific TCR sequences for a number of disease-related epitopes. We used the VDJdb database [37], which aggregates such experiments, to compute $P_{\rm gen}^{\rm func}$ of all TRB known to be reactive against several epitopes. In Fig. 4 we show results for 4 epitopes associated with Hepatitis C, and 5 epitopes associated with Influenza A. The net fraction of the repertoire specific to these epitopes (10^{-7} to 10^{-4}) is large in the sense defined above, meaning that any individual is likely to have many copies of reactive T cells in their naive repertoire.

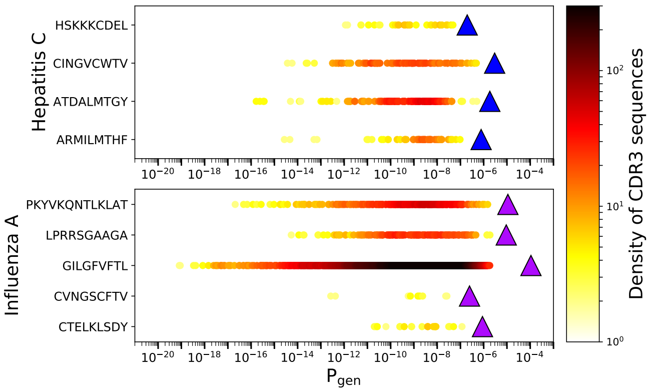

FIG. 4: Generation probabilities of human CDR3s that respond to hepatitis C and influenza A epitopes. $P_{\rm gen}^{\rm aa}$ of sequences that respond to an epitope are plotted as circles (color encodes density of the points). The fraction of the repertoire specific to each epitope ($P_{\rm gen}^{\rm func}$ as defined in Eq. 7) is obtained as the sum of the $P_{\rm gen}^{\rm aa}$ for each of the corresponding sequences (values plotted as triangles).

The presence of any specific TCR in the repertoire will be affected by the recombination probability of both its α and β chains, and also by function-dependent selective pressures. Assessing accurately the fraction of reactive TCRs in the blood is beyond the scope of this method. However, it is still interesting to ask whether epitope-specific TRB sequences had higher generation probabilities than regular sequences, either because of observational biases, or because the immune system might have evolved to make them more likely to be produced. To answer that question, we display in Fig. 5 the $P_{\rm gen}^{\rm aa}$ distribution of the sequences listed in VDJdb that are specific to any epitope of each of 6 commonly studied viruses. For comparison we plot the $P_{\rm gen}^{\rm aa}$ distribution of the full TRB sequence repertoire of a healthy donor (data taken from Emerson *et al.* [9]).

The viral distributions are very similar to each other, and also to the healthy repertoire background, meaning that the ability of a CDR3 to respond to a particular disease epitope is not strongly correlated with its generation probability. To see whether this result was confirmed in the case of a real infection, we repeated the same analysis on TRB RepSeq data from T-cells responding to three different types of pathogens (fungus, bacteria, and toxin) [1]. Consistently, we found that their distribution of generation probability was identical to that of naive sequences (SI Fig. S10 and SI Sec. VII).

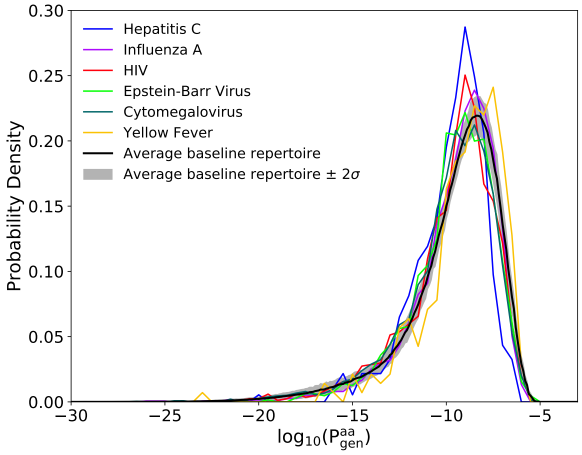

FIG. 5: Distributions of TRB generation probabilities $P_{\rm gen}^{\rm aa}$ for sequences in the VDJdb database that bind to any epitopes of 6 different viruses (colored curves). For comparison, we plot (black curve) the same distribution for the unsorted TRB repertoire of a typical healthy subject; the 2σ variance represents biological variability across multiple individuals (data from Emerson *et al.* [9])

F. Model accurately predicts the frequencies of sequences and of groups of specific sequences

To compare OLGA's predictions with sequence occurrence frequencies in real data, we used the aggregated TRB repertoire of 658 human subjects described in Emerson *et al.* [9] as a test resource. More specifically, we measured the frequencies in this large dataset of the specific CDR3 sequences contained in the VDJdb database [37], and compared them to the values assigned by OLGA. When measuring frequencies we discarded read count information, recording only the presence or absence of nucleotide sequences in each individual in order to eliminate effects of clonal expansion and PCR amplification bias, averaging over the 648 individuals in the Emerson *et al.* [9] dataset to get reliable estimates of frequencies. Each sequence in the VDJdb database is displayed as a dot in Fig. 6, and the resulting distribution shows a strong correspondence between mean frequency in the large data set and the predicted $P_{\rm gen}^{\rm aa}$ of that sequence.

We then measured the fraction of CDR3s in the aggregated repertoire that is specific to epitopes associated with 6 viruses (using lists of specific sequences in VDJdb), and compared it to OLGA's prediction, $P_{\rm gen}^{\rm func}$. The agreement was again excellent (triangles in Fig. 6). Again we observe that most epitope-specific sequence groups have large enough frequencies to be found in any individual. Thus, the model can be used to predict the size of repertoire subsets specific to any epitope, as long as specificity data are available for this epitope.

G. Generation probability of sequence motifs

OLGA can also compute the generation probability of any sequence motif, encoded by a string of multiple choices of amino acids. We apply this feature to calculate the net frequency of epitope-specific motifs, and of

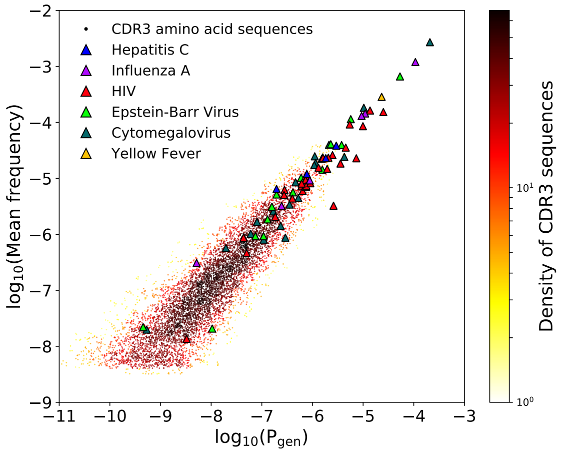

FIG. 6: Mean occurrence frequencies across a collection of 658 human samples of all CDR3 sequences in the VDJdb database, plotted against their computed $P_{\text{gen}}^{\text{aa}}$ (dots, colored by their density in the plot). Also, the net occurrence frequency in the VDJdb database of epitope-related collections of sequences, plotted against their computed $P_{\text{gen}}^{\text{func}}$ (triangles, colored to identify the virus the epitope belongs to).

TABLE I: Epitope-specific TCR motifs for the Epstein-Barr virus HLA-A*0201-BMLF$_{1280}$ (BMLF) and influenza virus HLA-A*0201-M$_{158}$ (M1) epitopes from Dash *et al.* [2], and their generation probabilities. Each motif was associated with specific V/J gene choices. In the motifs we use the conventions: X, any one amino acid; [A..B], any one of the listed amino acids; X{0,}, arbitrary amino acid string.

epitope : chain : V/J	CDR3 motif	P_{gen}
BMLF : α : 5/31	CAXD[NSDA]NARLMF	$1.8 \cdot 10^{-7}$
BMLF : β : 20-1/1-2,1-3	CSARDX[TV]GNX{0,}	$5.1 \cdot 10^{-7}$
M1 : α : 27/42	CAXGGSQGNLIF	$2.2 \cdot 10^{-5}$
M1 : β : 19/all	CASSXR[SA][STAG]X[ET]Q[YF]F	$1.7 \cdot 10^{-6}$

motifs that define the TRA sequence of invariant T-cells.

T-cell sequences that can bind a given epitope are often closely related to each other, and this similarity can sometimes be partially captured by sequence motifs. We evaluated the probabilities of motifs derived from a recent study of CDR3 sequence specificity to a variety of epitopes [2]. We took two motifs corresponding to TRA and TRB VJ-CDR3 combinations of TCRs that are known to bind the Epstein-Barr virus HLA-A*0201-BMLF$_{1280}$ (BMLF) and the influenza virus HLA-A*0201-M$_{158}$ (M1) epitopes. The motifs and generation probabilities are reported in Table I.

As a second application, we estimated the probabilities of generating a TRA chain corresponding to one of the motifs associated with Mucosal associated invariant T cells (MAIT) and invariant natural killer T cells (iNKT).

The motifs, which were collected from Gherardin *et al.* [13], and their probabilities are shown in Table II. The relatively high values for these motifs imply that these invariant chains are generated with high frequency in the primary repertoire and shared by all individuals, confirming the conclusions of Venturi *et al.* [42].

TABLE II: Generation probabilities of motifs corresponding to invariant T cell (iNKT and MAIT cells) TRA chain, assembled from serquence in Gherardin *et al.* [13].

Type	V/J	CDR3 motif	P_{gen}
iNKT	10/18	CVVSDRGSTLGRLYF	$1.26 \cdot 10^{-6}$
MAIT	1-2/33	CAV[KSM]DSNYQLI[WF]	$1.79 \cdot 10^{-5}$
MAIT	1-2/12	CAVMDSSYKLIF	$4.71 \cdot 10^{-6}$
MAIT	1-2/20	CAVSDNDYKLSF	$3.11 \cdot 10^{-7}$

IV. DISCUSSION

Because the composition of the immune repertoire results from a stochastic process, the frequency with which distinct T- and B-cell receptors are generated is a quantity of primary interest. This frequency is computationally difficult to evaluate because each amino acid sequence can be created by a very large number of recombination events. Our tool overcomes that challenge with dynamic programming, allowing it to process ~ 50 sequences per second on a single CPU. In its current state OLGA can compute the probabilities of CDR3 sequences and motifs, with or without V/J restriction, of 4 chain loci (human and mouse TRB, human TRA, and human IGH), but the list can readily be expanded by learning recombination models for other loci and species using IGoR [25] which shares the same model format. Obvious additions include the light chains of BCR [40], and more mouse models. While the algorithm evaluates the probability of single chains, recent analyses show that chain pairing in TCR is close to independent [5, 15]. The probability of generating a whole TCR receptor can thus be computed by taking the product over the two chains.

