Bioinformatics, 32(13), 2016, 1943–1951
doi: 10.1093/bioinformatics/btw112
Advance Access Publication Date: 7 March 2016
Original Paper

Sequence analysis

repgenHMM: a dynamic programming tool to infer the rules of immune receptor generation from sequence data

Yuval Elhanati[1],*, Quentin Marcou[1], Thierry Mora[2],* and Aleksandra M. Walczak[1],*

[1]Laboratoire de physique théorique, CNRS, UPMC and Ecole normale supérieure, Paris, France and [2]Laboratoire de physique statistique, CNRS, UPMC and Ecole normale supérieure, Paris, France

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on October 30, 2015; revised on February 12, 2016; accepted on February 20, 2016

Abstract

Motivation: The diversity of the immune repertoire is initially generated by random rearrangements of the receptor gene during early T and B cell development. Rearrangement scenarios are composed of random events—choices of gene templates, base pair deletions and insertions—described by probability distributions. Not all scenarios are equally likely, and the same receptor sequence may be obtained in several different ways. Quantifying the distribution of these rearrangements is an essential baseline for studying the immune system diversity. Inferring the properties of the distributions from receptor sequences is a computationally hard problem, requiring enumerating every possible scenario for every sampled receptor sequence.

Results: We present a Hidden Markov model, which accounts for all plausible scenarios that can generate the receptor sequences. We developed and implemented a method based on the Baum–Welch algorithm that can efficiently infer the parameters for the different events of the rearrangement process. We tested our software tool on sequence data for both the alpha and beta chains of the T cell receptor. To test the validity of our algorithm, we also generated synthetic sequences produced by a known model, and confirmed that its parameters could be accurately inferred back from the sequences. The inferred model can be used to generate synthetic sequences, to calculate the probability of generation of any receptor sequence, as well as the theoretical diversity of the repertoire. We estimate this diversity to be $\approx 10^{23}$ for human T cells. The model gives a baseline to investigate the selection and dynamics of immune repertoires.

Availability and implementation: Source code and sample sequence files are available at https://bitbucket.org/yuvalel/repgenhmm/downloads.

Contact: elhanati@lpt.ens.fr or tmora@lps.ens.fr or awalczak@lpt.ens.fr

1 Introduction

The ability of the adaptive immune system to identify a wide range of threats rests upon the diversity of its lymphocyte receptors, which together make up the immune repertoire. Each such receptor can bind specifically to antigenic molecules, and initiate an immune response against the threat. T cell receptors (TCR) are composed of two protein chains, called alpha and beta. B cell receptors (BCR) share a very similar structure, with a light chain and heavy chain playing the same role. Each chain is produced according to the same process of V(D)J rearrangement. In each new cell and for each of the two chains, two germline segments for alpha chains (Vα and Jα genes), or three segments for beta chains (Vβ, Dβ and Jβ genes), are assembled together to form the recombined gene coding for the chain. In addition, at the junctions where the segments are joined, the ends of the segments are trimmed, and random nucleotides are inserted (see Fig. 1a for a diagram describing the alpha chain rearrangement process). This process creates a large initial diversity of possible receptors, which are later selected according to their recognition functionality. An important property of this process is that it is redundant, as many different V(D)J rearrangements may lead to the exact same sequence. It is thus impossible to unambiguously reconstruct the scenario from the sequence alone, a problem that is aggravated by sequencing errors.

The rearrangement process is random, as is each of the elements composing it—choice of germline segments, number of deleted nucleotides, number and identity of insertions. Since the rearrangement process is the basis of repertoire diversity, it is important to study its distribution quantitatively. With recent advances in high throughput sequencing, there is a growing body of data on repertoires for both T and B cells, in a variety of situations (Bolotin *et al.*, 2012; Georgiou *et al.*, 2014). Using large sequence datasets of rearranged, non-productive genes, the probability distribution of rearrangement events in human TCR beta chain and BCR heavy chains could be inferred using statistical methods, gaining important insights into the random processes underlying repertoire diversity (Elhanati *et al.*, 2015; Murugan *et al.*, 2012). However, these studies are based on a brute force approach, which enumerates every possible rearrangement scenario for each observed sequence. This is a very computationally costly procedure, which is unrealistic for very large datasets.

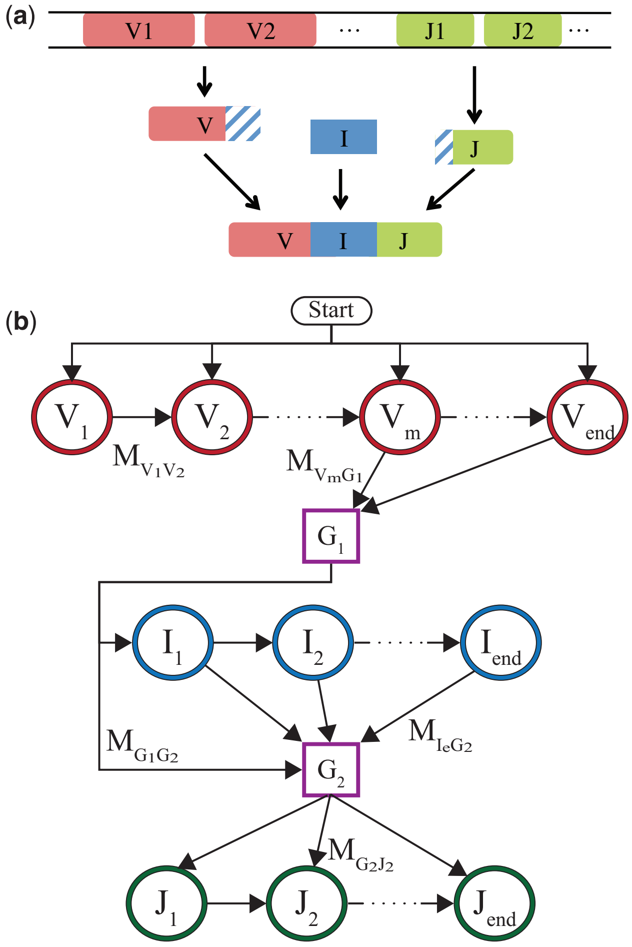

Fig. 1. (**a**) Schematic description of the rearrangement process for the alpha chains. Random V and J genes are chosen from the genome. A random number of nucleotides are trimmed from their facing ends. These ends are then joined with an insertion segment of variable length and composition. (**b**) Markov model for this rearrangement process, when the V and J gene choices are known. By progressing one path following the arrows, the model produces a rearranged receptor gene. Each state denoted by a circle emits a nucleotide. V and J states each emit one nucleotide from the chosen template, up to an error rate. Emissions from the I states are drawn from a specified distribution. The states represented by squares are nonemitting ghost states. The arrows represent the allowed transitions, some of them are marked on the diagram with $M_{SS'}$. The probabilities of the transitions and emissions are the parameters of the HMM, as described in the main text

