Simulation of adaptive immune receptors and repertoires with complex immune information to guide the development and benchmarking of AIRR machine learning

Authors

Maria Chernigovskaya[1,☯](0000-0002-1507-4171), Milena Pavlović[2,3,☯](0000-0002-2484-3868), Chakravarthi Kanduri[2,3](0000-0002-4783-9060), Sofie Gielis[4](0000-0001-7678-281X), Philippe A. Robert[1,5](0000-0003-1345-5015), Lonneke Scheffer[2](0000-0001-8900-075X), Andrei Slabodkin[1](0000-0002-9320-1666), Ingrid Hobæk Haff[6], Pieter Meysman[4](0000-0001-5903-633X), Gur Yaari[7](0000-0001-9311-9884), Geir Kjetil Sandve[2,3,❇](0000-0002-4959-1409), Victor Greiff[1,❇,#](0000-0003-2622-5032)

[1] Department of Immunology, University of Oslo and Oslo University Hospital, Oslo, Norway
[2] Department of Informatics, University of Oslo, Oslo, Norway
[3] UiO:RealArt Convergence Environment, University of Oslo, Norway
[4] Department of Computer Science, University of Antwerp, Antwerp, Belgium
[5] University Hospital Basel, Basel, Switzerland
[6] Department of Mathematics, University of Oslo, Oslo, Norway
[7] Faculty of Engineering, Bar-Ilan University, Tel Aviv, Israel

[☯]Equal contribution, [❇]Joint supervision, [#]Correspondence: victor.greiff@medisin.uio.no

Abstract

Machine-learning methods (ML) have shown great potential in the adaptive immune receptor repertoire (AIRR) field. However, there is a lack of large-scale ground-truth experimental AIRR data suitable for AIRR-ML-based disease diagnostics and therapeutics discovery. Simulated ground-truth AIRR data are required to complement the development and benchmarking of robust and interpretable AIRR-ML approaches where experimental data is inaccessible or insufficient as of yet. The challenge for simulated data to be useful is the ability to incorporate key features observed in experimental repertoires. These features, such as complex antigen or disease-associated immune information, cause AIRR-ML problems to be challenging. Here, we introduce LIgO, a modular software suite, which simulates AIRR data for the development and benchmarking of AIRR-based machine learning. LIgO incorporates different types of immune information both on the receptor and the repertoire level and preserves native-like generation probability distribution. Additionally, LIgO assists users in determining the computational feasibility of their simulations. We show two examples where LIgO simulation supports the development and validation of AIRR-ML methods: (1) how individuals carrying out-of-distribution immune information impacts receptor-level prediction performance and (2) how immune information co-occurring in the same AIRs have an impact on the performance of conventional receptor-level encoding and repertoire-level classification approaches. The LIgO software guides the advancement and assessment of interpretable AIRR-ML methods.

Introduction

B- and T-cell receptors (BCRs and TCRs, or AIRs — adaptive immune receptors) are the agents of the adaptive immune system that recognize and neutralize antigens, such as bacteria, viruses, cancer cells, or autoantigens. Immune memory of past and ongoing immune states, i.e. different disease or infection states, is stored in AIR repertoires (AIRRs). In the manuscript text, we will interchangeably use the terms AIRR and repertoire, as well as receptor and AIR. The immune memory is stored as immune information or "immune signal", which may be encoded in AIRR frequency, e.g., clonal expansion (*1*), receptor sequence, e.g., antigen-specific binding motifs or disease-associated motifs where motifs may be entire sequences (*2*) or subsequences (*3–5*) or as a combination thereof. The complexity of AIRR biology contributes to the long-standing problem of how to extract and use these immune signals to understand the mechanisms of adaptive immunity and utilize them for immunodiagnostics (*2*, *6*, *7*) and immunotherapeutic design (*8*, *9*).

The recognition of antigens can be defined at the receptor level and at the repertoire level. On the receptor level, each immune receptor is specific to one or more antigens, and is potentially polyreactive to several antigens or pathogens using several possible paratopes (*10–12*). This makes antigen recognition difficult to quantitatively describe because each antigen can be bound by many receptors with very little sequence overlap (*11*, *13–18*). On the repertoire level, immune recognition is determined by a set of antigen-specific AIRs and usually, it remains unknown which AIRs from the whole repertoire are specific to the target and how the distribution (e.g., frequency, positional weight matrices) of antigen-specific AIRs changes across immune states. Recognition on the repertoire level is also complex to describe and quantify because each individual possesses a different set of AIRs with a genetic sequence diversity of about 10^8–10^9 different AIRs per individual and low sequence overlap (<1%) across individuals (*8*, *19*, *20*). Altogether, repertoire-level research problems range from the identification of repertoires of a given immune state containing immune-state-associated AIR biomarkers to the identification of the specific receptors within a repertoire that confers an immune recognition property to this repertoire. Furthermore, as few as one AIR per million to several thousands may be associated with a disease in naive and non-naive repertoires (*21*, *22*), indicating that the immune signal may be very sparingly represented (*23*, *24*).

Machine learning is a powerful tool for detecting complex signals in various types of biological data (*25–32*). Previous studies have shown that even relatively simple ML approaches can be successfully applied to classify immune repertoires based on the immune status (repertoire-based classification), with their performance improving with dataset size (*2*, *4*, *33*). These few promising successes showcase the potential of ML for detecting complex signals in AIRR data but many challenges remain for repertoire-based diagnostics to reach a level of maturity sufficient for clinical application (*6*). For example, although the accuracy achieved for predicting CMV status from TCR repertoires was found to be relatively high (*2*), CMV is known to leave a particularly strong mark on the adaptive system with a high percentage of CD8+ T cells specific to CMV increasing with age (*34*). To achieve clinically acceptable accuracy levels for predicting a variety of disease states, more sophisticated and sensitive ML approaches will be needed (*8*, *24*, *35–39*). Similarly to repertoire classification, it remains challenging to predict antigen binding for individual AIRs (receptor-based classification) given, for instance, the current scarcity of the antigen-labeled data, as well as challenges in quantification of generalization and negative data definition (*3*, *9*, *15*, *16*, *40–44*).

Progress on both repertoire and receptor-based ML classification problems is hindered by a lack of large-scale data with known ground-truth information, such as corresponding immune status and immune signal (*45*, *46*). Such large-scale ground truth data is required for the development and benchmarking of novel AIRR-adapted ML approaches as well as the benchmarking of established approaches to test their predictive performance and capability to recover immune signals (*45*, *47–50*). Specifically, the lack of ground truth data hinders the benchmarking under various conditions such as sample size and AIRR architecture (*45*). Therefore, it is challenging to assess using only experimental data whether an AIRR-ML method successfully learned disease-associated features or spurious factors that are associated with high predictive performance (*46*). In contrast, simulation enables the generation of AIRR datasets of virtually unconstrained size, with fine-grained control over introduced signals (ground truth) (*45*). While simulations can bridge the gap between the lack of suitable experimental data to guide the development and benchmarking of novel AIRR-ML methods relevant for real-world applications, not all simulation approaches are created equal. A key challenge for simulations is to faithfully represent the characteristics and statistical properties of experimental AIR(R)s with high fidelity.