OLGA can be used to compute baseline receptor frequencies and to identify outlying sequences in repertoire sequencing datasets. In Elhanati *et al.* [8] we used it to shed light on the question of public repertoires — composed of sequences shared by many individuals — and predict quantitatively its origin by convergent recombination [23, 24, 41]. Deviations from the baseline expectancy have been used to identify disease-associated TCR from cohorts of patients [9, 10, 12, 34, 48], and to identify clusters of reactive TCRs from tetramer experiments [14] and vaccination studies [30]. Such estimates could be made faster and more reliable by OLGA, especially for rare sequences, and without the need for a negative control cohort [29]. In the future, OLGA could be useful in vaccine and therapy design by focusing attention on clonotypes that are likely to be present in every individual.

We applied OLGA to an experimental database of TCR responding to a variety of disease-associated epitopes. These selected TCR do not differ in their generation probabilities from those of random TCR found in the blood of healthy donors. However, some viral epitopes bind a much larger fraction of the repertoire than others. This observation has potentially important consequences for vaccine design. Since vaccine epitopes stimulate TCR in a pre-existing repertoire, epitopes targeting receptor sequences that are more likely to be generated will have a higher success rate in a wider range of individuals. OLGA can be used to identify such epitopes by computing their specific repertoire fractions, $P_{\text{gen}}^{\text{func}}$. While our examples are restricted to TCR, OLGA can also handle BCR and could be used to compute the generation probabilities of BCR precursors of highly reactive or broadly neutralizing antibodies, and thus guide vaccine design in that case as well. The algorithm does not yet handle hypermutations, and extending it to include them would be a useful development.

Acknowledgements. The work of TM and AMW was supported in part by grant ERCCOG n. 724208. The work of ZS and CC was supported in part by NSF grant PHY-1607612. The work of CC was also supported in part by NSF grant PHY-1734030. The work of YE was supported by a fellowship from the V Foundation. The authors declare no conflicts of interest.

[1] Becattini, S., Latorre, D., Mele, F., Foglierini, M., De Gregorio, C., Cassotta, A., Fernandez, B., Kelderman, S., Schumacher, T. N., Corti, D., Lanzavecchia, A., and Sallusto, F. (2015). Functional heterogeneity of human memory cd4+ t cell clones primed by pathogens or vaccines. *Science*, **347**(6220), 400–406.

[2] Dash, P., Fiore-Gartland, A. J., Hertz, T., Wang, G. C., Sharma, S., Souquette, A., Crawford, J. C., Clemens, E. B., Nguyen, T. H. O., Kedzierska, K., La Gruta, N. L., Bradley, P., and Thomas, P. G. (2017). Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature*, **547**(7661), 89–93.

[3] DeWitt, W. S., Lindau, P., Snyder, T. M., Sherwood, A. M., Vignali, M., Carlson, C. S., Greenberg, P. D., Duerkopp, N., Emerson, R. O., and Robins, H. S. (2016). A Public Database of Memory and Naive B-Cell Receptor Sequences. *PLoS One*, **11**(8), e0160853.

[4] DeWitt, W. S., Smith, A., Schoch, G., Hansen, J. A., Matsen, F. A., and Bradley, P. H. (2018). Human T cell receptor occurrence patterns encode immune history, genetic background, and receptor specificity. *bioRxiv*, page 313106.

[5] Dupic, T., Marcou, Q., Mora, T., and Walczak, A. M. (2018). Genesis of the $\alpha\beta$ T-cell receptor. *arXiv:1806.11030*.

[6] Elhanati, Y., Sethna, Z., Marcou, Q., Jr, G. C., Mora, T., and Walczak, A. M. (2015). Inferring processes underlying B-cell repertoire diversity. *Philos Trans R Soc Lond, B, Biol Sci*, **370**, 20140243.

[7] Elhanati, Y., Marcou, Q., Mora, T., and Walczak, A. M. (2016). repgenhmm: a dynamic programming tool to infer the rules of immune receptor generation from sequence data. *Bioinformatics*, **32**(13), 1943–1951.

[8] Elhanati, Y., Sethna, Z., Callan, C. G., Mora, T., and Walczak, A. M. (2018). Predicting the spectrum of TCR repertoire sharing with a data-driven model of recombination. *Immunological reviews*, **284**(1), 167–179.

[9] Emerson, R. O., DeWitt, W. S., Vignali, M., Gravley, J., Hu, J. K., Osborne, E. J., Desmarais, C., Klinger, M., Carlson, C. S., Hansen, J. A., Rieder, M., and Robins, H. S. (2017). Immunosequencing identifies signatures of cytomegalovirus exposure history and HLA-mediated effects on the T cell repertoire. *Nature Genetics*, **49**(5), 659–665.

[10] Faham, M., Carlton, V., Moorhead, M., Zheng, J., Klinger, M., Pepin, F., Asbury, T., Vignali, M., Emerson, R. O., Robins, H. S., Ireland, J., Baechler-Gillespie, E., and Inman, R. D. (2017). Discovery of T Cell Receptor β Motifs Specific to HLA-B27-Positive Ankylosing Spondylitis by Deep Repertoire Sequence Analysis. *Arthritis Rheumatol.*, **69**(4), 774–784.

[11] Freeman, J. D., Warren, R. L., Webb, J. R., Nelson, B. H., and Holt, R. a. (2009). Profiling the T-cell receptor beta-chain repertoire by massively parallel sequencing. *Genome Res.*, **19**(10), 1817–1824.

[12] Fuchs, Y. F., Eugster, A., Dietz, S., Sebelefsky, C., Kühn, D., Wilhelm, C., Lindner, A., Gavrisan, A., Knoop, J., Dahl, A., Ziegler, A. G., and Bonifacio, E. (2017). CD8+T cells specific for the islet autoantigen IGRP are restricted in their T cell receptor chain usage. *Sci. Rep.*, **7**(March), 1–10.

[13] Gherardin, N. A., Keller, A. N., Woolley, R. E., Le Nours, J., Ritchie, D. S., Neeson, P. J., Birkinshaw, R. W., Eckle, S. B., Waddington, J. N., Liu, L., Fairlie, D. P., Uldrich, A. P., Pellicci, D. G., McCluskey, J., Godfrey, D. I., and Rossjohn, J. (2016). Diversity of T Cells Restricted by the MHC Class I-Related Molecule MR1 Facilitates Differential Antigen Recognition. *Immunity*, **44**(1), 32–45.

[14] Glanville, J., Huang, H., Nau, A., Hatton, O., Wagar, L. E., Rubelt, F., Ji, X., Han, A., Krams, S. M., Pettus, C., Haas, N., Arlehamn, C. S. L., Sette, A., Boyd, S. D., Scriba, T. J., Martinez, O. M., and Davis, M. M. (2017). Identifying specificity groups in the T cell receptor repertoire. *Nature*, **547**(7661), 94–98.

[15] Grigaityte, K., Carter, J. A., Goldfless, S. J., Jeffery, E. W., Ronald, J., Jiang, Y., Koppstein, D., Briggs, A. W., Church, G. M., and Atwal, G. S. (2017). Single-cell sequencing reveals $\alpha\beta$ chain pairing shapes the T cell repertoire. *bioRxiv:213462*.

[16] Heather, J. M., Ismail, M., Oakes, T., and Chain, B. (2017). High-throughput sequencing of the T-cell receptor repertoire: pitfalls and opportunities. *Brief. Bioinform.*, (September 2016), bbw138.

[17] Horns, F., Vollmers, C., Dekker, C. L., and Quake, S. R. (2017). Signatures of selection in the human antibody repertoire: selective sweeps, competing subclones, and neutral drift. *bioRxiv*, **doi.org/10.1101/145052**.

[18] Howie, B., Sherwood, A. M., Berkebile, A. D., Berka, J., Emerson, R. O., Williamson, D. W., Kirsch, I., Vignali, M., Rieder, M. J., Carlson, C. S., and Robins, H. S. (2015). High-throughput pairing of T cell receptor a and b sequences. *Sci. Transl. Med.*, **7**(301), 301ra131.

[19] Jiang, N., He, J., Weinstein, J. A., Penland, L., Sasaki, S., He, X.-S., Dekker, C. L., Zheng, N.-Y., Huang, M., Sullivan, M., Wilson, P. C., Greenberg, H. B., Davis, M. M., Fisher, D. S., and Quake, S. R. (2013). Lineage structure of the human antibody repertoire in response to influenza vaccination. *Sci. Transl. Med.*, **5**(171), 171ra19.

[20] Komech, E., Pogorelyy, M., Egorov, E., Britanova, O., Rebrikov, D., Bochkova, A., Shmidt, E., Shostak, N., Shugay, M., Lukyanov, S., Mamedov, I., Lebedev, Y., Chudakov, D., and Zvyagin, I. (2018). CD8+ T cells with characteristic TCR beta motif are detected in blood and expanded in synovial fluid of ankylosing spondylitis patients. *Rheumatology (Oxford, England)*, **in press**(March), 1–8.

[21] Lindau, P. and Robins, H. S. (2017). Advances and Applications of Immune Receptor Sequencing in Systems Immunology. *Curr. Opin. Syst. Biol.*

[22] Lythe, G., Callard, R. E., Hoare, R. L., and Molina-París, C. (2016). How many TCR clonotypes does a body maintain? *Journal of theoretical biology*, **389**, 214–224.

[23] Madi, A., Shifrut, E., Reich-Zeliger, S., Gal, H., Best, K., Ndifon, W., Chain, B., Cohen, I. R., and Friedman, N. (2014). T-cell receptor repertoires share a restricted set of public and abundant CDR3 sequences that are associated with self-related immunity. *Genome Res.*, **24**(10), 1603–12.

[24] Madi, A., Poran, A., Shifrut, E., Reich-Zeliger, S., Greenstein, E., Zaretsky, I., Arnon, T., Laethem, F. V., Singer, A., Lu, J., Sun, P. D., Cohen, I. R., and Friedman, N. (2017). T cell receptor repertoires of mice and humans are clustered in similarity networks around conserved public CDR3 sequences. *eLife*, **6**.

[25] Marcou, Q., Mora, T., and Walczak, A. M. (2018). High-throughput immune repertoire analysis with IGoR. *Nature Communications*, **9**(1), 561.

[26] Mora, T. and Walczak, A. (2018). Quantifying lymphocyte receptor diversity. In J. D. Das and C. Jayaprakash, editors, *Syst. Immunol.*, pages 185–199. CRC Press.

[27] Murugan, A., Mora, T., Walczak, A. M., and Callan, C. G. (2012). Statistical inference of the generation probability of T-cell receptors from sequence repertoires. *Proceedings of the National Academy of Sciences of the United States of America*, **109**(40), 16161–6.