In this report we present a dynamic programming approach to learn the distribution of rearrangement scenarios from large numbers of non-productive sequences in an efficient way. This approach is based on a Hidden Markov Models (HMM) formulation of the problem, and learns its parameters using a modified Baum–Welch (BW) algorithm to avoid the full enumerations of all scenarios. Many studies have described algorithms designed to process or post-process large numbers of rearranged TCR or BCR genes and extract the template V(D)J genes of the rearrangement (Bonissone and Pevzner, 2015; Brochet *et al.*, 2008; Frost *et al.*, 2015; Gadala-Maria *et al.*, 2015; Gaëta *et al.*, 2007; Munshaw and Kepler, 2010; Ohm-Laursen *et al.*, 2006; Paciello *et al.*, 2015; Ralph and Matsen, 2015; Russ *et al.*, 2015; Souto-Carneiro *et al.*, 2004; Thomas *et al.*, 2013; Volpe *et al.*, 2006; Wang *et al.*, 2008; Ye *et al.*, 2013). These tools process each sequence separately to obtain the best (but often incorrect) alignment to a V(D)J combination, sometimes using dynamic programming or HMM (Gaëta *et al.*, 2007; Munshaw and Kepler, 2010; Ralph and Matsen, 2015; Volpe *et al.*, 2006), and assume an implicit, *ad hoc* prior on rearrangements. By contrast, our algorithm explores all plausible alignments for each sequence from data to learn accurately the distribution of rearrangement events.

Once the model of rearrangement has been learned by our procedure, the entire distribution of possible sequences and their probabilities is accessible. Our algorithm can calculate the probability of any sampled sequence, even if it is not part of the data used to learn the model, and it can generate arbitrary numbers of synthetic sequences with the exact same statistics as the data. Such large samples can be used to investigate the origin of shared sequences, or public repertoires, between different individuals. The procedure can also calculate the entropy of the rearrangement process—a classical measure of sequence diversity. This enables us to further our understanding of the generation process, quantify the baseline state of the immune system and evaluate subsequent processes such as somatic selection. Finally, our work produces insights not just on the data sequences, but on the underlying biological processes.

2 Methods

2.1 Model

The algorithm assumes a general form for the probability distribution of possible rearrangements, and then finds the parameters of that distribution that best fit the sequence data (Elhanati *et al.*, 2015; Murugan *et al.*, 2012). For simplicity we first describe the model for the alpha chain of TCRs. The case of the beta chains will be described later.

The model specifies probability distributions for each of the rearrangement events: V and J gene choices $P(V, J)$, number of deletions conditioned on the gene being deleted $P(\text{del}V|V)$ and $P(\text{del}J|J)$, and insertion length and nucleotide identity $P(\text{ins})$. Together these distributions form the parameter set $\theta = \{P(V,J), P(\mathrm{del}V|V), \ldots\}$. The probability of a given rearrangement scenario $r = (V, J, \mathrm{del}V, \mathrm{del}J, \mathrm{ins})$ is then given by:

$$P_{\mathrm{rearr}}(r|\theta) = P(V,J)P(\mathrm{del}V|V)P(\mathrm{del}J|J)P(\mathrm{ins}). \qquad (1)$$

The specific form of the model assumes some dependencies between the events. In the above formula, for instance, the probabilities of each V and J choice can be dependent, but the insertion is independent from both, while the deletion probabilities are dependent only on the identity of the gene being deleted. The model choice is done based on biological knowledge, and has been validated by previous work in the case of the beta chain (Murugan *et al.*, 2012).

To avoid confounding factors related to thymic or peripheral selection, the inference of the parameters is performed on non-productive genes. During the maturation of cells, some rearrangement events produce non-productive genes that are either out of frame, having the wrong combination of insertions and deletions, or contain a stop codon. When this happens, the other chromosome in the same cell may undergo a second successful rearrangement event, ensuring the survival of the cell. Yet the non-productive rearrangements remain and are part of the sequence dataset. Since these sequences have no effect on the cell in which they reside, they have not been selected. Studying their statistics is thus equivalent to studying the generation process itself, with no selection. Nonproductive reads may also result from sequencing errors or problems in the assembly of sequences. However, contigs are typically designed to cover the entire CDR3, limiting such frameshifts to the genomic regions, typically to the longer V segment. To ensure that we analyze truly non-functional sequences, we exclude the very rare sequences with a frameshift in the genomic V segment, and restrict our analysis to sequences with a frameshift within the CDR3.

More generally, standard error correcting techniques applied to the raw sequencing data, such as correction using molecular barcodes and clustering methods, has been shown to limit the number of errors and misattributions of out-of-frame sequences (Bolotin *et al.*, 2012; Georgiou *et al.*, 2014; Robins, 2013; Shugay *et al.*, 2014).

Specifically, the data analyzed in this work were error-corrected by clustering raw reads as explained in Robins *et al.* (2009) and Zvyagin *et al.* (2014).

While the model specifies the distribution over rearrangement scenarios, the data consists of recombined sequences, denoted by $\mathbf{s}$, which can be the result of different scenarios r. The recombination events are hidden variables, and the likelihood of a sequence is the sum of the probabilities of all scenarios leading to it, $P(\mathbf{s}) = \sum_{r \to \mathbf{s}} P_{\mathrm{rearr}}(r)$. The likelihood of the sequence dataset cannot easily be maximized with respect to the model parameters. To overcome this problem, the Expectation–Maximization (EM) algorithm can be used to maximize the likelihood in an iterative manner. In each iteration, new model parameters θ' are chosen to increase the likelihood until the maximum is obtained. In the expectation step, the log-likelihood of hidden rearrangements is averaged over their posterior distribution under the current model θ, to form $Q(\theta'|\theta) = \sum_{a=1}^{n} \sum_{r} P(r|\mathbf{s}^{(a)}, \theta) \log P_{\mathrm{arran}}(r|\theta')$, where the sum on a runs over the n sequences $(\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(n)})$ in the dataset. The maximization step consists of maximizing $Q(\theta'|\theta)$ over θ' to obtain the new parameter set. Because of the simple factorized form of Eq. 1, this second step is equivalent to calculating the frequency of each rearrangement event under the posterior $P(r|\mathbf{s}^{(a)}, \theta) = P(r, \mathbf{s}^{(a)}|\theta)/P(\mathbf{s}^{(a)}|\theta)$. For example, $P(V, J)$ is updated according to:

$$P'(V,J) = \frac{1}{n} \sum_{a=1}^{n} \sum_{r:V,J} P(r|s^{(a)}, \theta), \qquad (2)$$

where the sum on $r: V, J$ runs over scenarios with gene choices V and J. Similar update rules are used for the other model parameters $P(\mathrm{del}V|V)$, $P(\mathrm{del}J|J)$, and $P(\mathrm{ins})$. The sums over possible scenarios, for each data sequence $s^{(a)}$, are computationally heavy. To make them easier, we now introduce an equivalent HMM formulation of the model.