There exist various tools for simulating a set of individual AIRs with or without set parameters (*39*, *51–60*). These tools accurately mimic the V(D)J-recombination process to varying degrees, but most do not simulate the selection or implantation of putative immune signals (*52*). When introducing such immune signals, it is crucial not to perturb existing and underlying statistical properties of AIRR data, such as germline gene recombination statistics (*51*, *61*) and the degree of sharing of public AIR sequences between individuals (*62*), since perturbated properties could lead to ML methods achieving high accuracy based on artificial features rather than meaningful signals.

Here, we implemented the simulation suite LIgO, which enables the generation of immune-signal-labeled synthetic AIR(R) datasets for the development and benchmarking of AIRR-based machine learning methods. Importantly, the LIgO suite allows simulating AIR(R) datasets at scales way beyond what is currently experimentally feasible. Immune signals in AIRs can be simulated based on either rejection sampling or signal implantation that preserves AIR generation probability distribution through importance sampling. LIgO can be used with immuneML, an open-source ecosystem for AIRR-ML analysis (*33*). We show the applicability of LIgO on two distinct machine learning use cases, investigating (1) how individuals carrying out-of-distribution immune signals impact receptor-based prediction performance and exploring (2) the limitations of conventional encoding schemes for AIRR binary classification when immune signals co-occur within the same AIR.

Results

Immune signal and immune event formalization in AIR(R) data

Adaptive immune receptors (AIRs) can be labeled with their antigen recognition capacity (one or multiple antigens) and adaptive immune receptor repertoires (AIRRs) with the individual's disease state(s). These labels can be used to formulate a machine learning (ML) classification problem on AIRR data (Fig. 1B). Both receptor-based (AIR-based) and repertoire-based (AIRR-based) classification may be considered as a binary classification problem (the AIR binds or does not bind to the antigen), a multi-label classification problem (the AIRR is associated with multiple disease states), or a multi-class classification problem (the AIRR is associated with a specific stage of a disease, e.g., stages of cancer). As of yet, there is limited understanding of how immune signals are encoded inside AIRRs (*8*). To simulate AIRR data for benchmarking receptor and repertoire-based AIRR-ML methods, there is a need to account for different types of putative immune signals that ML methods are expected to retrieve and the statistical properties of synthetic immune signals vis-à-vis AIRR-ML tasks (Fig. 1A).

To support a broad variety of potentially interesting AIRR-ML problem formulations, we opted for a very flexible, yet structured, specification of immune signals. We decided to separate between biological immune events — e.g., disease, allergy, vaccination, or other situations that elicit an adaptive immune response — and associated encoded signals in the receptors. The AIRs produced during the immune response and associated with a (biological) immune event are considered immune-event specific. Each immune-event specific receptor contains one or more immune signals, which, for instance, reflects its binding rules to the immune event antigen(s) (Fig. 1C). To prevent excessive complexity, we limit the number of immune signals per receptor, ensuring that each receptor contains no more than two immune signals.

We define that on the receptor level, an *immune signal* consists of a set of *motifs*, where each motif is a distribution over amino acids or nucleotides of any length up to an entire AIR sequence (*2*). Furthermore, additional AIR information is defined, such as V and J genes, and the distribution of the motif set locations within the CDR3 (Fig. 1C), in line with previous studies (*2–4*, *63*). In addition, AIR information may include derived features such as CDR3 length (*64*), physicochemical properties (*65*), or binding energy (*66*). In the case of paired chain data, the immune signal would include this information from both chains (*3*, *15*, *16*, *67*).

On the receptor level, we define an *immune event* as a receptor label associated with a set of immune signals (Fig. 1C). For example, an HIV immune event may contain several immune signals corresponding to different HIV epitopes. If an AIR is polyreactive, it is linked to multiple immune events. If an AIR is not specific to any annotated antigen, then it has no label.

On the repertoire level, we define an immune event as a distribution of immune signals in conjunction with a distribution of clone counts (Fig. 1C), which can represent, for example, clonal expansion (*1*). Our definition of immune signals and events on the receptor and repertoire levels is designed to encompass a wide range of data formats and biological signals in a comprehensive manner that have been previously reported on experimental data (*8*). From here on, we will not separate the terms immune event and immune signal on the repertoire and receptor level.