[28] Pogorelyy, M. V., Elhanati, Y., Marcou, Q., Sycheva, A. L., Komech, E. A., Nazarov, V. I., Britanova, O. V., Chudakov, D. M., Mamedov, I. Z., Lebedev, Y. B., Mora, T., and Walczak, A. M. (2017). Persisting fetal clonotypes influence the structure and overlap of adult human T cell receptor repertoires. *PLoS Comput. Biol.*, **13**(7), e1005572.

[29] Pogorelyy, M. V., Minervina, A. A., Chudakov, D. M., Mamedov, I. Z., Lebedev, Y. B., Mora, T., and Walczak, A. M. (2018a). Method for identification of condition-associated public antigen receptor sequences. *Elife*, **7**(D), 1–13.

[30] Pogorelyy, M. V., Minervina, A. A., Touzel, M. P., Sycheva, A. L., Komech, E. A., Kovalenko, E. I., Karganova, G. G., Egorov, E. S., Komkov, A. Y., Chudakov, D. M., Mamedov, I. Z., Mora, T., Walczak, A. M., and Lebedev, Y. B. (2018b). Precise tracking of vaccine-responding T-cell clones reveals convergent and personalized response in identical twins. *arXiv:1804.04485*.

[31] Qi, Q., Liu, Y., Cheng, Y., Glanville, J., Zhang, D., Lee, J.-Y., Olshen, R. A., Weyand, C. M., Boyd, S. D., and Goronzy, J. J. (2014). Diversity and clonal selection in the human T-cell repertoire. *Proceedings of the National Academy of Sciences*, **111**(36), 13139–13144.

[32] Robins, H. S., Campregher, P. V., Srivastava, S. K., Wacher, A., Turtle, C. J., Kahsai, O., Riddell, S. R., Warren, E. H., and Carlson, C. S. (2009). Comprehensive assessment of T-cell receptor beta-chain diversity in alphabeta T cells. *Blood*, **114**(19), 4099–4107.

[33] Robins, H. S., Srivastava, S. K., Campregher, P. V., Turtle, C. J., Andriesen, J., Riddell, S. R., Carlson, C. S., and Warren, E. H. (2010). Overlap and Effective Size of the Human CD8+ T Cell Receptor Repertoire. *Sci. Transl. Med.*, **2**(47), 47ra64–47ra64.

[34] Seay, H. R., Yusko, E., Rothweiler, S. J., Zhang, L., Posgai, A. L., Campbell-Thompson, M., Vignali, M., Emerson, R. O., Kaddis, J. S., Ko, D., Nakayama, M., Smith, M. J., Cambier, J. C., Pugliese, A., Atkinson, M. A., Robins, H. S., and Brusko, T. M. (2016). Tissue distribution and clonal diversity of the T and B cell repertoire in type 1 diabetes. *JCI Insight*, **1**(20), 1–19.

[35] Sethna, Z., Elhanati, Y., Dudgeon, C. R., Callan, C. G., Levine, A. J., Mora, T., and Walczak, A. M. (2017a). Insights into immune system development and function from mouse T-cell repertoires. *Proceedings of the National Academy of Sciences*, **114**(9), 2253–2258.

[36] Sethna, Z., Elhanati, Y., Dudgeon, C. S., Callan, C. G., Levine, A. J., Mora, T., and Walczak, A. M. (2017b). Insights into immune system development and function from mouse T-cell repertoires. *Proceedings of the National Academy of Sciences*, **114**(9), 2253–2258.

[37] Shugay, M., Bagaev, D. V., Zvyagin, I. V., Vroomans, R. M., Crawford, J. C., Dolton, G., Komech, E. A., Sycheva, A. L., Koneva, A. E., Egorov, E. S., Eliseev, A. V., Van Dyk, E., Dash, P., Attaf, M., Rius, C., Ladell, K., McLaren, J. E., Matthews, K. K., Clemens, E. B., Douek, D. C., Luciani, F., van Baarle, D., Kedzierska, K., Kesmir, C., Thomas, P. G., Price, D. A., Sewell, A. K., and Chudakov, D. M. (2018). VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Research*, **46**(D1), D419–D427.

[38] Sims, J. S., Grinshpun, B., Feng, Y., Ung, T. H., Neira, J. A., Samanamud, J. L., Canoll, P., Shen, Y., Sims, P. A., and Bruce, J. N. (2016). Diversity and divergence of the glioma-infiltrating t-cell receptor repertoire. *Proceedings of the National Academy of Sciences*, **113**(25), E3529–E3537.

[39] Six, A., Mariotti-Ferrandiz, M. E., Chaara, W., Magadan, S., Pham, H.-P. P., Lefranc, M.-P. P., Mora, T., Thomas-Vaslin, V., Walczak, A. M., Boudinot, P., Mariotti-Ferrandiz, E., Chaara, W., Magadan, S., Pham, H.-P. P., Lefranc, M.-P. P., Mora, T., Thomas-Vaslin, V., Walczak, A. M., and Boudinot, P. (2013). The past, present and future of immune repertoire biology - the rise of next-generation repertoire analysis. *Front. Immunol.*, **4**(413), 413.

[40] Toledano, A., Elhanati, Y., Benichou, J. I. C., Walczak, A. M., Mora, T., and Louzoun, Y. (2018). Evidence for shaping of light chain repertoire by structural selection. *Frontiers in Immunology*, **9**, 1307.

[41] Venturi, V., Chin, H. Y., Price, D. A., Douek, D. C., and Davenport, M. P. (2008). The Role of Production Frequency in the Sharing of Simian Immunodeficiency Virus-Specific CD8+ TCRs between Macaques. *The Journal of Immunology*, **181**(4), 2597–2609.

[42] Venturi, V., Rudd, B. D., and Davenport, M. P. (2013). Specificity, promiscuity, and precursor frequency in immunoreceptors. *Curr. Opin. Immunol.*, **25**(5), 639–645.

[43] Vollmers, C., Sit, R. V., Weinstein, J. A., Dekker, C. L., and Quake, S. R. (2013). Genetic measurement of memory B-cell recall using antibody repertoire sequencing. *Proc. Natl. Acad. Sci. U.S.A.*, **110**(33), 13463–13468.

[44] Wang, C., Sanders, C. M., Yang, Q., Schroeder, H. W., Wang, E., Babrzadeh, F., Gharizadeh, B., Myers, R. M., Hudson, J. R., Davis, R. W., and Han, J. (2010). High throughput sequencing reveals a complex pattern of dynamic interrelationships among human T cell subsets. *Proceedings of the National Academy of Sciences of the United States of America*, **107**(4), 1518–23.

[45] Weinstein, J. A., Jiang, N., White, R. A., Fisher, D. S., and Quake, S. R. (2009). High-throughput sequencing of the zebrafish antibody repertoire. *Science (80-.).*, **324**(5928), 807–810.

[46] Woodsworth, D. J., Castellarin, M., and Holt, R. a. (2013). Sequence analysis of T-cell repertoires in health and disease. *Genome Med.*, **5**(10), 98.

[47] Wu, J., Pendegraft, A. H., Byrne-Steele, M., Yang, Q., Wang, C., Pan, W., Lucious, T., Seay, T., Cui, X., Elson, C. O., Han, J., and Mannon, P. J. (2018). Expanded tcr-cdr3 clonotypes distinguish crohn's disease and ulcerative colitis patients. *Mucosal Immunology*, **11**(5), 1487–1495.

[48] Zhao, Y., Nguyen, P., Ma, J., Wu, T., Jones, L. L., Pei, D., Cheng, C., and Geiger, T. L. (2016). Preferential Use of Public TCR during Autoimmune Encephalomyelitis. *J. Immunol.*, **196**(12), 4905–4914.

Appendix A: Additional matrix definitions for VDJ algorithm

Recall that the generative VDJ model is defined as:

$$P_{\rm gen}^{\rm rec}(E) = P_{\rm V}(V)P_{\rm DJ}(D,J)P_{\rm delV}(d_V|V)P_{\rm delJ}(d_J|J)P_{\rm delD}(d_D,d'_D|D)P_{\rm insVJ}(\ell_{\rm VD})p_0(m_1)\left[\prod_{i=2}^{\ell_{VD}} S_{\rm VD}(m_i|m_{i-1})\right] \times P_{\rm insDJ}(\ell_{\rm DJ})q_0(n_{\ell_{DJ}})\left[\prod_{i=1}^{\ell_{DJ}-1} S_{\rm DJ}(n_i|n_{i+1})\right], \tag{A1}$$

with

$$P_{\rm gen}^{\rm aa}(a_1,\ldots,a_L) = \sum_{\boldsymbol{\sigma}\sim\boldsymbol{a}} P_{\rm gen}^{\rm nt}(\sigma_1,\ldots,\sigma_{3L}) = \sum_{E\to\boldsymbol{\sigma}\sim\mathbf{a}} P_{\rm gen}^{\rm rec}(E). \tag{A2}$$

As described in the main text, the dynamic programming algorithm can be summarized by the summation over the positions $x_1,\ x_2,\ x_3$, and x_4 of the following matrix multiplication:

$$P_{\rm gen}^{\rm aa}(a_1,\ldots,a_L) = \sum_{x_1,x_2,x_3,x_4} \mathcal{V}_{x_1}\mathcal{M}^{x_1}{}_{x_2} \times \sum_{D}\left[\mathcal{D}(D)^{x_2}{}_{x_3}\mathcal{N}^{x_3}{}_{x_4}\mathcal{J}(D)^{x_4}\right]. \tag{A3}$$

The interpretation of the left (subscript) and right (superscript) indices are detailed in the main text, and schematized in Fig. S1. The sums are performed iteratively using matrix multiplications, as detailed in Fig. **??**. As in the main text, the nucleotide indices will often be suppressed along with the implicit dependence on the amino acid sequence $(a_1,\ldots,a_L)$. For a given nucleotide position x_j, it will be convenient to refer to the amino acid index, and the position in the codon (from both the left and the right), so we introduce the following (graphically shown in the cartoon below): $x_j = 3(i_j-1)+u_j$, and u, so that i_j encodes the codon that index x_j belongs to, and u_j its position (from 1 to 3) within that codon, while u_j^* denotes the position taken from the right of index x_j+1 within its codon, so that $u_j^*=2$ if $u_j=1$, $u_j^*=1$ if $u_j=2$, and $u_j^*=3$ if $u_j=3$.