2.2 HMM formulation

The almost linear structure of rearrangements allows for their description as a Markov chain. Hidden Markov models lend themselves to the much more efficient forward-backward algorithm for marginal estimations, and in combination with Expectation–Maximization, the Baum–Welch (BW) algorithm, for parameter inference. In general however, the V and J gene choices may be correlated in their joint distribution $P(V, J)$, breaking the Markovian nature of the rearrangement statistics. To preserve the Markovian structure, we built a separate HMM for each choice of the pair of germline genes (V, J), and use the forward-backward algorithm to calculate the marginals of the other rearrangement events conditioned on that choice. These conditional marginals will then be combined for all (V, J) to perform the maximization step of EM.

For a chosen (V, J) pair, a Hidden Markov Model (HMM) is constructed to yield the recombined sequences in accordance with Eq. 1. Figure 1b shows a diagram of the model. The model proceeds through a sequence of hidden states S, which emit nucleotides s_i, for $i = 1, \ldots, L$ where L is the sequence length, thus producing the entire sequence $\mathbf{s} = (s_1, \ldots, s_L)$.

We distinguish between two types of emitting states, represented by circles in Figure 1b. First, 'genomic states,' or V and J states, are defined for each position on the genomic templates V and J, and are denoted by $V_1, V_2, \ldots, V_{\mathrm{end}}$ for V states, and likewise for J states. These states emit the nucleotide encoded in the genomic template at the corresponding position, or, with a small error probability p_{err}, a different nucleotide. These non-templated emissions can be caused by sequencing errors, B cell hypermutations or uncharted alleles. Second, 'insertions states' emit random nontemplated nucleotides according to a distribution. I_1 corresponds to the first inserted nucleotide, I_2 to the second one and so on. In addition, we introduce two 'ghost states' G_1 and G_2, represented by rectangles in Figure 1b, between the V and I states, and between I and J states. These states do not emit nucleotides and their sole function is to reduce the number of possible transitions between states by isolating the state types, thus easing computations.

Each sequence is the result of a stochastic path through a series of states, defined in Figure 1b, and their random emissions. To illustrate how the HMM operates, we follow a possible path leading to the production of a light chain for a given choice of V and J. The chain starts from the V_1 state, going along the V gene and emitting nucleotides from the gene. Most of these nucleotides are those of the genomic template, up to the error rate. At some point (possibly before all V states are exhausted to account for potential V deletions) the process transitions to the first ghost state G_1. From G_1 the process goes to the first insertion state I_1, or directly to G_2 if there are no insertions. Each insertion state emits a nucleotide. After a certain number of insertions, the process moves to the second ghost state, G_2, and then on to a J state (but not necessarily J_1 to account for J deletions). Finally the process will continue along the J states until J_{end}, completing the sequence.

The HMM is defined by two sets of parameters which map directly onto $\theta\backslash P(V,J)$. The first set of parameters is the transition probabilities $M_{SS'}$ between any two states S and S' connected by an arrow in Figure 1. The transition rates between V states and G_1, and between G_2 and J states, can be mapped onto the deletion profiles of the V and J genes respectively, and the transition rates between the I states and G_2 can be mapped onto the distribution of the number of insertions. The transition matrix $M_{SS'}$ is very sparse, thanks in part to the ghost states, allowing for quick computations as we will see below. The second set of parameters are the emission probabilities $E_S(s)$ that nucleotide s is emitted by state S. If S is a genomic (V or J) state, then $E_S(s) = 1 - p_{\mathrm{err}}$ if s is the template nucleotide, which we denote by σ_S, and $p_{\mathrm{err}}/3$ otherwise. If S is an insertion state, it is given by a distribution $E_I(s)$, which we assume to be common to all insertion states, i.e. independent of the order of insertions.

2.3 A modified Baum–Welch algorithm

The Baum–Welch (BW) algorithm finds the parameters for a given HMM which maximize the likelihood of producing the observed sequences (Bishop, 2006; Durbin *et al.*, 1998). It is an instance of the EM algorithm, where the maximization step is performed using the forward-backward algorithm. Since our HMM is conditioned on the knowledge of the (V,J) pair, which is itself a hidden variable, BW cannot be applied without modification. However, we can still use the forward-backward algorithm to calculate the posterior probabilities of rearrangement events for a given sequence $\mathbf{s} = (s_1,\ldots,s_L)$ and a given putative (V,J) choice, and combine these probabilities at the end.

The forward pass of the forward-backward algorithm calculates $\alpha_i(S)$, the joint probability of the model being in a specific state S and emitting the sequence up to the ith nucleotide, $(s_1,\ldots,s_i)$. The backwards pass does the same for $\beta_i(S)$, the conditional probability of emitting the sequence upstream from position i, given that the state in this position is S:

$$\alpha_i(S) := P(s_1,\ldots,s_i,S|V,J), \tag{3}$$

$$\beta_i(S) := P(s_{i+1},\ldots,s_L|S,V,J). \tag{4}$$

These probabilities are calculated using the following recursion relations:

$$\alpha_i(S) = E_S(s_i)\sum_{S'} M_{SS'}\alpha_{i-1}(S'), \tag{5}$$

$$\beta_i(S) = \sum_{S'} E_{S'}(s_{i+1})M_{S'S}\beta_{i+1}(S'). \tag{6}$$

Since our transition matrix $M_{SS'}$ is very sparse, the sum over S' has few terms and can be calculated efficiently. Having obtained these forward and backward probabilities for a sequence given a choice of V and J genes, the posterior marginal probabilities for each transition $(S \to S')$, as well as the posterior emission probabilities are calculated as

$$P(S \to S'|V,J,\mathbf{s}) \propto \sum_i \alpha_i(S)M_{S'S}E_{S'}(s_{i+1})\beta_{i+1}(S'), \tag{7}$$

$$P(\mathrm{err}|V,J,\mathbf{s}) \propto \sum_i \sum_{S\in V,J} \alpha_i(S)\beta_n(S)(1-\delta_{\sigma_S,s_i}), \tag{8}$$

$$P_{\mathrm{ins}}(s|V,J,\mathbf{s}) \propto \sum_i \sum_{S\in I} \alpha_n(S)\beta_n(S)\delta_{s,s_i}, \tag{9}$$

up to a normalization constant, where δ denotes Kroeneker's delta.