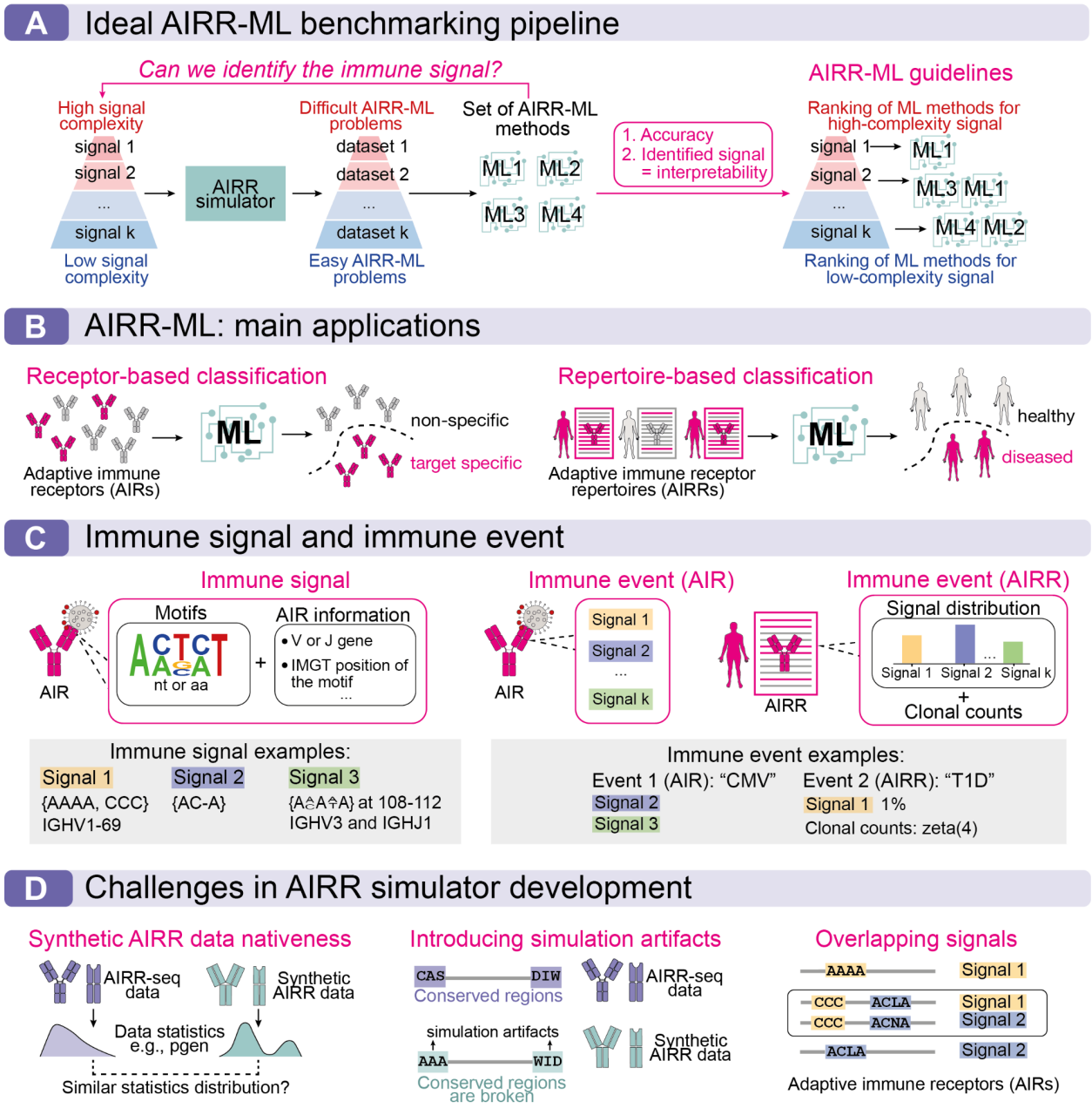


Figure 1: Formalization of an AIRR simulator suitable for AIRR-ML benchmarking both on the receptor and the repertoire level (A) An ultimate goal of AIRR-ML benchmarking is to establish guidelines for achieving effective repertoire and receptor classification by taking into account the various complexities of immune signals. To achieve this goal, there is a need for ranking AIRR-ML approaches based on their prediction accuracy and ability to recover interpretable immune signals. (B) The two main classes of AIRR-ML classification problems are receptor-based and repertoire-based classification. The aim of the receptor-based classification is to predict antigen binding for individual AIRs, and the aim of the repertoire-based classification is to classify immune repertoires based on their immune status. (C) We define an immune signal as the union of a set of motifs and AIR-specific information (the latter may be absent). A motif is a distribution over nucleotides or amino acids at multiple positions. AIR-specific information can be, e.g., V or J gene, position of motif in a CDR3, signal of the other chain for the paired data. We illustrate signal definition with three different signals. Signal 1 consists of the two k-mers AAAA and CCC, and the V gene IGHV1-69, which means that only receptors with IGHV1-69 and containing either AAAA or CCC are considered signal 1 specific. Similarly to signal 1, signal 2 consists of a gapped 4-mer and signal 3 consists of a motif represented by a PWM on the 108-112 IMGT positions and restricted by IGHV3 and IGHJ1 genes. Formally, we define an immune event on the receptor and the repertoire levels. On the receptor level, an immune event is a set of immune signals. On the repertoire level, an immune event is defined as a distribution of immune signals along with clonal count distribution. Concrete examples of immune events on the receptor level may include, for instance, "cytomegalovirus infection (CMV)" which consists of signal 2 and signal 3 described

above. And an AIR is considered as CMV-specific if it contains signal 2 or signal 3. We may also define an AIRR carrying a hypothetical T1D (type 1 diabetes) immune event if 1% of AIRs contain signal 1 and the clone counts of the signal-specific receptors follow a zeta distribution. (D) Developing an AIRR simulator that implements immune signal and event concepts of varying levels of complexity and is also suited for AIRR-ML benchmarking both on the repertoire and receptor level presents several challenges. (i) For real-world relevance, simulated AIRR data with implanted signals should be as similar as possible to experimental AIRR-seq data. Specifically, such similarity may be defined in terms of generation probability distribution (pgen), where pgen is the probability of a given V(D)J recombination process to generate a given AIR. (ii) A simulation process might introduce simulation artifacts into data, for example, break the conserved receptor regions. In this case, even if an AIRR-ML approach perfectly detects the implanted signal, it will differ from biologically relevant signals. (iii) One receptor might contain multiple signals corresponding to different immune events and thus immune signals and immune events can be overlapping both on the receptor and repertoire levels which complicates their simulation.

Simulation approaches for generating biologically relevant AIR(R) data

To fit the AIRR-ML problem formulation, one needs to simulate signal-specific (positive) AIRs and non-specific (negative) AIRs and control their frequencies in an AIRR dataset to encode an immune event. In this study, we consider two approaches to simulate biologically relevant AIRs carrying immune signals — rejection sampling and signal implantation (Fig. 2C). With rejection sampling, a pool of random background AIRs is generated and only the AIRs containing a predefined signal(s) are considered signal-specific. The main advantage of the rejection sampling approach is that it does not modify the original AIRs, meaning that receptor-based statistics like generation probability for each individual AIR are not artificially perturbed (Fig. 1D). However, rejection sampling is not computationally feasible for immune signals with low population probability, since estimated time of finding an immune-signal-specific receptor in background receptors is inversely proportional to its population probability and large number of AIRs needs to be generated for each receptor that is kept after signal filtering. An alternative way to generate signal-specific receptors is signal implantation, which replaces a substring of a CDR3 region of a given background AIR with one of the immune signal (gapped) k-mer, or motif in general. Unlike rejection sampling, computational efficiency of signal implantation does not depend on the population probability of the immune signal. However, signal implantation may introduce artifacts to simulated AIRs because it modifies the original sequence and may create unrealistic sequences (Fig. 1D). Also, signal implantation does not work for every immune signal, since some signals cannot be easily implanted when an AIR is already constructed - e.g., if the signal to be implanted is a specific V gene, then substituting the original V gene with the new one might potentially conflict with the CDR3 sequence of the original AIR. Both rejection sampling and signal implantation may be coupled with an optional importance sampling step. Importance sampling minimizes potential perturbation of simulated AIR pgen distribution, which, if left unchecked, may cause artificial increase of ML performance. Depending on the defined signals, LIgO will allow testing whether rejection sampling or implantation methods can be performed. Altogether, both the rejection sampling and the implantation approach allow generation of datasets with a controlled number of signal-specific receptors in the synthetic data.