We now define the explicit forms for each of the matrices (note that we retain the indexing x_j from Eq A3):

a. $\mathcal{V}_{x_1}$

Contribution from the templated V genes. $\mathcal{V}_{x_1}$ can be a 1x1 or 1x4 matrix depending on u_1. $\mathbf{s}^V$ is the sequence of the V germline gene (read $5'$ to $3'$) from the conserved residue (generally the cysteine C) to the end of the gene. l_V is the length of $\mathbf{s}^V$. These equations are given in the main text.

$$\begin{aligned}
\mathcal{V}_{x_1}(\sigma) &= \sum_V P_{\rm V}(V)P_{\rm delV}(l_V - x_1|V)\mathbb{I}(s^V_{x_1}=\sigma)\mathbb{I}(\mathbf{s}^V_{1:x_1}\sim\mathbf{a}_{1:i_1}) \quad \text{if } u_1=1,\\
\mathcal{V}_{x_1}(\sigma) &= \sum_V P_{\rm V}(V)P_{\rm delV}(l_V - x_1|V)\mathbb{I}((\mathbf{s}^V_{1:x_1},\sigma)\sim\mathbf{a}_{1:i_1}) \quad \text{if } u_1=2,\\
\mathcal{V}_{x_1} &= \sum_V P_{\rm V}(V)P_{\rm delV}(l_V - x_1|V)\mathbb{I}(\mathbf{s}^V_{1:x_1}\sim\mathbf{a}_{1:i_1}) \quad \text{if } u_1=3.
\end{aligned} \tag{A4}$$

b. $\mathcal{M}^{x_1}{}_{x_2}$

Contribution from the non-templated N1 insertions (VD junction). $\mathcal{M}^{x_1}{}_{x_2}$ is defined as the product of transfer matrices, and can be a 1x1, 1x4, 4x1, or 4x4 matrix depending on u_1 and u_2. The transfer matrices are defined by the summed contributions of the Markov insertion model of all codons consistent with the amino acid a (thus summations are over nucleotides y, y_1, and y_2 to consider all allowed codons):

$$T_a(\tau,\sigma) = \sum_{(y_1,y_2,\sigma)\sim a} S_{\mathrm{VD}}(\sigma|y_2)S_{\mathrm{VD}}(y_2|y_1)S_{\mathrm{VD}}(y_1|\tau) \tag{A5}$$

$$F_a(\tau,\sigma) = S_{\mathrm{VD}}(\sigma|\tau)\mathbb{I}[\exists\sigma',\sigma'' \text{ s.t. } (\sigma,\sigma',\sigma'')\sim a] \tag{A6}$$

$$D_a(\tau,\sigma) = \sum_{(y_1,y_2,\sigma)\sim a} S_{\mathrm{VD}}(y_2|y_1)S_{\mathrm{VD}}(y_1|\tau) \tag{A7}$$

$$lT_a(\tau,\sigma) = \sum_{(\tau,y,\sigma)\sim a} S_{\mathrm{VD}}(\tau|y)p_0(y) \tag{A8}$$

$$lD_a(\tau,\sigma) = \sum_{(\tau,y,\sigma)\sim a} p_0(y) \tag{A9}$$

If $i_2 > i_1$:

$$\mathcal{M}^{x_1}{}_{x_2} = P_{\mathrm{insVD}}(x_2-x_1)L^{u_1}_{a_{i_1}}T_{a_{i_1+1}}\dots T_{a_{i_2-1}}R^{u_2}_{a_{i_2}} \tag{A10}$$

where:

$$L^{u_1}_{a_{i_1}} = \begin{cases} lT_{a_{i_1}} & \text{if } u_1 = 1 \\ diag(p_0) & \text{if } u_1 = 2 \\ S_{\mathrm{VD}}^{-1}p_0 & \text{if } u_1 = 3 \end{cases} \quad \text{and} \quad R^{u_2}_{a_{i_2}} = \begin{cases} F_{a_{i_2}} & \text{if } u_2 = 1 \\ D_{a_{i_2}} & \text{if } u_2 = 2 \\ T_{a_{i_2}}\vec{1} & \text{if } u_2 = 3 \end{cases} \tag{A11}$$

If $i_1 = i_2$:

$$\mathcal{M}^{x_1}{}_{x_2} = P_{\mathrm{insVD}}(x_2-x_1) \times \begin{array}{c|ccc} & u_2 = 1 & u_2 = 2 & u_2 = 3 \\ \hline u_1 = 1 & \mathbb{1} & 0 & 0 \\ u_1 = 2 & lD_{a_{i_1}} & \mathbb{1} & 0 \\ u_1 = 3 & lT_{a_{i_1}}\vec{1} & diag(p_0)\vec{1} & 1 \end{array} \tag{A12}$$

c. $\mathcal{D}(D)^{x_2}{}_{x_3}$

Contribution from the templated D genes. $\mathcal{D}(D)^{x_2}{}_{x_3}$ can be a 1x1, 1x4, 4x1, or 4x4 matrix depending on u_2^* and u_3^*. $\mathbf{s}^D$ is the sequence of the D germline gene (read 5′ to 3′) with length l_D.

$$\begin{aligned} \mathcal{D}(D)^{x_2}{}_{x_3}(\tau,\sigma) &= \sum_{d'_D} P_{\mathrm{delD}}(d_D,d'_D|D)\mathbb{I}[s^D_{d_D+1}=\tau]\mathbb{I}[s^D_{l_D-d'_D}=\sigma]\mathbb{I}[\mathbf{s}^D_{d_D+1:l_D-d'_D}\sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^*=1 \text{ and } u_3^*=1, \\ \mathcal{D}(D)^{x_2}{}_{x_3}(\tau,\sigma) &= \sum_{d'_D} P_{\mathrm{delD}}(d_D,d'_D|D)\mathbb{I}[s^D_{dD+1}=\tau]\mathbb{I}[(\mathbf{s}^D_{d_D+1:l_D-d'_D},\sigma)\sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^*=1 \text{ and } u_3^*=2, \\ \mathcal{D}(D)^{x_2}{}_{x_3}(\tau) &= \sum_{d'_D} P_{\mathrm{delD}}(d_D,d'_D|D)\mathbb{I}[s^D_{d_D+1}=\tau]\mathbb{I}[\mathbf{s}^D_{d_D+1:l_D-d'_D}\sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^*=1 \text{ and } u_3^*=3, \end{aligned} \tag{A13}$$

$$\begin{aligned} \mathcal{D}(D)^{x_2}{}_{x_3}(\tau,\sigma) &= \sum_{d'_D} P_{\mathrm{delD}}(d_D,d'_D|D)\mathbb{I}[s^D_{l_D-d'_D}=\sigma]\mathbb{I}[(\tau,\mathbf{s}^D_{d_D+1:l_D-d'_D})\sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^*=2 \text{ and } u_3^*=1, \\ \mathcal{D}(D)^{x_2}{}_{x_3}(\tau,\sigma) &= \sum_{d'_D} P_{\mathrm{delD}}(d_D,d'_D|D)\mathbb{I}[(\tau,\mathbf{s}^D_{d_D+1:l_D-d'_D},\sigma)\sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^*=2 \text{ and } u_3^*=2, \\ \mathcal{D}(D)^{x_2}{}_{x_3}(\tau) &= \sum_{d'_D} P_{\mathrm{delD}}(d_D,d'_D|D)\mathbb{I}[(\tau,\mathbf{s}^D_{d_D+1:l_D-d'_D})\sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^*=2 \text{ and } u_3^*=3, \end{aligned} \tag{A14}$$

$$\begin{aligned}\mathcal{D}(D)^{x_2}{}_{x_3}(\sigma) &= \sum_{d'_D} P_{\text{delD}}(d_D, d'_D|D)\mathbb{I}[s^D_{l_D-d'_D} = \sigma]\mathbb{I}[\mathbf{s}^D_{d_D+1:l_D-d'_D} \sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^* = 3 \text{ and } u_3^* = 1,\\ \mathcal{D}(D)^{x_2}{}_{x_3}(\sigma) &= \sum_{d'_D} P_{\text{delD}}(d_D, d'_D|D)\mathbb{I}[(\mathbf{s}^D_{d_D+1:l_D-d'_D}, \sigma) \sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^* = 3 \text{ and } u_3^* = 2,\\ \mathcal{D}(D)^{x_2}{}_{x_3} &= \sum_{d'_D} P_{\text{delD}}(d_D, d'_D|D)\mathbb{I}[\mathbf{s}^D_{d_D+1:l_D-d'_D} \sim \mathbf{a}_{i_2:i_3}] \quad \text{if } u_2^* = 3 \text{ and } u_3^* = 3\end{aligned} \tag{A15}$$

where $d_D = l_D - (x_3 - x_2) - d'_D$

d. $\mathcal{N}^{x_3}{}_{x_4}$

Contribution from the non-templated N2 insertions (DJ junction). $\mathcal{N}^{x_3}{}_{x_4}$ is defined as the product of transfer matrices, and can be a 1x1, 1x4, 4x1, or 4x4 matrix depending on u_3^* and u_4^*. The transfer matrices are defined by the summed contributions of the Markov insertion model of all codons consistent with the amino acid a (thus summations are over nucleotides y, y_1, and y_2 to consider all allowed codons):

$$T'_a(\tau,\sigma) = \sum_{(\sigma,y_2,y_1)\sim a} S_{\text{DJ}}(\sigma|y_2)S_{\text{DJ}}(y_2|y_1)S_{\text{DJ}}(y_1|\tau) \tag{A16}$$

$$F'_a(\tau,\sigma) = S_{\text{DJ}}(\sigma|\tau)\mathbb{I}[\exists\sigma',\sigma'' \text{ s.t. } (\sigma'',\sigma',\sigma) \sim a] \tag{A17}$$

$$D'_a(\tau,\sigma) = \sum_{(\sigma,y_2,y_1)\sim a} S_{\text{DJ}}(y_2|y_1)S_{\text{DJ}}(y_1|\tau) \tag{A18}$$

$$lT'_a(\tau,\sigma) = \sum_{(\sigma,y,\tau)\sim a} S_{\text{DJ}}(\tau|y)q_0(y) \tag{A19}$$

$$lD'_a(\tau,\sigma) = \sum_{(\sigma,y,\tau)\sim a} q_0(y) \tag{A20}$$

If $i_4 > i_3$:

$$\mathcal{N}^{x_3}{}_{x_4} = P_{\text{insDJ}}(x_4 - x_3){L'}^{u_3^*}_{a_{i_3}} T'_{a_{i_3+1}} \dots T'_{a_{i_4-1}} {R'}^{u_4^*}_{a_{i_4}} \tag{A21}$$

where:

$${L'}^{u_3^*}_{a_{i_3}} = \begin{cases} F'_{a_{i_3}} & \text{if } u_3^* = 1\\ D'_{a_{i_3}} & \text{if } u_3^* = 2\\ T'_{a_{i_3}}\vec{1} & \text{if } u_3^* = 3\end{cases} \qquad \text{and} \qquad {R'}^{u_4^*}_{a_{i_4}} = \begin{cases} lT'_{a_{i_4}} & \text{if } u_4^* = 1\\ diag(q_0) & \text{if } u_4^* = 2\\ S_{\text{DJ}}^{-1}q_0 & \text{if } u_4^* = 3\end{cases} \tag{A22}$$