The existence of ghost states requires making a small adjustment to this scheme. Each ghost state introduces an offset between the state index and the corresponding position on the sequence. Thus, V states in the n position correspond to position n in the sequence, I states to position $n-1$ in the sequence, and J states to position $n-2$.

Once the posterior marginals have been evaluated for each data sequence, and for each putative choice of V and J, we can combine them to obtain the update equations of our modified Baum–Welch algorithm:

$$M_{S'S} \leftarrow \frac{1}{n}\sum_{a=1}^{n}\sum_{V,J} P(V,J|\mathbf{s}^{(a)})P(S \to S'|V,J,\mathbf{s}^{(a)}), \tag{10}$$

$$p_{\mathrm{err}} \leftarrow \frac{1}{n}\sum_{a=1}^{n}\sum_{V,J} P(V,J|\mathbf{s}^{(a)})P(\mathrm{err}|V,J,\mathbf{s}^{(a)}), \tag{11}$$

$$E_S(s) \leftarrow \frac{1}{n}\sum_{a=1}^{n}\sum_{V,J} P(V,J|\mathbf{s}^{(a)})P_{\mathrm{ins}}(s|V,J,\mathbf{s}^{(a)}), \tag{12}$$

where the posterior probability $P(V,J|\mathbf{s}^{(a)})$ is calculated using Bayes' rule, $\propto P(\mathbf{s}^{(a)}|V,J)P(V,J)$, with $P(\mathbf{s}^{(a)}|V,J) = \sum_S \alpha_L(S)$.

Finally, the algorithm outputs the probability that any sequence $\mathbf{s}$ was generated as $P(\mathbf{s}) = P(\mathbf{s}|V,J)P(V,J)$. This probability can be used to calculate the log-likelihood of the model, $\sum_{a=1}^{n}\log P(\mathbf{s}^{(a)})$, and track the progress of the BW algorithm at each iteration.

2.4 Pre-alignments

For a given sequence, there may be many potential candidates for the segments (V,J), but not all are equally plausible, especially when sequence reads are long, and not all should be considered. Before starting the inference procedure, the sequences are locally aligned against all genomic templates using the Smith-Waterman algorithm. By creating a shortlist of genomic genes that had an alignment score larger than a tunable threshold, the inference procedure can exclude certain gene choices *a priori*. This saves considerable computation time by omitting rearrangement scenarios that would have a negligible effect due to high numbers of errors.

In addition, this pre-alignment procedure provides us with a mapping between the positions along the sequence read and each genomic gene. Thus, each of the V and J states of the HMM may only be present at a single position along the sequence, drastically limiting the number of states that we need to consider at each position and improving the speed of computations.

2.5 Beta chains

For the beta chain of the TCR, the model is similar to the one in Eq. 1, with the addition of a D gene choice, its deletions from both the left and right sides $(\mathrm{del}Dl, \mathrm{del}Dr)$ and two independent insertion events at the VD and DJ junctions $(\mathrm{ins}VD, \mathrm{ins}DJ)$:

$$\begin{aligned} P_{\mathrm{rearr}}(r|\theta) = {} & P(V,D,J)P(\mathrm{del}V|V)P(\mathrm{ins}VD) \\ & \times P(\mathrm{del}Dl,\mathrm{del}Dr|D)P(\mathrm{ins}DJ)P(\mathrm{del}J|J). \end{aligned} \tag{13}$$

A similar HMM as for the alpha chain can be built by adding genomic D states and having two types of insertion states, for the VD and DJ junctions, and extra ghost states to separate them. Then, the procedure described above can be applied *mutatis mutandis*.

In addition to the V and J gene, the D gene has to be chosen. An HMM is built for each triplet of genomic segments (V, D, J). D genes are short and deleted on both sides. For this reason, they are much harder to align. Even the position of a candidate genomic D

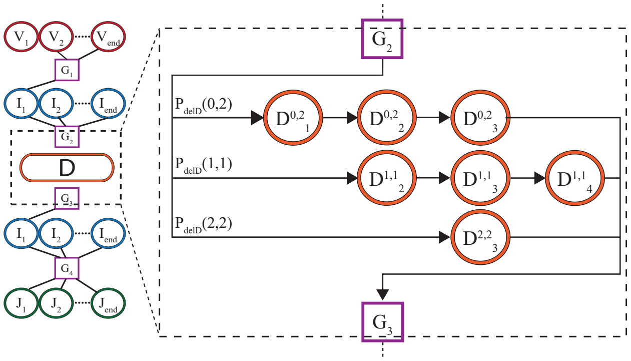

Fig. 2. Subdiagram of Markov model for beta chain, focusing on the D gene. Each row corresponds to a different pattern of deletions $(\mathrm{del}Dl, \mathrm{del}Dr)$ for the left and right ends of the D segments. State $D_d^{(\mathrm{del}Dl,\mathrm{del}Dr)}$ corresponds to the *d*th base in the D gene, when *l* bases are deleted from the left and *r* from the right. Each row is entered from the ghost state G_2 with probability $P_{\mathrm{del}D}(l,r) = P(\mathrm{del}Dl, \mathrm{del}Dr)$, and then proceeds deterministically until G_3.

segment along the sequence is no longer unambiguous as is the case for V and J segments. For this reason, we do not pre-align the D genes to the sequence. Instead, for each sequence all D genes with all possible locations are considered. Technically this means that the sequence positions at which D states may occur are not pre-specified.

During the rearrangement process D genes are deleted from both sides. The number of bases truncated from the left and right ends of the gene are correlated—in the extreme case, the sum of both deletions cannot exceed the length of the gene. Since the left and right D deletions correspond to transitions from non-adjacent states, this correlation cannot be described using a Markov model. We have addressed this issue by enumerating all the possible deletions from the left and the right as separate states. In practice, we define different types of D states for each choice of deletions, as depicted in Figure 2. Each D deletion profile $(\mathrm{del}Dl, \mathrm{del}Dr)$ defines a separate subchain of the Markov chain going along the D gene, which is entered from the previous ghost state with probability $P(\mathrm{del}Dl, \mathrm{del}Dr|D)$.

2.6 Entropy estimates

The inferred model can be used to characterize the diversity of the distribution of all possible receptors, and not just of the sampled sequences used when inferring that distribution. We quantify the diversity of a stochastic quantity X using the Shannon Entropy: $H(X) = -\sum_X P(X)\log_2 P(X)$ (measured in bits). For instance, $H(\mathbf{s})$ gives a measure of sequence diversity. Since a uniform distribution with 2^H outcomes has entropy H, the number 2^H can, even for non-uniform distributions, be interpreted as an effective diversity number, sometimes called true diversity. The entropy of rearrangements can be calculated explicitly thanks to the factorized form of the distribution. For example, for alpha chains:

$$H(r) = H(V,J) + \sum_V P(V)H(\mathrm{del}V|V) + \sum_J P(J)H(\mathrm{del}J|J) + H(\mathrm{ins}). \quad (14)$$

This expression clearly separates the contributions from (*V*, *J*) segment choice, deletions and insertions. The entropy of sequences $H(\mathbf{s})$ cannot be calculated in this way since receptor sequences can be produced by multiple rearrangements, but can easily be estimated by averaging $\log P(\mathbf{s})$, where $P(\mathbf{s})$ is calculated by the forward-backward algorithm as explained above.