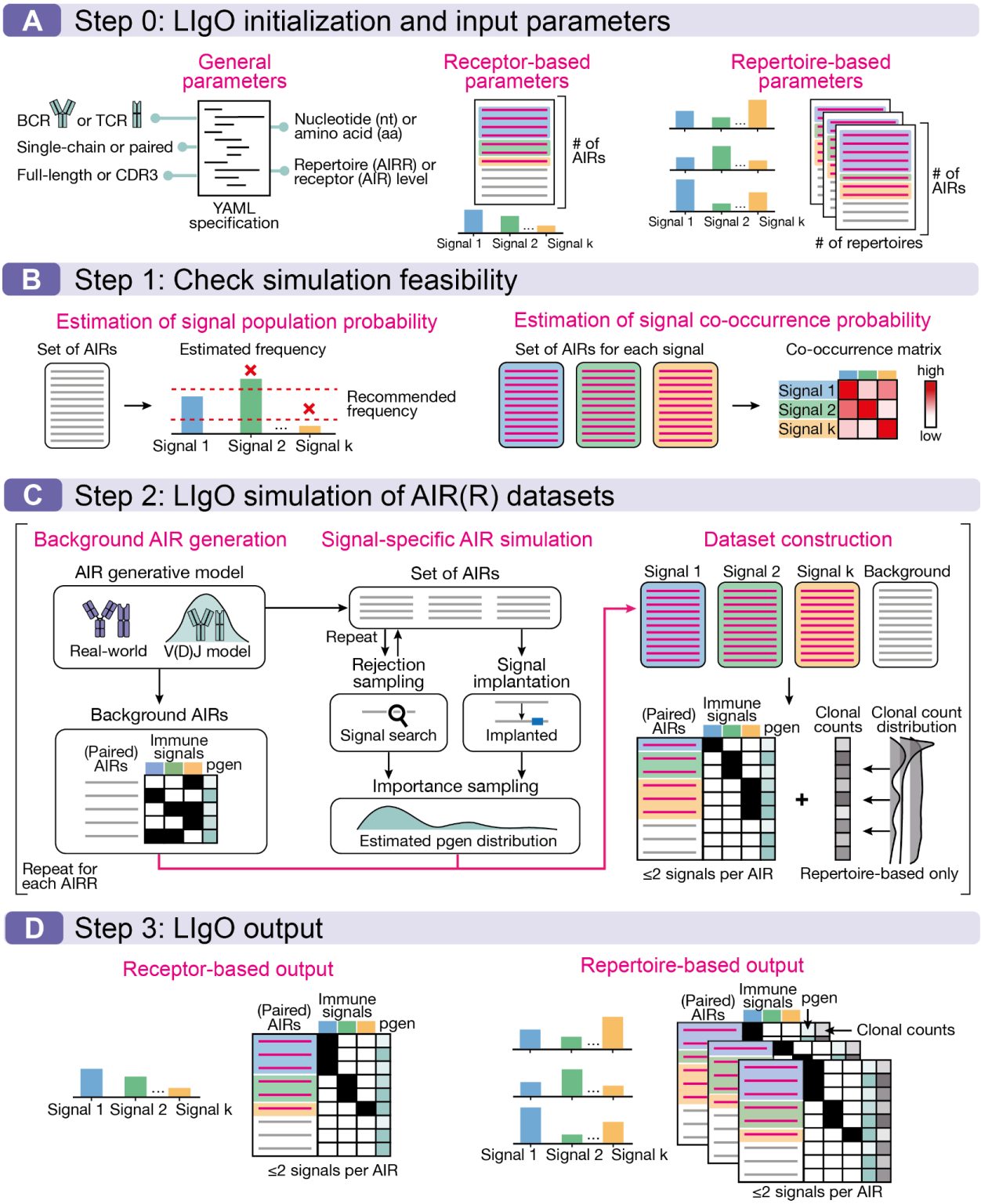


Figure 2: LIgO workflow for configuring AIRR-ML benchmarking datasets

(A) For initialization, a user may define general simulation parameters, such as the type of AIR (BCR or TCR), receptor- or repertoire-based simulation, paired- or single-chain receptors, type of sequence alphabet — nucleotide (nt) or amino acid (aa), output format — full-length or CDR3 only. For the receptor-based simulation, a number of generated AIRs and a set of immune signals with their frequencies in the final dataset are defined. For the repertoire-based simulation, a number of repertoires, number of receptors in a repertoire, and a mixture of immune signals corresponding to each repertoire with their frequencies are defined. In total, the sum of all immune signal frequencies including co-occurring signal frequencies for each AIRR should be less or equal to 1. (B) The optional second step helps users to check simulation feasibility based on the parameters defined during initialization. The first helper function estimates the frequency of each immune signal in a set of AIRs. It is not recommended to

use rejection sampling for immune events whose estimated frequency falls below the recommended lower bound, by default 0.001%. In addition, the user can redefine the immune signal with low frequency in order to increase its frequency, e.g., simplify the immune signal definition by removing certain V gene(s) or by simplifying the motif in the definition. It is also not recommended to use immune signals with estimated frequency above the upper bound of 10% because they may impact the biological nativeness of immune signals given the rarity of such high-frequency signals. The second helper function estimates the co-occurrence of defined immune signals. In order to limit computational complexity, LIgO restricts the number of immune signals per one AIR to at most two and all AIRs containing three or more immune signals will be removed during the simulation process. If the user wants to limit the number of immune signals per AIR by one, then immune signals frequently co-occurring in the background may decrease rejection sampling efficiency. It is recommended for the user to redefine immune signals in this case. (C) In the main simulation step, LIgO requires a generative model to obtain background AIRs, which can either be simulated from a V(D)J-recombination process (e.g., IGoR (*51*, *68*) or immuneSIM (*52*), where a custom V(D)J model can be inferred from real-world data) or taken from experimental data (e.g., iReceptor (*69*) or OAS (*70*)). Subsets of immune-signal specific receptors can be simulated using rejection sampling or implantation from a set of AIRs which may or may not have the same V(D)J generation parameters as the background AIRs. Additionally, if the simulation is defined by a V(D)J model, the immune-signal specific AIRs can be subsampled according to their generation probabilities during the importance sampling step. Then, background and immune-signal-specific AIRs are combined in one dataset with respect to all immune signals, and AIRs containing more than two immune signals are removed. For repertoire-based simulation, clonal counts are sampled from a zeta distribution (*1*) with user-defined parameters for each immune event or constructed based on the normalized receptor generation probabilities. If the simulation is performed on the repertoire level, then the main simulation step should be repeated *n* times, where *n* is the number of desired AIRRs to be simulated. (D) LIgO reports all parameters used during the simulation steps. For the receptor level simulation, a set of (paired) AIRs (V and J genes, CDR3 region and optionally the full-length receptor) is reported together with their specificity to each immune signal and immune event, generation probability (optional), and immune signal position in each AIR. For the repertoire level simulation, receptors of each repertoire are annotated with the immune signal specificity matrix, immune signal position in each AIR, generation probability of each AIR (optional), and clonal counts. Repertoire-level metadata with immune signal and immune event annotations is also available in a metadata file accompanying the dataset. LIgO outputs datasets in AIRR-compliant format both for receptor and repertoire-level simulations.

LIgO simulation workflow

Here we present the LIgO workflow that enables simulation of immune receptor (AIR) and repertoire (AIRR) data for the development and benchmarking of AIRR-based machine learning. As the first step (Fig. 2B), LIgO estimates the feasibility of a simulation described with a given set of user-defined simulation parameters (Fig. 2A) by assessing two potential bottlenecks (Fig. 2B). First, LIgO determines if the user-defined signals can be efficiently generated using rejection sampling. If this is not the case, LIgO suggests using signal implantation optionally coupled with importance sampling or modifying the signal(s) definition, because attempting to simulate rare immune events using rejection sampling can lead to the first computational bottleneck. The second bottleneck may arise if many AIRs violate our complexity restriction, which specifies that each AIR must contain no more than two immune signals. In such cases, these AIRs will be eliminated from the simulation process which may also increase simulation time. LIgO assesses if any two immune signals frequently co-occur in the same AIR and suggests the user to redefine co-occurring immune signals. Both bottlenecks can notably increase the simulation time and may render it quasi-infinite. Although the feasibility assessment step is optional, we recommend users to check the simulation feasibility before running the main simulation step.

LIgO initiates the main simulation step by obtaining background receptors to be used as a basis for simulation (Fig. 2C). It does so by either utilizing a generative model, which offers the flexibility of, for instance, using a custom IGoR V(D)J model (*51*, *61*, *68*) or an experimental AIRR-seq dataset. While the V(D)J model can simulate an unlimited number of AIRs, it may contain simulation artifacts. Experimental AIRR-seq data might be more realistic, but they are of a limited size and may contain unknown artifacts (*46*), such as batch effects and AIRR imprinting of previous diseases, (*45*), thus potentially rendering them insufficient for complex LIgO simulations.