If $i_3 = i_4$:

$$\mathcal{N}^{x_3}{}_{x_4} = P_{\text{insDJ}}(x_4 - x_3) \times \begin{array}{c|ccc} & u_4^* = 1 & u_4^* = 2 & u_4^* = 3\\ \hline u_3^* = 1 & \mathbb{1} & lD'_{a_{i_3}} & lT'_{a_{i_3}}\vec{1}\\ u_3^* = 2 & 0 & \mathbb{1} & diag(q_0)\vec{1}\\ u_3^* = 3 & 0 & 0 & 1\end{array} \tag{A23}$$

e. $\mathcal{J}(D)^{x_4}$

Contribution from the templated J genes. $\mathcal{J}(D)^{x_4}$ can be a 1x1 or 4x1 matrix depending on u_4^*. $\mathbf{s}^J$ is the sequence of the J germline gene (read $5'$ to $3'$) and l_J gives the length of the sequence up to the conserved residue (generally

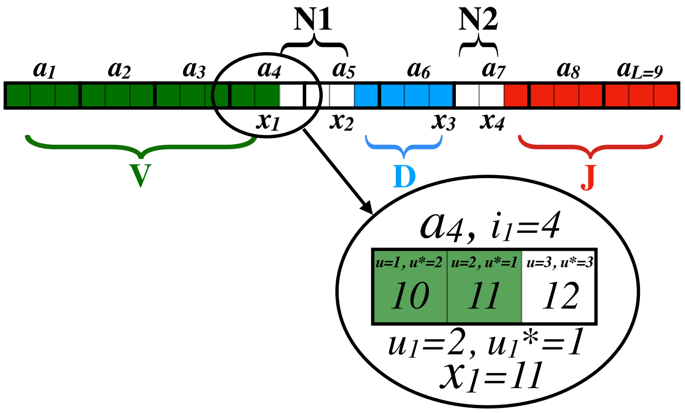

FIG. S1: Schematic of the partitioning of an amino acid sequence into sections for the purpose of constructing the probability matrices underlying the dynamic programming method for computing its net generation probability. The indexing conventions are also highlighted.

either F or W).

$$\begin{aligned}\mathcal{J}(D)^{x_4}(\tau) &= \sum_J P_{\mathrm{D,J}}(DJ)P_{\mathrm{delJ}}(d_J|J)\mathbb{I}(s^J_{d_J+1}=\tau)\mathbb{I}(\mathbf{s}^J_{d_J+1:l_J}\sim \mathbf{a}_{i_4:L}) \quad \text{if } u_4^*=1,\\ \mathcal{J}(D)^{x_4}(\tau) &= \sum_J P_{\mathrm{D,J}}(DJ)P_{\mathrm{delJ}}(d_J|J)\mathbb{I}((\tau,\mathbf{s}^J_{d_J+1:l_J})\sim \mathbf{a}_{i_4:L}) \quad \text{if } u_4^*=2,\\ \mathcal{J}(D)^{x_4} &= \sum_J P_{\mathrm{DJ}}(D,J)P_{\mathrm{delJ}}(d_J|J)\mathbb{I}(\mathbf{s}^J_{d_J+1:l_J})\sim \mathbf{a}_{i_4:L}) \quad \text{if } u_4^*=3.\end{aligned} \tag{A24}$$

where $dJ = l_J - 3L - x_4 - 1$

Appendix B: VJ recombination

The model used for VJ recombination is quite similar to the model for VDJ recombination with the main differences being the lack of a D segment and an N2 insertion segment. However, a strong correlation between V and J templates is observed in the TRA chain, so we include a joint V, J distribution to allow for this correlation. Due to this similarity, the algorithm used to compute P$_{\mathrm{gen}}$ is very similar. The VJ generative model is:

$$P^{\mathrm{rec}}_{\mathrm{gen}}(E) = P_{\mathrm{VJ}}(V,J)P_{\mathrm{delV}}(d_V|V)P_{\mathrm{delJ}}(d_J|J) \times P_{\mathrm{insVJ}}(\ell_{\mathrm{VJ}})p_0(m_1)\left[\prod_{i=2}^{\ell_{VJ}} S_{\mathrm{VJ}}(m_i|m_{i-1})\right] \tag{B1}$$

with nucleotide and amino acid P_{gen}s being defined the same as for the VDJ recombination model (Eq A2). The dynamic programing algorithm also has a similar form to Eq A3, and can be summarized as (retaining all notation conventions from before):

$$P_{\mathrm{gen}}(a_1,\ldots,a_L) = \sum_{x_1,x_2}\sum_J \mathcal{V}(J)_{x_1}\mathcal{M}^{x_1}{}_{x_2}\mathcal{J}(J)^{x_2} \tag{B2}$$

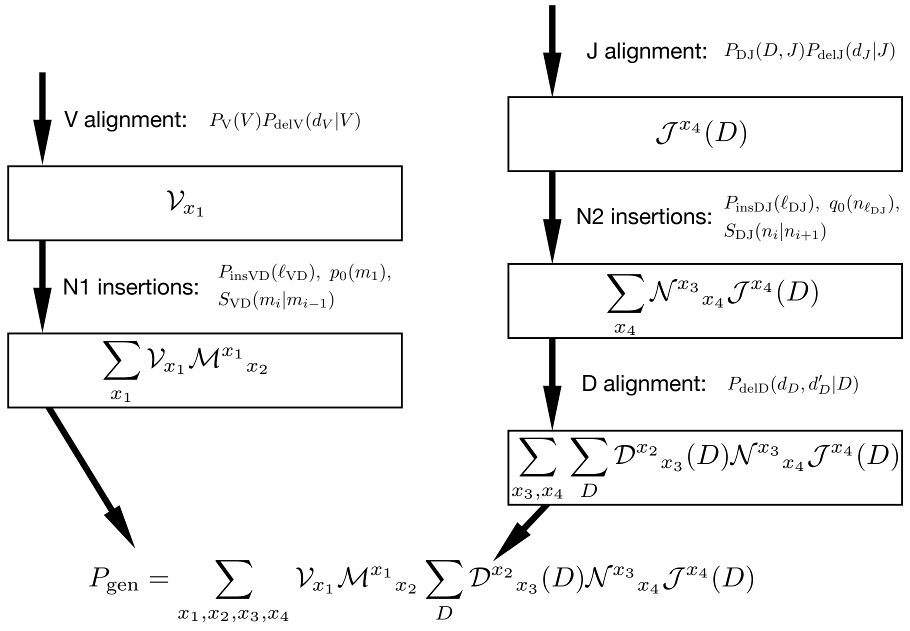
J alignment: $P_{\mathrm{DJ}}(D,J)P_{\mathrm{delJ}}(d_J|J)$
V alignment: $P_{\mathrm{V}}(V)P_{\mathrm{delV}}(d_V|V)$
$\mathcal{J}^{x_4}(D)$
$\mathcal{V}_{x_1}$
N2 insertions: $P_{\mathrm{insDJ}}(\ell_{\mathrm{DJ}}),\ q_0(n_{\ell_{\mathrm{DJ}}}),$ $S_{\mathrm{DJ}}(n_i|n_{i+1})$
N1 insertions: $P_{\mathrm{insVD}}(\ell_{\mathrm{VD}}),\ p_0(m_1),$ $S_{\mathrm{VD}}(m_i|m_{i-1})$
$\sum_{x_4}\mathcal{N}^{x_3}{}_{x_4}\mathcal{J}^{x_4}(D)$
$\sum_{x_1}\mathcal{V}_{x_1}\mathcal{M}^{x_1}{}_{x_2}$
D alignment: $P_{\mathrm{delD}}(d_D,d'_D|D)$
$\sum_{x_3,x_4}\sum_{D}\mathcal{D}^{x_2}{}_{x_3}(D)\mathcal{N}^{x_3}{}_{x_4}\mathcal{J}^{x_4}(D)$
$P_{\mathrm{gen}}=\sum_{x_1,x_2,x_3,x_4}\mathcal{V}_{x_1}\mathcal{M}^{x_1}{}_{x_2}\sum_{D}\mathcal{D}^{x_2}{}_{x_3}(D)\mathcal{N}^{x_3}{}_{x_4}\mathcal{J}^{x_4}(D)$

FIG. S2: Schematic of the OLGA VDJ algorithm implementation breakdown. Each of the 5 segments (V, N1, D, N2, J), and their associated model contributions, are considered from the edges of the CDR3 towards the inside. This is done both from the left side (V, N1) and the right side (D, N2, J) of the read to efficiently account for the correlations for the D and J genes. Including inner segments (N2, D, N2) requires summing over an index, indicating that all possible allowed start and end positions of the segment are considered.

f. $\mathcal{V}(\mathcal{J})_{x_1}$

Contribution from the templated V genes.

$$\begin{aligned}\mathcal{V}(J)_{x_1}(\sigma) &= \sum_V P_{\mathrm{VJ}}(V,J)P_{\mathrm{delV}}(l_V-x_1|V)\mathbb{I}(s^V_{x_1}=\sigma)\mathbb{I}(\mathbf{s}^V_{1:x_1}\sim\mathbf{a}_{1:i_1}) \quad \text{if } u_1=1,\\ \mathcal{V}(J)_{x_1}(\sigma) &= \sum_V P_{\mathrm{VJ}}(V,J)P_{\mathrm{delV}}(l_V-x_1|V)\mathbb{I}((\mathbf{s}^V_{1:x_1},\sigma)\sim\mathbf{a}_{1:i_1}) \quad \text{if } u_1=2,\\ \mathcal{V}(J)_{x_1} &= \sum_V P_{\mathrm{VJ}}(V,J)P_{\mathrm{delV}}(l_V-x_1|V)\mathbb{I}(\mathbf{s}^V_{1:x_1}\sim\mathbf{a}_{1:i_1}) \quad \text{if } u_1=3.\end{aligned} \tag{B3}$$

g. $\mathcal{M}^{x_1}{}_{x_2}$

Contribution from the non-templated N insertions (VJ junction). $\mathcal{M}^{x_1}{}_{x_2}$ is identical to the definition of $\mathcal{M}^{x_1}{}_{x_2}$ from the VDJ algorithm (except using the parameters $S_{\mathrm{VJ}}, P_{\mathrm{insVJ}}$, and p_0 from a VJ recombination model).

h. $\mathcal{J}(J)^{x_2}$

Contribution from the templated J genes.