3 Results

3.1 Implementation

The method was implemented in C++ (std11) as a command line tool. OpenMP was used for parallelization.

The main pipeline has two parts, the alignment module and the inference module. The input for the entire pipeline is a FASTA or plain text file with unique recombined and non-productive nucleotide sequences, and FASTA files with the genomic templates for the different V, J germline segments (as well as D in the case of heavy or beta chains). Genomic files can be obtained from genomic databases such as IMGT (Giudicelli *et al.*, 2005). The alignment code was written to read IMGT FASTA files and extract gene names.

Between the alignment and inference procedures, alignments below a certain threshold are discarded to improve performance. The threshold can be tuned for different datasets. Sequences that do not align well to at least one known gene of each type are completely eliminated as a quality control. Setting the thresholds should be done carefully, keeping a large majority of the sequences with at least one good alignment, but excluding ones which had only low score alignments. To this end, an auxiliary module is included that outputs statistics on the best alignment score for each sequence. Curated alignment files are saved at the end of the alignments stage, and used as input for the inference. Apart from the alignments, the only parameters needed for inference are the maximum numbers of insertions and deletions.

The output of the main pipeline is the value of the model parameters: the joint distribution of segment usage; the distributions of deletions at each of the deletion sites conditioned on the gene; the distribution of the number and composition of inserted nucleotides at the junctions; and a global error rate accounting for sequencing errors, hypermutations or genomic variants.

Two more modules are included. First, given a list of sequences and an inferred model, the software can compute generation probabilities for all sequences. Second, a synthetic sequence generation module that can produce sequences from a given model. This module can be used to study the properties of the distribution or to verify the inference algorithm using a known model, as we will see below.

3.2 Application to alpha chain data

The algorithm was applied to human TCR alpha chain sequences from Zvyagin *et al.* (2014). The data consist of around 60 000 unique non-productive sequences, each 151 bp long, covering both

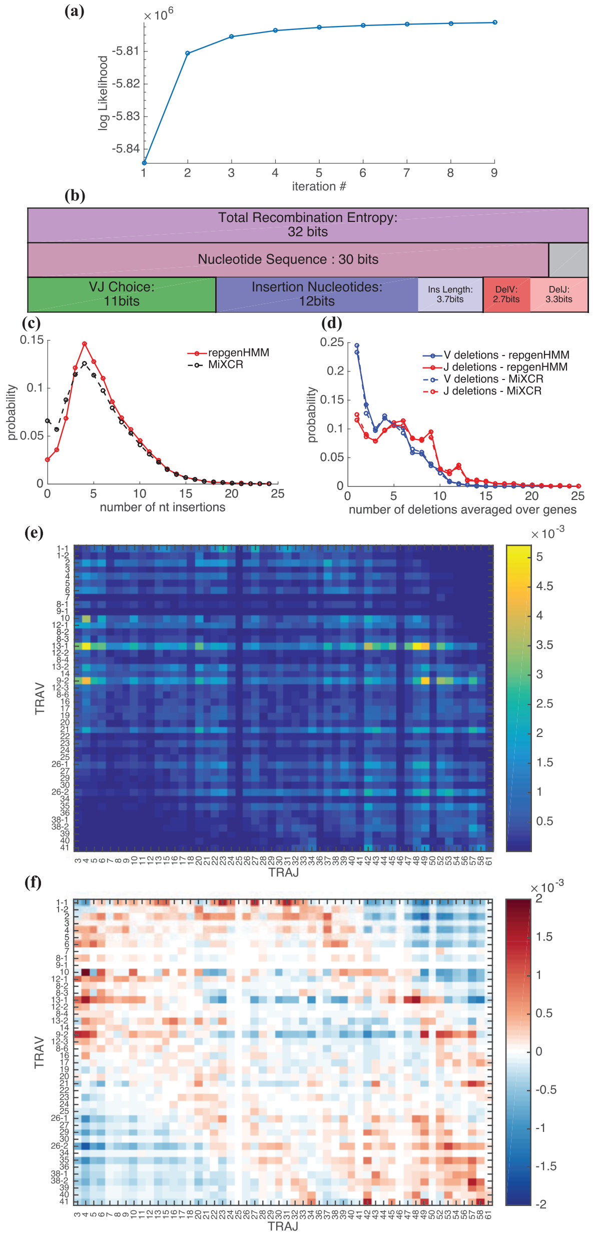

Fig. 3. TCR alpha chain rearrangement distribution inferred from sequence data taken from Zvyagin *et al.* (2014). (**a**) The log-likelihood of the data given the model saturates as a function of the number of iterations of the Expectation–Maximization algorithm. (**b**) Shannon entropy of rearrangements (top row) and sequences (middle row). The sequence entropy is lower than the total recombination entropy because of convergent rearrangements. The rearrangement entropy is the sum of entropies of its elementary events (bottom row). (**c**) Distribution of the number of inserted nucleotides (solid curve). For comparison, the same distribution obtained by the MiXCR software is represented by a dashed line. (**d**) Distributions of the number of deletions for both V and J genes, averaged over genes. (**e**) Joint distribution for V and J usage, $P(V, J)$. Genes are ordered by position along the genome. (**f**) The covariance $P(V, J) - P(V)P(J)$ clearly shows strong correlations for genes that are either close to the separation between the V and J segments, or far from it

the V and J segments. Sequences were aligned to lists of genomic sequences from the IMGT online database (Giudicelli *et al.*, 2005), and then given as input to the inference algorithm. The model converged rapidly as can be seen by the quick saturation of the likelihood (Fig. 3a).

The entropy of the rearrangement process quantifies the diversity of possible scenarios. It was calculated using Eq. 14 and found to be 32 bits (Fig. 3b, top line). This entropy can be partitioned into contributions from each of the rearrangement events—segment choice, insertions and deletions (bottom line). The largest contribution to the entropy comes from the insertions, followed closely by gene choice. We also estimated the entropy of the sequence distribution (middle line), which is smaller than the rearrangement entropy because of convergent rearrangements—multiple rearrangements leading to the same sequence (grey box). This estimate was based on samples of simulated sequences. Undersampling bias and error were corrected for by using samples of different sizes and validating the convergence of the entropy. The entropy of the alpha chain sequences is 30 bits, which corresponds to a diversity number of about 10^9. Inferred insertion and deletion profiles for the alpha rearrangements are shown in Figure 3c and d, with the deletion profile averaged over genes. The peak of the insertion distribution is at 5 bp, similar to previous results for the beta chain.