Once the background receptor dataset has been obtained, LIgO achieves the desired amount of immune-signal-specific AIRs using rejection sampling and signal implantation (Fig. 2C). To enhance the

rejection sampling efficiency for immune signals involving specific V or J genes when using a generative model to obtain the background receptors, LIgO constructs a skewed V(D)J model. This specialized model generates AIRs containing only the required V or J genes, automatically meeting the gene criteria for the given immune signal. If the simulation is defined by a V(D)J model, the user can evaluate the generation probability (pgen) of each AIR during the simulation process. Although pgen evaluation is time-consuming, the generation probability distribution can be used to further filter signal-specific AIRs during the importance sampling step (Fig. 3E). Importance sampling controls the overall pgen distribution and provides higher chances for signal-specific AIRs with more likely pgens of being included in the final dataset. If the user wishes to add background receptors in the resulting AIR(R) dataset(s), LIgO provides background receptors generated by the V(D)J model or experimental data. Additionally, they can be further filtered so they are devoid of immune signals. If the filtration is not performed, immune signals in the background receptors may impact the overall immune signal frequencies in the final simulated dataset(s).

Upon completion of the simulation, LIgO constructs AIR(R) dataset(s) with the desired frequencies of immune-signal-specific receptors for each immune event. In the scenario of repertoire-based simulation, vectors of simulated clonal frequencies can be optionally added to each AIRR. By adjusting the parameters of the clonal frequency distribution for each immune signal, users have the capability to simulate processes resembling affinity maturation and the expansion of signal-specific clones. LIgO outputs AIRR-compliant data, ensuring that each receptor is productive and contains no more than two immune signals (Fig. 2E). Additionally, to facilitate further ML processing, the data is annotated with a binary matrix showing for every immune signal its presence in every receptor and, if so, the positions in the receptor sequence.

Comparison of simulation strategies implemented in the LIgO simulation workflow

As discussed earlier, LIgO simulation approaches have their advantages and disadvantages. Rejection sampling maintains the AIR sequence unchanged but lacks computational efficiency compared to signal implantation. We investigated how one can use signal implantation to attain results similar to those obtained through rejection sampling (Fig. 3A). First, if an immune signal is defined as a k-mer or more generally as a motif, then the user should take into account the natural signal occurrence distribution across CDR3 positions and implant the motif according to the natural distribution. For a 3-mer amino acid signal, only approximately half of the motifs were evenly distributed across the CDR3, while the other half tended to occur on one CDR3 position (Fig. 3B, Supplementary Fig. 1). Second, if an immune signal is implanted into any CDR3 position, then around 20% of all signal-specific AIRs will have zero pgen after implantation, which means that these zero-pgen AIRs cannot be generated through the VDJ recombination model. If we restrict the implantation position and do not implant in the first and the last two CDR3 positions corresponding to conserved CDR3 regions, then the percentage of zero-pgen AIRs drops to around 2% (Fig. 3C). In general, signal implantation decreases pgen of the original AIR by two to four orders of magnitude on average (Fig. 3D). However, when signal implantation is coupled with importance sampling, LIgO will no longer produce signal-specific AIRs with zero pgen values, and the overall pgen difference after implantation is less pronounced than with pure signal implantation (Fig. 3C, Fig. 3D). Finally, we demonstrated that by employing signal implantation coupled with importance sampling to simulate signal-specific AIRs, we can achieve an overall similar pgen distribution as those generated through rejection sampling, while not increasing the computational time (Fig. 3E).