$$\begin{aligned}\mathcal{J}(J)^{x_2}(\tau) &= P_{\mathrm{delJ}}(d_J|J)\mathbb{I}(s^J_{d_J+1}=\tau)\mathbb{I}(\mathbf{s}^J_{d_J+1:l_J}\sim\mathbf{a}_{i_2:L}) \quad \text{if } u_2^*=1,\\ \mathcal{J}(J)^{x_2}(\tau) &= P_{\mathrm{delJ}}(d_J|J)\mathbb{I}((\tau,\mathbf{s}^J_{d_J+1:l_J})\sim\mathbf{a}_{i_2:L}) \quad \text{if } u_2^*=2,\\ \mathcal{J}(J)^{x_2} &= P_{\mathrm{delJ}}(d_J|J)\mathbb{I}(\mathbf{s}^J_{d_J+1:l_J})\sim\mathbf{a}_{i_2:L}) \quad \text{if } u_2^*=3.\end{aligned} \tag{B4}$$

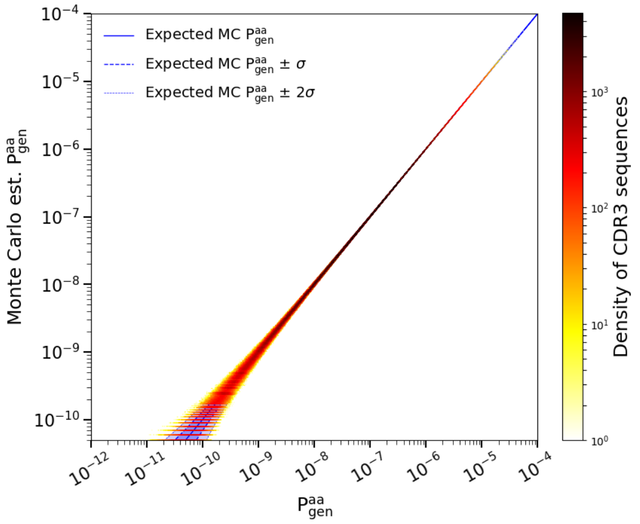

FIG. S3: Monte Carlo estimate of the generation probability of amino acid human TRA CDR3 sequences, $P^{\text{aa}}_{\text{gen}}$, versus OLGA's calculation. The horizontal lines at the lower left of the plot represent CDR3s that were generated once, twice, etc, in the MC sample. The one- and two-sigma curves display the deviations from exact equality between simulated and computed P_{gen} to be expected on the basis of Poisson statistics.

where $dJ = l_J - 3L - x - 1$

This algorithm is validated in the same manner to the VDJ algorithm, i.e. comparing to Monte Carlo (MC) estimation (Fig S3).

Appendix C: Dependence on model parameters and structure

In order to efficiently compute the summation in Eq. A3 the summations of the model contributions from each of the 5 segments of a CDR3 (V genomic, N1 insertions, D genomic, N2 insertions, and J genomic) are performed in a specific order (summarized in Fig S2). Specifically, we start at the left and right ends of the CDR3 read and move inwards, summing over positional indices at each step. As the D and J segments are correlated, it is useful to consider the V and N1 contributions separately from the D, N2, and J and to do the final summation over the index x_2 after the D, N2, and J components are summed over all D alleles (notice the D dependencies in Fig S2). This breakdown is useful to highlight the most computationally intensive steps: N2 insertions and the D alignment. These steps (along with the N1 insertions) require considering that the associated segment could begin and end at each allowed position. This is mathematically seen as the summation over positions and computing a matrix indexed by two indices, leading to an $O(L^2)$ complexity. The N2 insertions and D alignments are further aggravated due to model correlations between the D and J genes requiring repeating the steps for N2 insertions and D alignment for each D allele. The runtime of OLGA is thus most sensitive to the maximum number of N2 insertions and the length and number of the D alleles. The effects of varying these parameters is best illustrated by comparing runtimes for mouse TRB, human TRB, and human IGH models (Table S1). In a similar fashion, the most computationally intensive step of computing P_{gen} of a VJ model (e.g. human TRA) is the insertion step, and due to correlations between the V and J genes this is repeated for each J allele in a similar fashion as the D alleles However, as the J region of a human TRA is fairly large, many of these J genes can be excluded from alignment (if they contribute 0 probability), yielding the much faster computation rate of 184 seqs/CPU second.

TABLE S1: Model comparison

Species/Chain	max insertions	# D alleles	Average computation speed
Mouse TRB	11	2	70.4 seqs/CPU second
Human TRB	30	3	35.6 seqs/CPU second
Human IGH	60	35	2.05 seqs/CPU second

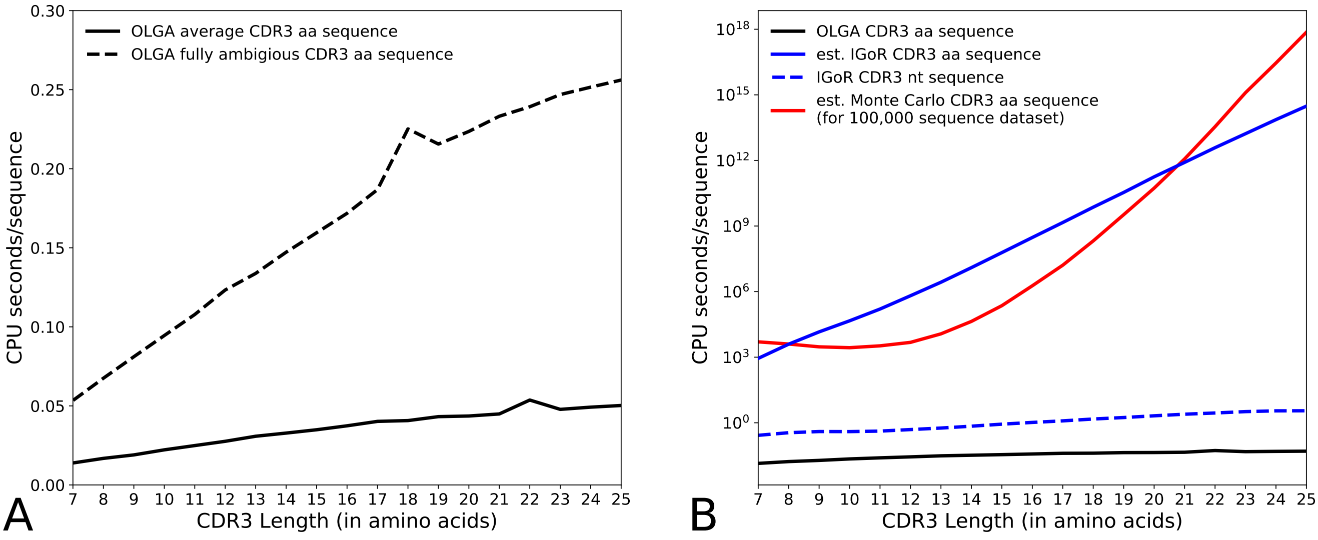

FIG. S4: A) Computational performance of OLGA as a function of CDR3 length. We compare performance averaged over a sample of human TRB amino acid CDR3 sequences to the worst case scenario of CDR3 sequences composed of fully ambiguous amino acids X. In both cases the time for a single sequence increases roughly linearly (i.e. less than the algorithmic worst case of $O(L^2)$). B) Computational performance of different P_{gen} methods as a function of CDR3 length (log scale). The IGoR and OLGA runtimes are determined by running over the same statistical sample of human TRB sequences. OLGA runs over the translated amino acid CDR3 sequences while IGoR runs over nucleotide CDR3 (dashed blue line). In order to compare OLGA to how long it would take IGoR to compute P_{gen} of amino acid CDR3s we estimate by multiplying the IGoR runtime of single nucleotide sequences (dashed blue line) by the number of nucleotide sequences that translate to the given amino acid sequence (yielding the solid blue line). Monte Carlo runtime is estimated for a dataset of 100,000 sequences with an estimated coverage of 66% of sequences having at least one count. OLGA vastly outperforms both direct enumeration (est. IGoR) and Monte Carlo.

Appendix D: Timing, performance, model dependence

In order to analyze OLGA's computational performance as a function of CDR3 length, and to compare to other hypothetical methods, we use the human TRB model as an example.

As discussed in the previous section, the most computationally intensive steps of OLGA (N1, N2, and D) require at most $O(L^2)$ operations. In practice, OLGA's scaling of the computation speed as a function of CDR3 length, even for the worst case sequences, i.e. fully ambiguous amino acids of a given length, is closer to linear in the relevant regime due to the finite parameterization of the model (maximum number of insertions, maximum size of D sequences, etc). This is shown in Fig S4A.

We also compare OLGA to runtimes of IGoR (i.e. direct enumeration of recombination events) and a hypothetical Monte Carlo computation (Fig S4). As we will explain, neither the IGoR nor the MC are precise comparisons to OLGA, yet OLGA is faster than either.

The IGoR runtimes are for nucleotide sequences not amino acid sequences. In order for IGoR to compute the P_{gen} of an amino acid sequence, it would need to compute and sum the P_{gen} of each nucleotide sequence that codes for the amino acid sequence. These sequences can be enumerated for extremely short CDR3 lengths, however the number explodes exponentially in CDR3 length. Even for a CDR3 length of 4, by enumerating all nucleotide sequences for an amino acid sequence IGoR computes 0.33 seqs/CPU second compared to the 122 seqs/CPU second for OLGA. For longer CDR3 lengths we approximate how long IGoR would take by computing the average number of CDR3 nucleotide sequences per CDR3 amino acid sequence for a given length. OLGA not only heavily outperforms this exponential blowup, but actually outperforms IGoR when IGoR is computing a *single* nucleotide sequence of a given amino acid sequence.

The Monte Carlo runtime estimate comes from the setup of estimating the P_{gen} of 100,000 sequences. These P_{gen} would be estimated by simulating enough recombination events such that 66% of CDR3 sequences of a given length would be expected to have at least one count. There is a CDR3 length scaling due to the trend that shorter sequences tend to have higher P_{gen} (Fig S5B). The P_{gen} estimated using this methodology will be extremely noisy (Poisson noise on the expected number of counts) and not even give reliable estimates for many sequences.