The joint distribution of gene usage for V and J, represented in Figure 3e, shows a wide variety of gene usage probabilities, with clear dependencies on the ordering of genes according to their location on the chromosome (Lefranc and Lefranc, 2001). To better see these dependencies, in Figure 3f we plotted $P(V, J) - P(V)P(J)$ as a measure of the correlation between V and J choices. In Warmflash and Dinner (2006) and Hawwari and Krangel (2007), it was proposed that rearrangements can occur in several steps, following earlier accounts in mice (Huang and Kanagawa, 2001). When a V and a J segment are joined, the genomic segments that were between them are excised, but the segments located on the outer flanks remain. Then, successive rearrangements joining these outer segments might occur. It was hypothesized that early joining events involve V and J genes that are close to each other, hence proximal to the boundary between the V and J cassettes. Later joining events, on the other hand, should involve more distal genes as the proximal genes are likely to have been excised. This phenomenon is expected to lead to correlations between pairs of genes which are either both distal or both proximal, which is consistent with the results of Figure 3f. Notice also that in the intermediate range our model predicts low correlation within a certain window of distances between the V and J genes.

To control for the finite size of the datasets, we ran our model on subsamples of the data. The distributions inferred using half of the dataset (30 000 sequences) differed by only 1% from the whole dataset, as measured by the Kullback–Leibler divergence normalized by the entropy.

3.3 Test on synthetic data

In order to check the validity of the algorithm, we ran it on sequences that were produced according to a known model. We generated 100 000 synthetic sequences according to the model learned in the previous section, and relearned a model from these sequences using our algorithm. In Figure 4 we compare the parameters of the model used for generation to those of the inferred model. Sampling was repeated 5 times to estimate sample noise, which was found to be very small for all parameters, except for gene usage (error bars in Fig. 4b).

The insertion bias, i.e. the usage of the different nucleotides in inserted segments, is inferred almost perfectly, as seen in the inset of Fig. 4a. The probabilities for each V,J choice also show excellent agreement, within sampling errors (Fig. 4b). The distribution of the number of insertions also agrees very well (Fig. 4a). However, when

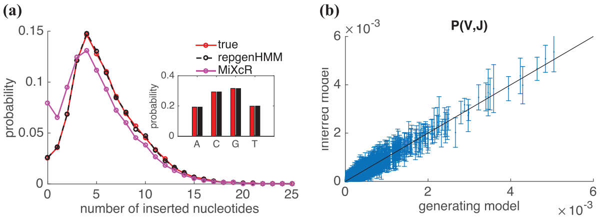

Fig. 4. Performance of the algorithm on synthetic data. Sequences generated using a known model were given as an input to the inference algorithm. The results of the inference are compared to the true model used for generation, for (**a**) the distribution of the number of insertions (inset: usage of inserted nucleotides) and (**b**) V, J gene usage. The error bars, which correspond to sample noise, are smaller than symbol size for (a). In (a) we also report the distribution of insertions obtained using MiXCR

inferring the same model, but with a higher error rate (1%), we found that the inferred insertion distribution noticeably overestimated the probability of zero insertions.

3.4 Comparison to existing tools

Results and performance can be compared with existing tools based on deterministic methods, which rely on the best alignment to the V and J genomic segments to call the number of deletions, insertions, and calculate their distributions. Here we compare our results to MiXCR (Bolotin *et al.*, 2015), a recent, representative and fast implementation of this method. We expect other existing software tools (Brochet *et al.*, 2008; Gaëta *et al.*, 2007; Thomas *et al.*, 2013; Ye *et al.*, 2013) to give equivalent results.

The distributions obtained by summarizing the MiXCR alignments are shown in Figure 3c and d, and do not agree with our inference. MiXCR assignments overestimate zero insertions by a twofold factor, and underestimate the central peak. This distortion is also apparent in synthetic data, for which we know the true distribution (Fig. 4a). In this case, our method infers the correct distribution of insertions with high accuracy, while MiXCR does not, and even identifies a spurious second peak in the distribution at zero insertions.

Because it treats each sequence in a fully probabilistic manner, our method is slower than MiXCR. On a multicore PC with 46 AMD Opteron processors, one iteration over a 50 000 database of alpha chains sequences takes about 200 s. For comparison, MiXCR could run over the same sequences in about 10 s. Note that the memory requirements of repgenHMM do not limit its performance. This slower time must be balanced by the obvious advantage conferred by the inference of the correct distribution. An accurate estimate of the distribution of insertions can be crucial for understanding the mechanistic details of the insertion process by TdT (Gouge *et al.*, 2015) or for estimating the number of cells with invariant receptors (NK or MAIT, for instance) (Gapin, 2014; Greenaway *et al.*, 2013) or zero-insertion clonotypes in cells that do not express TdT (Komori *et al.*, 1996).

3.5 Sharing of alpha chain sequences

The generation module of our software, which has no equivalent in existing tools, can be used to investigate important biological questions, such as the origin of receptor sharing between unrelated individuals. Sharing of receptor sequences is sometimes interpreted as resulting from convergent selection in response to common pathogens. However, it has recently been proposed that sharing could simply arise by chance, through the independent recombination of the same receptors in different individuals (Shugay *et al.*, 2013; Venturi *et al.*, 2011). In fact, we expect this to be the main source of sharing in non-productive sequences, because they are not subject to selection (Murugan *et al.*, 2012).

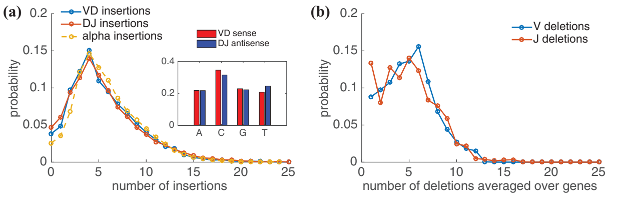

Fig. 5. TCR beta chain rearrangement distribution inferred from sequence data previously analyzed in (Murugan *et al.*, 2012). (**a**) Distribution of the number of insertions at both VD and DJ junctions, and comparison with the distribution of insertion in the alpha chain from Figure 3c. Inset: The nucleotide usage is identical for VD and DJ insertions when considered on opposite strands. (**b**) Distribution of the number of deletions on both the V and J genes, averaged over different genes

We investigated the sharing of alpha out-of-frame sequences in two unrelated individuals from Zvyagin *et al.* (2014). The data contained 1.7×10^{-7} shared sequences for every pair of sequences in the two individuals. We learned a distinct model for each of the two individuals, then generated a large number of sequences from each model, and counted the number of shared sequences between the two sets. The two generated datasets shared 1.4×10^{-7} sequences per pair, in good agreement with the data, confirming that independent convergent recombination was the main source of shared sequences between these two individuals.