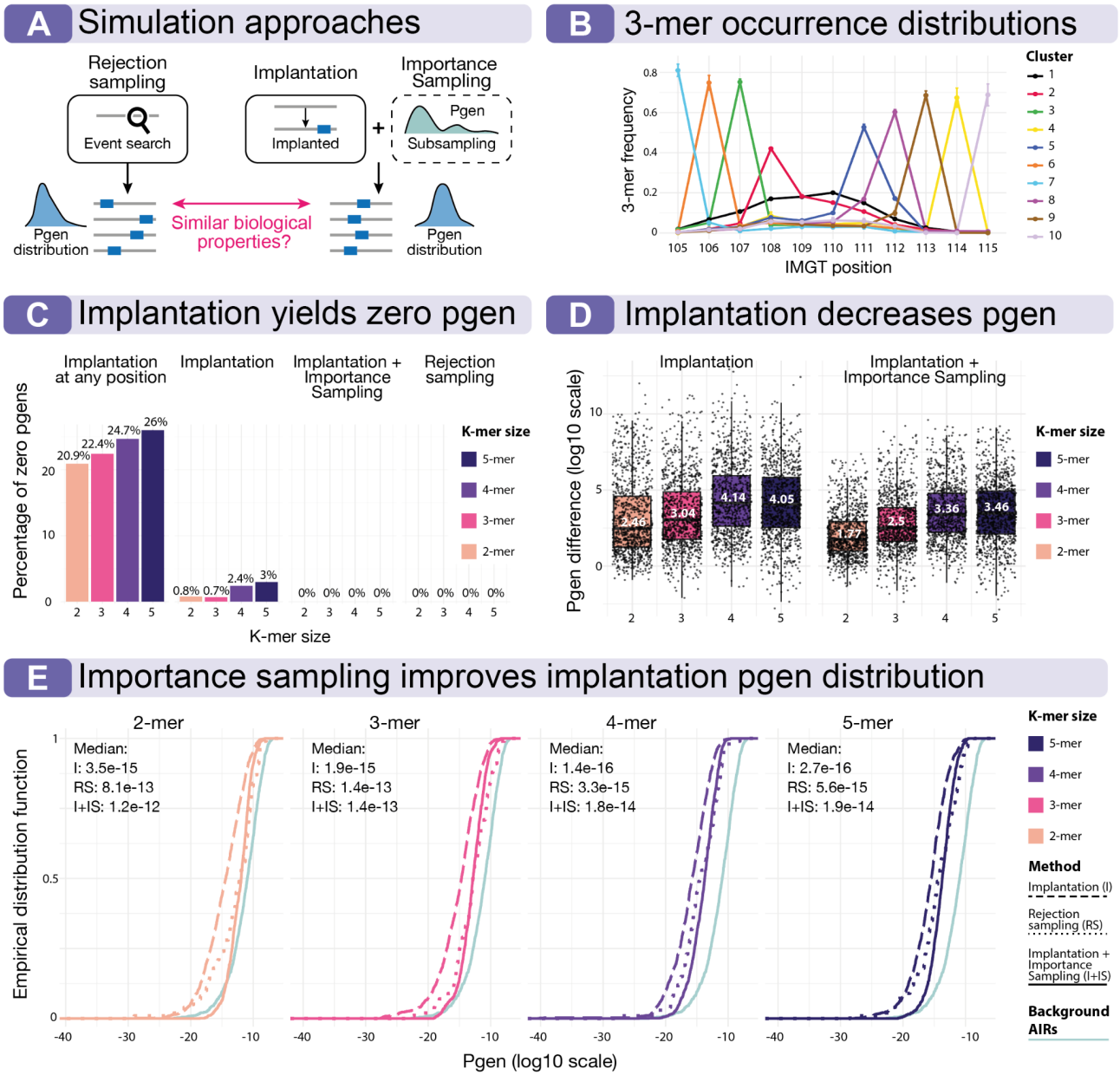


Figure 3: Comparison of biological properties of different simulation strategies

(A) We compared three simulation strategies — (i) rejection sampling, which keeps an AIR sequence but may be computationally inefficient, (ii) signal implantation, which changes an AIR sequence and thus may change its pgen, but is computationally efficient, and (iii) signal implantation combined with importance sampling which preserves the pgen distribution of simulated AIRs and is computationally efficient. (B) We illustrate that different immune signals may have distinct occurrence distribution across CDR3 positions, where positions are IMGT-standardized. Here we show position distribution patterns of 3-mers occurring in amino acid TRB sequences generated using the default IGoR model (*51*). Position frequencies were clustered into 10 clusters using hierarchical clustering, and one line on the plot represents the mean frequency for each cluster. Some 3-mers have one dominating position with occurrence probability above 0.4 while others 3-mers occurrence distribution is more uniform (cluster 1 shown in black). (C) Implantation at any CDR3 position and implantation in non-conserved CDR3 regions (shown as implantation) yields AIRs with pgens=0, which therefore can not be generated by the VDJ recombination model. However, when implantation is coupled with importance sampling, it prevents the generation of impossible AIRs, aligning with rejection sampling results. (D) Both implantation and implantation coupled with importance sampling decrease on average the pgen of the simulated AIR compared to the pgen of the original AIR before implanting. Each boxplot value corresponds to one AIR and represents log10 difference of its pgen before and after signal implantation. Implantation followed by importance sampling has lower median

log10 pgen difference values compared to implantation alone, the median value for each k-mer size is shown in white. (E) The pgen distribution of signal-specific AIRs generated using implantation (shown with dashed line) is shifted compared to the pgen distribution of signal-specific AIRs generated through rejection sampling (shown in dotted line). However, this shift may be corrected using the importance sampling strategy (shown in solid line). The turquoise solid line illustrates the pgen distribution of the background AIRs, which were used as the source of AIRs for all simulation strategies.

Use case 1: Out-of-distribution receptor-level simulation using LIgO

Train and test data composition may impact AIRR-ML models' quality and accuracy (*71*). For example, (i) if the train and test set overlap or contain highly similar sequences, then the accuracy of the trained ML model may be overly optimistic (a case of data leakage). (ii) If the training data is not comprehensive and representative, it may lead to generalization problems and underperformance on unseen data (*41*, *42*, *44*, *66*, *72–76*). Since different individuals have largely non-overlapping AIRRs (*77*, *78*) and, thus, potentially different immune signals to the same targets, we investigated how immune signal variability may impact ML accuracy and model generalization to other unseen (potentially out-of-distribution) individuals.

For this use case, we considered a scenario where an immune signal is defined as one motif, yet each individual carries a slightly different modification of this motif. Specifically, we defined the immune signal as a motif AA-A with four variations of the immune signal — AAAA, AANA, AACA, AAGA, reflecting AIRs from four different individuals (Fig. 4B). We simulated $4 \cdot 10^4$ AIRs and replicated the experiment ten times, which resulted in $4 \cdot 10^5$ of simulated AIRs in total. Simulation details can be found in the Methods section. We illustrated this use case with a binary receptor-based classification task (Fig. 4A), categorizing AIRs into signal-specific (i.e., an AIR contains one out of four signal 4-mers) or non-specific (i.e., an AIR does not contain any signal 4-mers). We trained and evaluated a logistic regression (LR) model using two train-test strategies — (i) random, which randomly splits all receptors from four individuals into train and test set, and (ii) leave-one-individual out, which places all AIRs from one individual into a test set and AIRs from all the other individuals into a training test (Fig. 4C).

The LR trained on the random train-test strategy achieved higher balanced accuracy than the LR trained on the leave-one-individual-out train-test approach (Fig. 4D). Based on interpretability analysis, it can be seen that the random train-test model obtained and learned information about all 4-mers while the leave-one-individual out model did not learn the signal that was not present in the training set. This use case illustrates the impact of train and test data on AIRR-ML quality. It demonstrates that while perfect prediction accuracy may be observed with a random split approach, prediction accuracy may come across as more limited based on a leave-one-individual-out split strategy. Leave-one-individual-out split strategy is considered more real-world relevant since it is likely that a limited number of individuals does not suffice to cover the entire antigen-specific AIR sequence diversity.