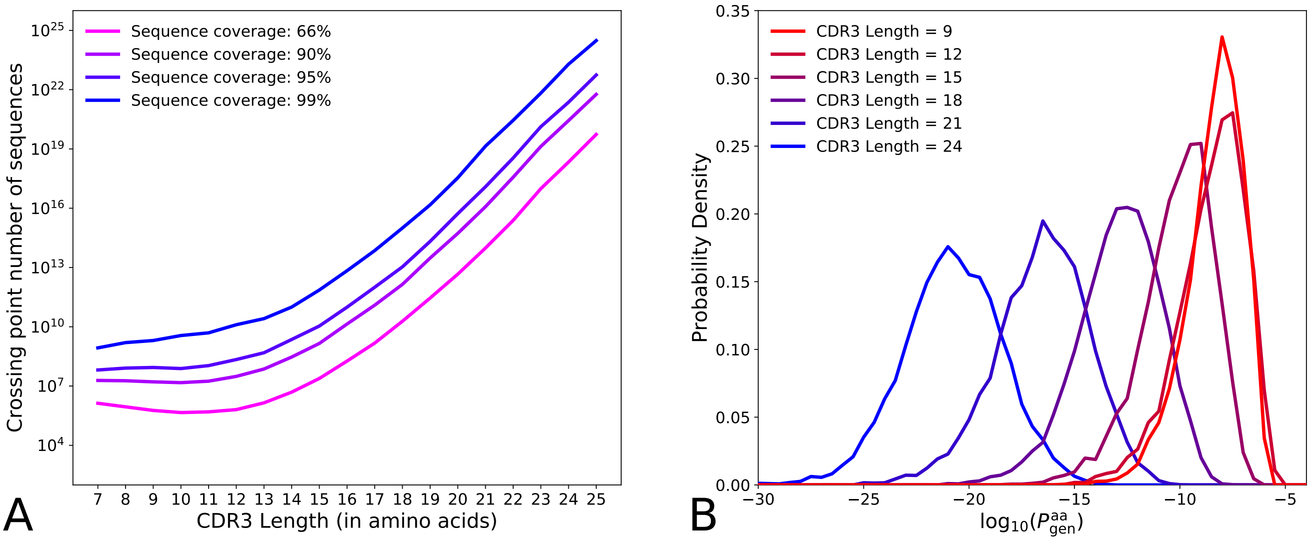

FIG. S5: A) The runtime of Mont Carlo P_{gen} estimation scales as $1/P_{\mathrm{gen}}$ while OLGA will scale with the number of sequences. This predicts a number of sequences for the 'crossing point' where the runtime of Monte Carlo P_{gen} estimation is comparable to OLGA P_{gen} for sequences with P_{gen} above some cutoff. For datasets with more sequences than these curves, Monte Carlo estimation may be faster (depending on the level of Poisson noise considered tolerable), while below these curves OLGA is always faster. We plot this as a function of CDR3 length where the P_{gen} cutoffs are determined to ensure that on average some fraction (66%, 90%, 95%, and 99%) of the sequences at that length get covered by the Monte Carlo estimation. B) $\log_{10}(P_{\mathrm{gen}})$ probability density distributions for a few examples of CDR3 lengths. These curves are used to determine the MC P_{gen} cutoffs per CDR3 length by determining, for a given curve, when the area under the curve and right of a P_{gen} cutoff matches the sequence coverage fraction.

It is true that the computation time for MC estimates scale as $1/P_{\mathrm{gen}}$ and not with the number of sequences. Thus, there is a hypothetical number of sequences when MC is faster than OLGA if we are willing to accept noisy estimates and to entirely miss some fraction of the CDR3s. This 'crossing point' number of sequences is plotted in Fig S5A and corresponds to completely unrealistic numbers of sequences, highlighting the fact that OLGA will not only give a more reliable P_{gen}, even for very unlikely sequences, but is also much faster than MC even for short, high P_{gen}, sequences. So, even overlooking the drawbacks and imprecision of MC estimation, for plausible sized datasets OLGA is still dramatically faster than MC.

Appendix E: Generation probability distributions from RNA-derived repertoires

The analyses described in the main text were mostly concerned with datasets derived by sequencing the genomic DNA contained in a sample of immune cells to directly obtain sequences of the rearranged TCR genes. Immune repertoires can alternatively be obtained by sequencing the mRNA expressed from the same genes, and many such RNA-based data sets exist. Given a TCR sequence, OLGA evaluates the probability of the primitive recombination event (or events) that must have occurred to create the initial T cell carrying that sequence, and the applicability of OLGA is independent of how the sequence was obtained (i.e. from DNA or RNA sequencing). OLGA relies on the availability of a suitable recombination model but that model is thought to vary very little with time (and disease status) for each individual subject and only moderately from individual to individual in a given species. The probability that a given sequence, once generated in a primitive event, will be captured in a sequencing experiment is at best roughly constant across sequences, and may vary substantially between different capture protocols.

For these reasons, it is interesting to investigate how these generation probability distributions vary across CDR3 repertoires obtained using different sequencing protocols in different biological contexts. In Fig. S6 we plot the results of running OLGA on a few recently published human TRB repertoires that were obtained using RNA sequencing. These samples comprise a study of patients with glioblastoma disease (Sims *et al.* [38]), a study of patients with Crohn's disease and ulcerative colitis (Wu *et al.* [47]), and a comprehensive study of the dynamics of TCRs in healthy individuals (Wang *et al.* [44]). Fig. S6 shows the generation probability distribution of data sets from these three sources, for comparison plotted together with the distribution obtained from DNA sequencing of the large human

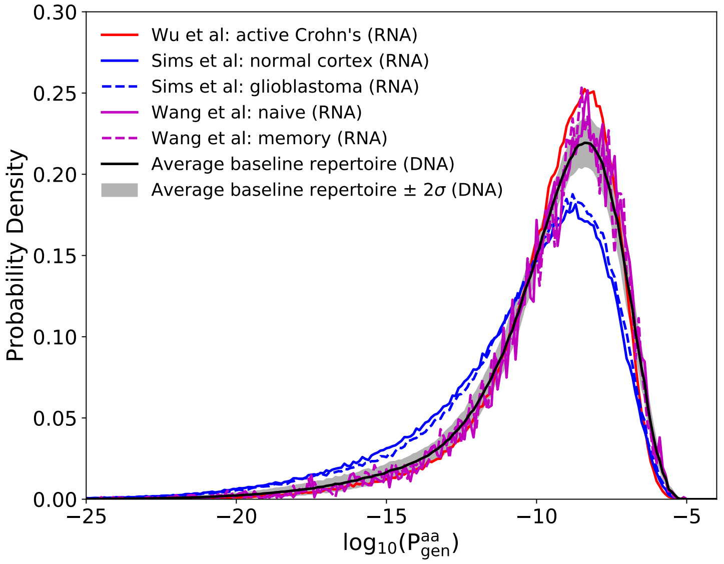

FIG. S6: Generation probability distributions for TRB CDR3 sequences taken from three different sources (Sims *et al.* [38], Wu *et al.* [47], Wang *et al.* [44]), compared to DNA RepSeq data from Emerson *et al.* [9] (black curve with standard deviation), using a model inferred from [9]. All distributions have approximately the same shape, with a slight bias in the data from Sims *et al.* [38], indicating how robust is the distribution. Data from Sims *et al.* are identified in the SI of [38] as IDs N01 (normal cortex) and G10 (glioblastoma). Data from Wang *et al.* is identified by Short Read Archive (SRA) accession numbers SRR030702 (naive) and SRS007450 (memory). See also Fig. S7.

sample of Emerson *et al.* [9]. As can be seen, two of the three RNA data sets give results quite consistent with the DNA-based results. The glioblastoma data (Sims *et al.* [38]) gives a distribution broadly similar to the other three, but with a systematic shift to higher frequency of occurrence of lower generation probability sequences. We do not know whether or not this difference is biologically significant, or an artifact of the used protocol. The difference does not seem to be due to sampling depth, as can be seen in Fig. S7, where multiple samples from Sims *et al.* [38] are plotted: the distributions derived from smaller samples are noisier than, but statistically consistent with, the distributions based on the largest samples.

Appendix F: Cross-species P_{gen}

TCR sequence repertoires are different in detail between species, both because the genomic templates differ and because of differences in the parameters of the recombination process itself. As a result, there are clear interspecies differences in CDR3 length distribution and amino acid composition. Nevertheless, the TRB CDR3 regions of different vertebrate species have the same overall structure and the same conserved residues at the two ends of the CDR3. As a result, a CDR3 from one species usually has a non-zero probability to be produced within a different species, a fact of some interest in the context of studies of cross-species sharing of T cell types. We explored this concept with OLGA by feeding TRB CDR3s produced by the human generation model to a mouse generation model and vice versa. The resulting generation probability distributions are plotted in Fig. S8.

The sequences that are produced in one species have substantially lower probability of being generated in the other species (Fig. S8). The effect is strongest for finding human sequences in a mouse repertoire (compare the black dashed curve with red solid curve): the bulk of the human sequences have extremely low generation probabilities in the mouse model. The effect is less strong for finding mouse sequences in a human repertoire (compare the red dashed curve with the black solid curve): a small fraction of the mouse sequences have generation probabilities that are as high as the highest generation probabilities of human sequences. The results are not symmetric - while the mouse TRBs processed using the human model have a distinct bi-model distribution, the human TRBs have a very flat and low mouse generation probabilities. Furthermore, mouse TRB sequences always have a non-zero probability of being generated in a human TRB context, however 27.4% of human TRB sequences have $P_{\text{gen}} = 0$ as defined by a mouse TRB model. This asymmetry is primarily due to differences in the insertion profiles (humans may have many more

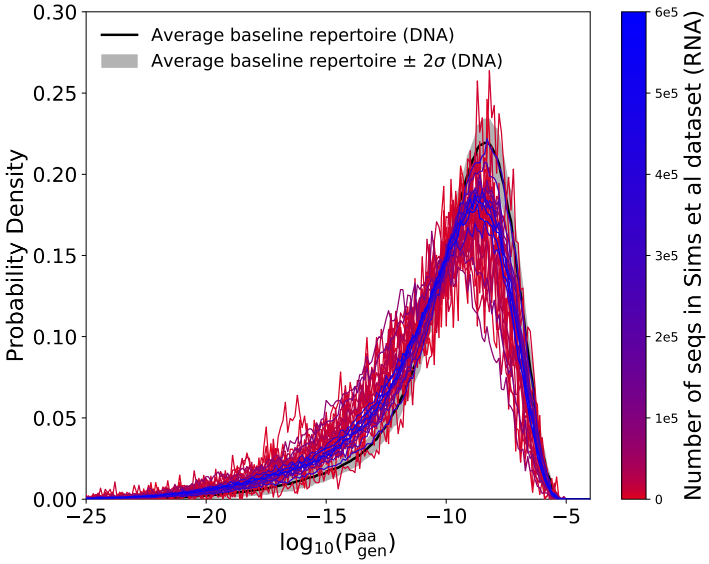

FIG. S7: Generation probability distributions for different samples from Sims *et al.* [38], compared to DNA RepSeq data from Emerson *et al.* [9] (black curve with standard deviation), using a model inferred from [9]. Color indicate sample size: larger datasets are blue, while small ones are red. The only effect of decreasing the sample size is increasing the noise, but the shape stays the same. All TRB datasets from the study are plotted.

inserted N1 and N2 nucleotides) and by extension CDR3 length. Nonetheless, these results suggest that there will be a non-negligible amount of sharing, entirely due to chance statistics, of CDR3 sequences between mouse and human repertoires. A more detailed view of this structure can be seen in a scatter plot of the generation probabilities between the two (Fig. S9). While there are many sequences with high generation probabilities in both the actual generative model and the cross species model, the cross species generation probabilities are much more variable and span many orders of magnitude, without much correlation to the correct species model.