3.6 Application to beta chain data

The beta and heavy chain algorithm was applied to the human TCR beta data that was already analyzed in (Murugan *et al.*, 2012) using brute-force methods. The inferred model parameters were all very similar to those reported in Murugan *et al.* (2012), confirming the validity of our algorithm. The distribution of the number of insertions and deletions are displayed in Figure 5. Remarkably, insertion profiles at the two insertion sites are very close to each other, as previously reported, but they also closely match the insertion profile of the alpha chain (Figure 5a). Nucleotide usage in each of the two inserted regions (between V and D, and between D and J) is shown in the inset. The VD insertion base usage is similar to the usage of the complementary bases (antisense) in the DJ region, suggesting that the biological mechanism is operating on the opposite strands for both insertions types, as previously noted (Murugan *et al.*, 2012). From the computational point of view, because the algorithm enumerates both the D gene choice and its deletions, its benefit is smaller for beta chains than for alpha chains.

4 Discussion

The strength of the adaptive immune system lies in its inherent diversity. This dynamic diversity is the result of several stochastic biological mechanisms, including complex enzymatic reactions that alter the DNA structure and composition. The action of some of the enzymes such as RAG and TdT have been studied extensively (see Schatz and Swanson, 2011; Jung and Alt, 2004; Schatz and Ji, 2011 for reviews). In our work we have studied the way in which the diversity is generated in a top down approach, focusing on statistical properties as inferred from sequenced receptor genes. In principle, this is a computationally difficult problem, due to the very large number of possible rearrangement scenarios.

The algorithm described here can be used to study the properties of generation of receptor chains of T cells in any species, from large sequence datasets. Our dynamic programming approach, which is a variant of the Baum–Welch algorithm, takes advantage of the linear structure of rearrangements to avoid a full enumeration of scenarios. In a brute-force approach such as the one presented in Murugan *et al.* (2012) and Elhanati *et al.* (2015), the algorithmic complexity scales as the product of the numbers of each independent rearrangement event. By contrast, the complexity of the method we presented here is additive with the number of insertions and deletions.

Despite technological advances, sequencing techniques still introduce errors. In addition, allelic variants and hypermutations in B-cells introduce additional discrepancies between known template genes and sequence reads. Our method can be used with models that account for such events. In the presented version of the algorithm, substitution errors are already fully accounted for, and can be partially interpreted as hypermutations, as has already been demonstrated for B cell heavy chains in Elhanati *et al.* (2015), albeit with a different implementation. Hypermutations are known to be context dependent (Cowell and Kepler, 2000; Dunn-Walters *et al.*, 1998; Oprea *et al.*, 2001; Spencer and Dunn-Walters, 2005). Our method could be modified to take this into account, by having the substitution rate depend on the local sequence motif around the location of the substitution. Insertion and deletion errors could likewise be handled by adding transitions that skip or repeat bases. Other potential future software developments include a more general model of nucleotide insertions, where each insertion depends on the previous one, or the addition of palindromic insertions. These modifications can be implemented readily, as they do not affect the Markovian structure of the process. In addition, a module calculating the probability of generation of amino-acid sequences (and not just nucleotide sequences) could be implemented using a similar approach with Markov transitions between codons.

We find many common features of generation for the alpha and beta chains of the TCR. There is a difference of diversity, due to the greater length and complexity of the beta compared to the alpha chain. The diversity number of the beta chain repertoire, estimated to be approximately 10^{14} in Murugan *et al.* (2012), is therefore much larger than that of the alpha chain reported here, 10^9. Assuming that the rearrangements of the alpha and beta chains are independent, the total TCR diversity is about 10^{23}.

Inferring statistical properties of the underlying biological processes can be thought of as a diagnostic tool for the properties of the immune system, and could be used in a variety of clinical settings. The generation process and subsequent selection shape the initial diversity of the immune system, and we have found this process to be remarkably universal across normal, healthy humans, expect for slight variations in gene usage (Murugan *et al.*, 2012; Elhanati *et al.*, 2015). However, infections or irregularities in the immune system can be seen as perturbations that will change these distributions. By comparing the normal distributions to different pathological situations, information on the reaction of the immune system can be extracted. This information could in turn be used for diagnosis, using fast computational tools such as the one presented here.

With more and more immune repertoire sequence data being collected, efficient algorithms are needed. The ability to quickly infer and analyze large datasets is essential both for our basic understanding of the adaptive immune system and also for specific clinical applications.

Acknowledgements

We thank members of the Chudakov and Lebedev groups at the Institute of Bioorganic Chemistry of the Russian Academy of Sciences, and in particular M. Pogorelyy, for sharing their TCR alpha data with us.

Funding

This work was supported by European Research Council Starting Grant n. 306312.

Conflict of Interest: none declared.

References

Bishop,C.M. (2006). *Pattern Recognition and Machine Learning*. New York, USA: Springer.

Bolotin,D.A. *et al.* (2012) Next generation sequencing for TCR repertoire profiling: platform-specific features and correction algorithms. *Eur. J. Immunol.*, **42**, 3073–3083.

Bolotin,D.A. *et al.* (2015) MiXCR: software for comprehensive adaptive immunity profiling. *Nat. Methods*, **12**, 380–381.

Bonissone,S. and Pevzner,P. (2015). Immunoglobulin classification using the colored antibody graph. In: Przytycka,T.M. (ed.) *Research in Computational Molecular Biology SE - 7, volume 9029 of Lecture Notes in Computer Science*. Switzerland: Springer International Publishing, pp. 44–59.

Brochet,X. *et al.* (2008) IMGT/V-QUEST: the highly customized and integrated system for IG and TR standardized V-J and V-D-J sequence analysis. *Nucleic Acids Res.*, **36**, 503–508.

Cowell,L.G. and Kepler,T.B. (2000) The nucleotide-replacement spectrum under somatic hypermutation exhibits microsequence dependence that is strand-symmetric and distinct from that under germline mutation. *J. Immunol.*, **164**, 1971–1976.

Dunn-Walters,D.K. *et al.* (1998) Base-specific sequences that bias somatic hypermutation deduced by analysis of out-of-frame human IgVH genes. *J. Immunol. (Baltimore, MD.: 1950)*, **160**, 2360–2364.

Durbin,R. *et al.* (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, UK: Cambridge University Press.