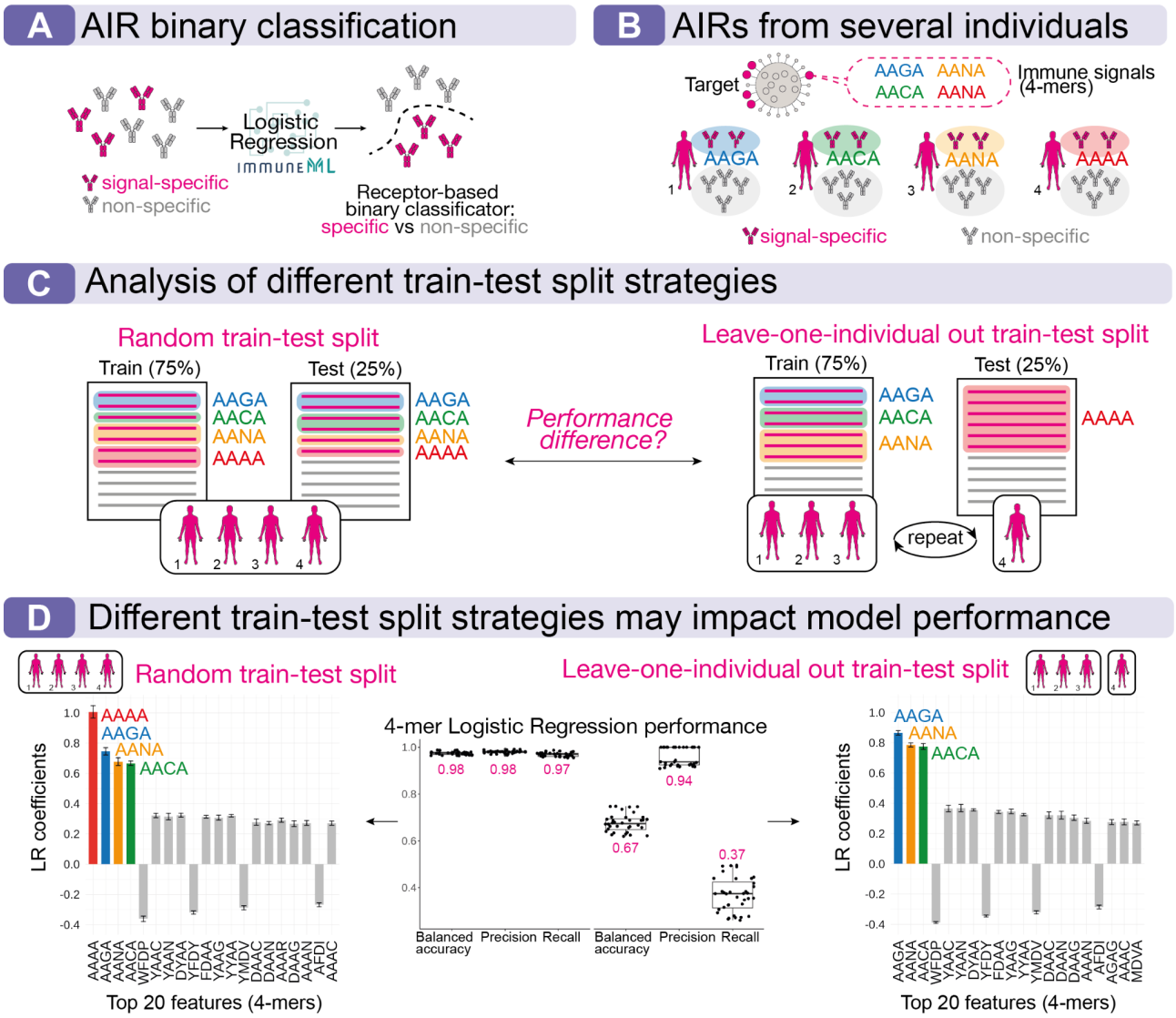


Figure 4: Use case 1: Out-of-distribution receptor-level simulation using LIgO

(A) We illustrate an example of an out-of-distribution problem using logistic regression as a ML method for receptor-based binary classification, i.e., classification of signal-specific vs non-specific AIRs. See Methods section for more details. (B) We defined the immune signal by four 4-mers differing in the third position — AAGA, AANA, AACA, AANA. We simulated four sets of 10000 AIRs using signal implantation, where 5000 AIRs were signal-specific, and 5000 AIRs were non-specific, i.e., not containing any signal 4-mers. We assumed that one set of AIRs corresponds to one individual, and each of the four sets of AIRs contained only one out of four signal 4-mers. We replicated the simulation ten times. (C) We applied two train-test split strategies to train the logistic regression model for receptor-level classification — (i) random split, where the signal-specific AIRs containing the four signal 4-mers are present both in the train and in the test set, and (ii) leave-one-individual-out split for every individual, where AIRs with only three 4-mers out of four were present in the train data and the AIRs with the fourth signal 4-mer was present in the test data only. (D) Performance comparison and interpretability analysis. We trained a 4-mer logistic regression with two train-test split strategies on all ten replicates resulting in ten ML models. The 4-mer logistic regressions trained on the random train-test split achieved higher accuracy and its largest coefficients corresponded to the four signal 4-mers (AAAA, AAGA, AANA, AACA). The 4-mer logistic regression trained on the leave-one individual out train-test split achieved lower performance with a median balanced accuracy of 0.57. The LR achieving the best performance was trained on the split, where all AIRs containing the 4-mer AAAA were placed in the test set. The largest coefficients of this LR contain three out of four signal-specific 4-mers (AAGA, AANA, AACA). The boxplots display model performances for all four splits within 10 replicates and the median performance value is shown in red. The bar plots display the top twenty logistic regression coefficients from the best model with the highest absolute value, and the error bars indicate the standard deviation for ten replicates.

Use case 2: Limitations of conventional encoding schemes for repertoire-level binary classification when immune signals co-occur within the same AIR

Several previous studies have suggested that the information needed for repertoire-based classification is sufficiently represented through conventional k-mer encoding of AIRs (*3*, *4*, *65*, *79–81*). However, immune signals may comprise multiple co-occurring motifs within the same receptor (*3*, *15*, *16*). In this case, conventional k-mer encoding that treats each k-mer in isolation may not be sufficient to represent the distinction between immune states. LIgO can simulate such complex immune signals that may pose challenges to ML methods that do not consider the co-occurrence of motifs within the same receptor. To demonstrate this, we simulated AIRR datasets for a repertoire classification problem, where the distinction between contrasting immune states cannot be observed using conventional k-mer encoding (Fig. 5A).

Specifically, using LIgO, we simulated many TRB repertoire datasets, where each dataset contained 200 repertoires (100 each of positive and negative-labelled) with 10^5 productive CDR3 amino acid sequences per repertoire. Simulation details are described in the Methods section. We assumed that the immune signal that differentiates positive and negative-labelled repertoires is defined as the joint co-occurrence of two amino acid 3-mers: GDT and SGL (Fig. 5B). The experiments were performed in two different settings with varying frequencies at which the immune signal occurs: 0.01% and 0.1%. The simulation with LIgO was carried out in such a way that the relative abundances of the chosen 3-mers, when considered in isolation, is similar for both positive and negative-labelled repertoires (Fig. 5D). The distinction between the labels will only be apparent when the receptor-level co-occurrence of GDT and SGL is considered.

We assessed the performance of the logistic regression (LR) with the conventional k-mer encoding on the LIgO-simulated data with co-occurring immune signals. In our previous study (Figure 3.a in (*63*)), we showed that the LR approach exhibited close to perfect performance even at signal frequency of 0.01% if a signal is defined as a set of motifs and each AIR contains no more than one signal motif. However, the same method exhibited poor performance even at a very high signal frequency of 0.1% in the current simulations where the signal is defined as two co-occurring 3-mers (Fig. 5E). As a proof of concept, we implemented a custom multiple instance learning approach (MIL) which is based on the underlying assumption that the immune signal is encoded in 3-mer pairs. The detailed description of the custom MIL approach can be found in the Methods section (Fig. 5C). In contrast to the LR with the conventional k-mer encoding, the custom MIL approach was able to perfectly classify the repertoires even at a signal frequency of 0.01% (Fig. 5E) with median balanced accuracy of 1. Given that the simulation design did not include any label or attribute noise, and that the immune signal frequency is quite distinct between positive and negative labelled repertoires (~ 10 AIRs vs ~ 1 AIR harbouring signal), it was expected that the ML model exhibited near-perfect performance. To summarize, benchmarking the standard k-mer ML approach for repertoire classification with LIgO simulated data shows that they are unable to detect co-occurring immune signals. This demonstrates how LIgO can be used to assess the capacity and limits of specific approaches.