Appendix G: Generation probability distributions from additional pathogen response datasets

In the main text, we displayed the distribution of generative probabilities for T cells known to respond to various pathogens, and even specific epitopes of particular pathogens. The T cell sequences are taken from databases that compile results from multiple experiments. We found that these distributions were, within statistical noise, indistinguishable from the background $P_{\rm gen}$ distribution of PBMCs drawn from the blood. In other words, it would seem that there is no correlation between ease of generation of a T cell and its likelihood to respond to a particular pathogen or epitope. A defect of this analysis is that the database agglomerates sequences from different experimental protocols, so that there is no way of knowing what biases might have affected the inclusion of any given sequence in the database. Obviously, it would be better to do a single well-controlled experiment in which T cells from a single donor are stimulated to expand by selected pathogens, and the expanded T cells sequenced. Such an experiment was reported by Becattini *et al.* [1] several years ago. In their experiment, CD4+ helper T cells were separated from peripheral blood samples, autologous monocytes from the same samples were incubated with three different pathogens (a fungus, a bacterium, and a toxin) in order to load pathogen epitopes, and helper T cell subsamples (typically containing several million T cells, and hundreds of thousands of clonotypes) were incubated with the prepared monocytes (this was done independently for samples from several donors). The T cells in the various samples that had proliferated under this treatment were separated out (typically yielding millions of cells) and their TRB sequences obtained using the Adaptive Biotechnology genomic DNA protocol. The result is a collection of lists of clones (defined by CDR3 amino acid sequence) from the blood of individual donors that can be said to have expanded under stimulation by the three different pathogens. The responses obtained in this way are quite polyclonal, with a few thousand clonotypes in each list of responding clones (the polyclonality perhaps being due to the fact that stimulation is with preparations of whole pathogens, as opposed to particular pathogen peptides). The $P_{\rm gen}$ distributions of the pathogen–responsive clones for different individuals and pathogens are plotted in Fig. S10. They are indistinguishable from the background

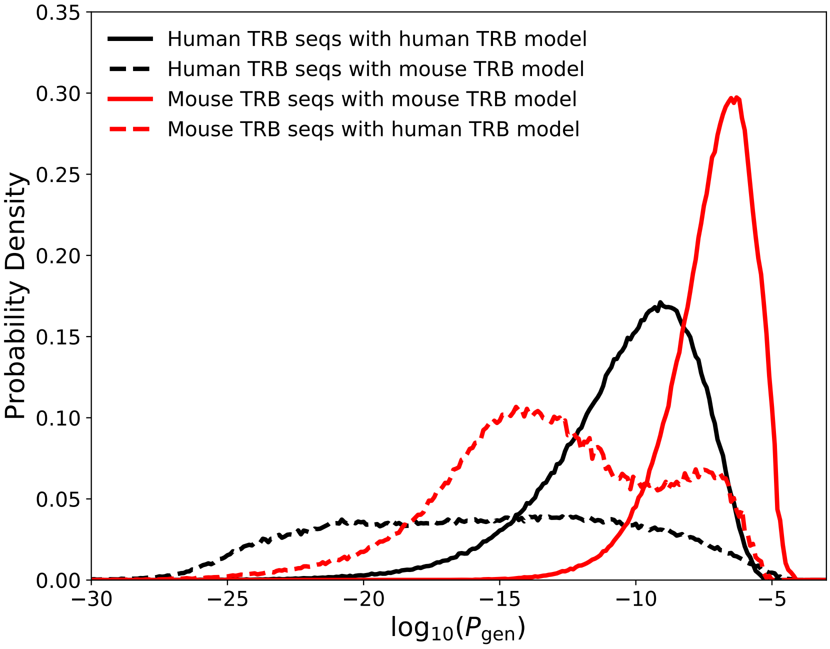

FIG. S8: Probability densities of $\log_{10}(P_{\text{gen}})$ for sequences generated from mouse TRB and human TRB models. The P_{gen} of a sequence is computed using either a mouse TRB model or a human TRB model depending on the curve. Models are based on data from Emerson *et al.* [9] for human TRB and Sethna *et al.* [36] for mouse TRB.

P_{gen} distribution derived from blood samples of healthy individuals, which is also plotted (along with its two-sigma variance across a population of individuals) for reference. These data further strengthen the conclusion that pathogen response activity is uncorrelated with P_{gen}.

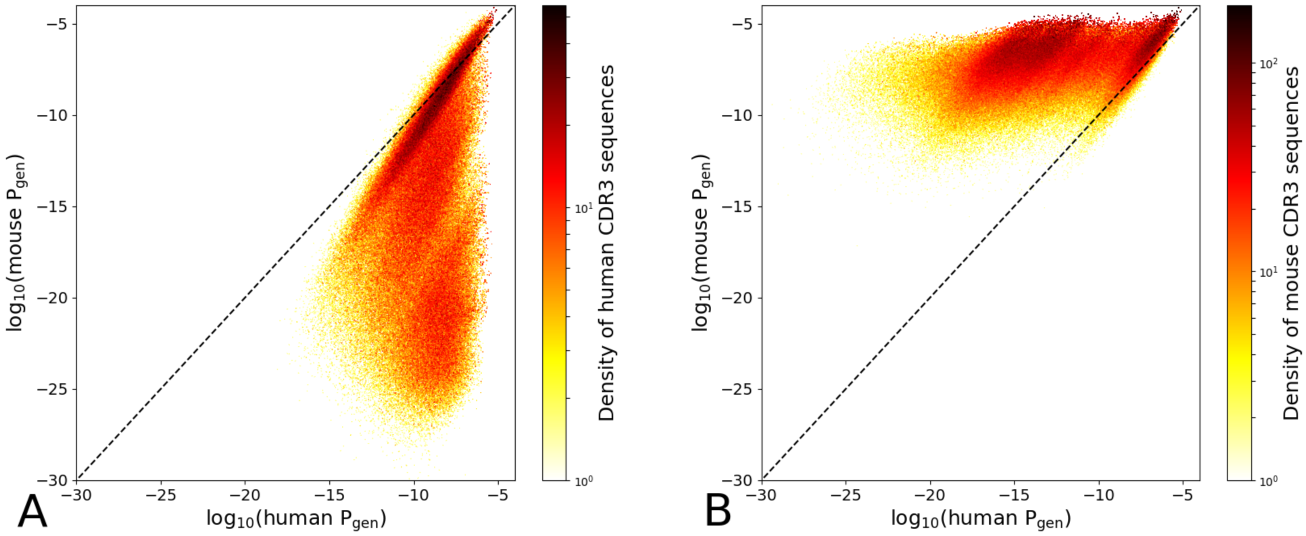

FIG. S9: Scatter plots of CDR3 sequence repertoires across their P_{gen} values as determined by a human TRB model or a mouse TRB model. The sequence repertoires are Monte Carlo samples from A) a human TRB model or B) a mouse TRB model. Projections of the scatter plots onto the two axes reproduce the distributions displayed in Fig. S8. Models are based on data from Emerson *et al.* [9] for human TRB and Sethna *et al.* [36] for mouse TRB.

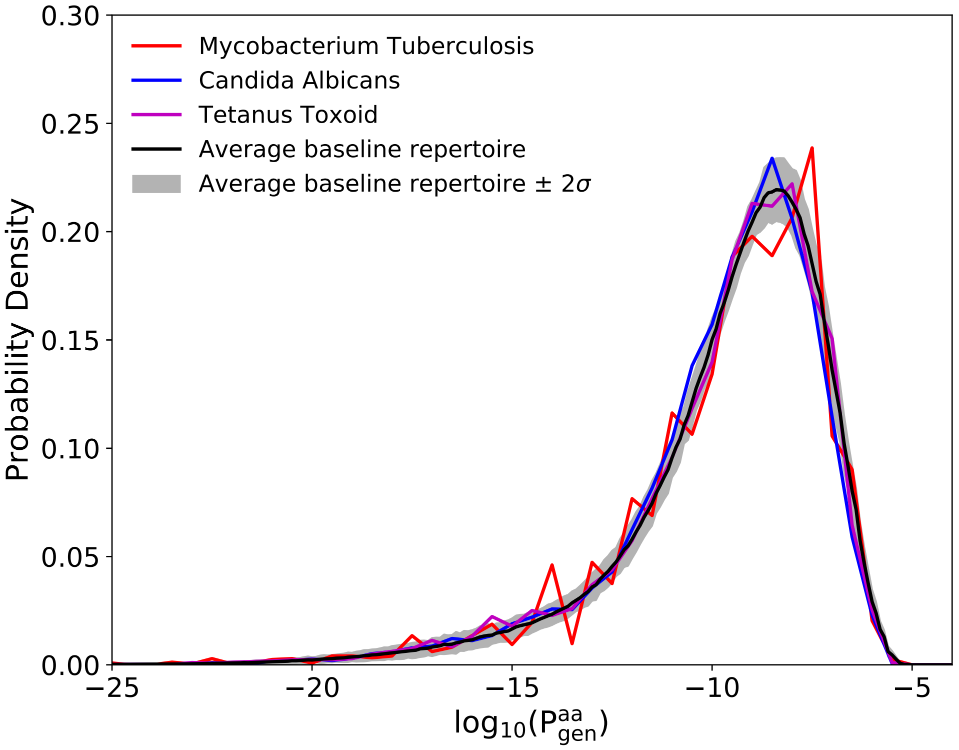

FIG. S10: $P_{\rm gen}$ distributions for human CD4+ T cell repertoires that have been incubated with three different pathogens (Becattini *et al.* [1]): the fungus Candida Albicans (CA), the bacterium Mycobacterium Tuberculosis (MT), and a toxin protein Tetanus Toxoid (TT). For comparison, the background distribution from human peripheral blood TRB sequences from Emerson *et al.* [9] (with its two sigma variation across multiple individuals) is also plotted. The plotted curves are averages over data from individual donors. The sizes of the responsive T cell repertoires are quite variable: the CA dataset has 39934 clonotypes from 5 donors, the TT dataset has 26573 clonotypes from 4 donors, and the MT dataset has 5082 clonotypes from 2 donors. The generation model was infered from Emerson *et al.* [9].