Elhanati,Y. *et al.* (2015) Inferring processes underlying B-cell repertoire diversity. *Philos. Trans. R. Soc. Lond. Ser. B: Biol. Sci.*, **370**, 20140243.

Frost,S.D.W. *et al.* (2015) Assigning and visualizing germline genes in antibody repertoires. *Philos. Trans. R. Soc. Lond. Ser. B: Biol. Sci.*, **370**, 20140240.

Gadala-Maria,D. *et al.* (2015) Automated analysis of high-throughput B-cell sequencing data reveals a high frequency of novel immunoglobulin V gene segment alleles. *Proc. Natl. Acad. Sci. U. S. A.*, **112**, E862–E870.

Gaëta,B.A. *et al.* (2007) iHMMune-align: hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences. *Bioinformatics*, **23**, 1580–1587.

Gapin,L. (2014) Check MAIT. *J. Immunol. (Baltimore, MD.: 1950)*, **192**, 4475–4480.

Georgiou,G. *et al.* (2014) The promise and challenge of high-throughput sequencing of the antibody repertoire. *Nat. Biotechnol.*, **32**, 158–168.

Giudicelli,V. *et al.* (2005) IMGT/GENE-DB: a comprehensive database for human and mouse immunoglobulin and T cell receptor genes. *Nucleic Acids Res.*, **33**, D256–D261.

Gouge,J. *et al.* (2015) Structural basis for a novel mechanism of DNA bridging and alignment in eukaryotic DSB DNA repair. *EMBO J.*, **34**, 1126–1142.

Greenaway,H.Y. *et al.* (2013) NKT and MAIT invariant TCRα sequences can be produced efficiently by VJ gene recombination. *Immunobiology*, **218**, 213–224.

Hawwari,A. and Krangel,M.S. (2007) Role for rearranged variable gene segments in directing secondary T cell receptor alpha recombination. *Proc. Natl. Acad. Sci. U. S. A.*, **104**, 903–907.

Huang,C.Y. and Kanagawa,O. (2001) Ordered and coordinated rearrangement of the TCR locus: role of secondary rearrangement in thymic selection. *J. Immunol.*, **166**, 2597–2601.

Jung,D. and Alt,F.W. (2004) Unraveling V(D)J recombination; insights into gene regulation. *Cell*, **116**, 299–311.

Komori,T. *et al*. (1996) Repertoires of antigen receptors in Tdt congenitally deficient mice. *Int. Rev. Immunol*., **13**, 317–325.

Lefranc,M.P. and Lefranc,G. (2001). *The T Cell Receptor FactsBook*. Factsbook. London: Academic Press.

Munshaw,S. and Kepler,T.B. (2010) SoDA2: a Hidden Markov Model approach for identification of immunoglobulin rearrangements. *Bioinformatics*, **26**, 867–872.

Murugan,A. *et al*. (2012) Statistical inference of the generation probability of T-cell receptors from sequence repertoires. *Proc. Natl. Acad. Sci. U. S. A.*, **109**, 16161–16166.

Ohm-Laursen,L. *et al*. (2006) No evidence for the use of DIR, D-D fusions, chromosome 15 open reading frames or VHreplacement in the peripheral repertoire was found on application of an improved algorithm, JointML, to 6329 human immunoglobulin H rearrangements. *Immunology*, **119**, 265–277.

Oprea,M. *et al*. (2001) The targeting of somatic hypermutation closely resembles that of meiotic mutation. *J. Immunol*., **166**, 892–899.

Paciello,G. *et al*. (2015) VDJSeq-Solver: in silico V(D)J recombination detection tool. *Plos One*, **10**, e0118192.

Ralph,D.K. and Matsen,F. (2015) Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation. *PLoS computational biology*, **12**, e1004409.

Robins,H. (2013) Immunosequencing: applications of immune repertoire deep sequencing. *Curr. Opin. Immunol*., **25**, 646–652. (Special section: Systems biology and bioinformatics/Immunogenetics and transplantation).

Robins,H.S. *et al*. (2009) Comprehensive assessment of T-cell receptor beta-chain diversity in alphabeta T cells. *Blood*, **114**, 4099–4107.

Russ,D.E. *et al*. (2015) HTJoinSolver: Human immunoglobulin VDJ partitioning using approximate dynamic programming constrained by conserved motifs. *BMC Bioinf*., **16**, 1–11.

Schatz,D.G. and Ji,Y. (2011) Recombination centres and the orchestration of V(D)J recombination. *Nat. Rev. Immunol*., **11**, 251–263.

Schatz,D.G. and Swanson,P.C. (2011) V(D)J recombination: mechanisms of initiation. *Annu. Rev. Genet*., **45**, 167–202.

Shugay,M. *et al*. (2013) Huge overlap of individual TCR beta repertoires. *Front. Immunol*., **4**, 466.

Shugay,M. *et al*. (2014) Towards error-free profiling of immune repertoires. *Nat. Methods*, **11**, 653–655.

Souto-Carneiro,M.M. *et al*. (2004) Characterization of the human Ig heavy chain antigen binding complementarity determining region 3 using a newly developed software algorithm, JOINSOLVER. *J. Immunol. (Baltimore, MD.: 1950)*, **172**, 6790–6802.

Spencer,J. and Dunn-Walters,D.K. (2005) Hypermutation at A-T base pairs: the a nucleotide replacement spectrum is affected by adjacent nucleotides and there is no reverse complementarity of sequences flanking mutated A and T nucleotides. *J. Immunol*., **175**, 5170–5177.

Thomas,N. *et al*. (2013) Decombinator: a tool for fast, efficient gene assignment in T-cell receptor sequences using a finite state machine. *Bioinformatics (Oxford, England)*, **29**, 542–550.

Venturi,V. *et al*. (2011) A mechanism for TCR sharing between T cell subsets and individuals revealed by pyrosequencing. *J. Immunol. (Baltimore, MD.: 1950)*, **186**, 4285–4294.

Volpe,J.M. *et al*. (2006) SoDA: implementation of a 3D alignment algorithm for inference of antigen receptor recombinations. *Bioinformatics (Oxford, England)*, **22**, 438–444.

Wang,X. *et al*. (2008) Ab-origin: an enhanced tool to identify the sourcing gene segments in germline for rearranged antibodies. *BMC Bioinf*., **9**, S20.

Warmflash,A. and Dinner,A.R. (2006) A model for TCR gene segment use. *J. Immunol*., **177**, 3857–3864.

Ye,J. *et al*. (2013) IgBLAST: an immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res*., **41**, 1–7.

Zvyagin,I.V. *et al*. (2014) Distinctive properties of identical twins' TCR repertoires revealed by high-throughput sequencing. *Proc. Natl. Acad. Sci. U. S. A.*, **111**, 5980–5985.