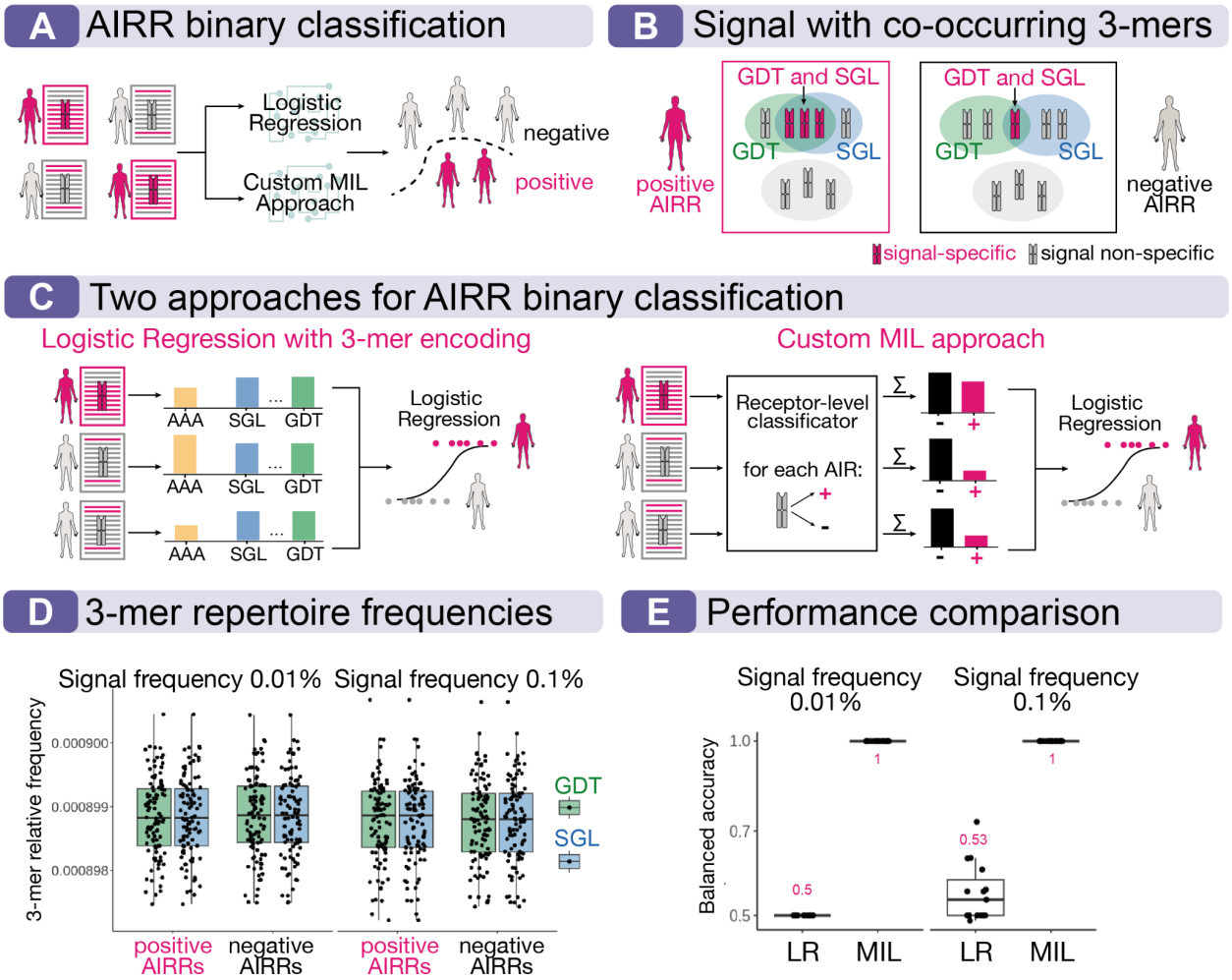


Figure 5: Use case 2: Limitations of the conventional k-mer encoding scheme for repertoire-level binary classification when immune signals co-occur within the same AIR

(A) We demonstrate the limitations of conventional k-mer encoding for a binary AIRR classification problem by utilizing LR and a custom multiple instance learning (MIL) approach. The task involves classifying repertoires as positive or negative based on the presence of an immune signal. Detailed methods can be found in the Methods section. (B) An example of a complex immune signal that may pose challenges to ML methods that neglect the co-occurrence of immune signals within the same AIR. The signal is composed of co-occurring GDT and SGL 3-mers within the same AIR and the frequency of such AIRs is higher in the positive repertoires. Moreover, the frequency of each 3-mer GDT and SGL remains identical in both the positive and negative repertoires, rendering it indistinguishable using the conventional k-mer encoding approach. We simulated two scenarios where positive repertoires consist of 0.01% and 0.1% of signal-specific receptors containing both GDT and SGL. For each scenario, we simulated 200 TRB repertoires (100 AIRRs in each class and 10^5 AIRs per repertoire) and replicated the simulation three times (1200 AIRRs and $1.2 \cdot 10^8$ TRBs in total). (C) We tested two MIL approaches for binary repertoire classification — (i) LR with conventional 3-mer encoding and a custom MIL approach. Briefly, the conventional LR approach (i) encodes every repertoire to a vector of normalized 3-mer frequencies across all AIRs within the repertoire. After that, vectors of 3-mers are classified using LR. The second custom MIL approach (ii) takes into account the 3-mer co-occurrence on the receptor level. First, every receptor is classified into signal-specific and non-specific based on the Fisher's exact test. On the second step, a LR classifies repertoires based on the number of signal-positive and negative receptors within the repertoire. (D) The relative frequency distribution (on the y-axis) of two 3-mers, GDT and SGL, that constitute the immune signal is shown for positive and negative repertoires and two different immune signal frequencies (0.01% and 0.1%). Note that the y-axis here represents normalized frequencies relative to the frequencies of all k-mers and is different from the immune signal frequency percentages. The visualization shows that the frequencies of the immune signal 3-mers is similar for positive and negative AIRRs when their joint co-occurrence is not considered. (E) Performance estimates (on the y-axis) of two different ML methods in a binary repertoire classification shown with balanced accuracy. As expected, the conventional LR with a 3-mer encoding performs poorly, achieving median balanced accuracy of 0.5 and 0.53 in two signal frequency scenarios. In contrast, the custom MIL, which considers the 3-mer co-occurrence, achieves a median balanced accuracy of 1 in both signal frequency scenarios.

Discussion

Simulations, as previously emphasized by Sandve and Greiff (*45*), enable the benchmarking of computational methods and are valuable in bioinformatics research, even in the presence of sufficient experimental data with known ground truth. Here, we developed the LIgO software suite for flexible simulation of AIRR data suitable for AIRR-ML benchmarking. While there exist other software suites for AIRR data simulation (*51*, *52*, *54*, *68*), these tools do not offer the possibility for configuring entire AIRR-ML benchmarking datasets for both repertoire and receptor-based classification tasks with complex and native-like immune signals (Supplementary Table 1). LIgO can be used together with the immuneML framework (*33*), enabling reproducible integration of simulation and subsequent AIRR-ML experiments (Supplementary Fig. 3).

To illustrate the usefulness of simulations for AIRR-ML research, we demonstrated LIgO on two use cases — (i) how train and test data split may drop the observed predictive performance in the out-of-distribution case, and (ii) how sequence encoding schemes can capture varying degrees of immune signal complexity. Using the ground truth information about immune signals present in LIgO-simulated AIRR data, we compared the immune information captured by the ML approaches to the original immune signals present in the data. The first use case demonstrated that the difference in underlying signal distribution between train and test sets may impact the ML method's observed performance. The second use case shows how LIgO simulations can be set up to reveal how a custom MIL method built based on a data assumption, like assuming that antigen specificity is determined by a pair of signals, indeed outperforms a standard approach if applied to data following this assumption. These use cases illustrate how LIgO can be used to investigate connections between the assumptions used for building an AIRR-ML method and the assumptions made about the AIRR data. These use cases are not yet feasible to reproduce on experimental data since ground truth information about immune signals present in every AIR is usually not known (*9*, *66*). The two demonstrated use cases are only example usages of the LIgO package. In addition, LIgO can also be utilized for benchmarking other AIRR methods, such as evaluating the performance of antigen-specific AIR clustering approaches (Valkiers et al. 2021; Rognes et al. 2022; Chang et al. 2023).

A main difficulty with AIRR simulations suitable for AIRR-ML benchmarking is the poorly described nature of immune signals. Although it is widely agreed upon that immune signals exist on the frequency level (clonal expansion) (*1*), sequence level (binding and structure-related sequence motifs) (*3*, *4*), and structure level (*3*), their antigen- or disease-specific shape remains largely uncharacterized. Therefore, simulation frameworks that aim to enable real-world relevant ML benchmarking need to encompass a large array of potential immune signal definitions and configurations to cover a broad area of the potential biological immune signal space. To this end, LIgO immune signals may be defined through gapped k-mers or motifs in general (*4*, *20*), repertoire generation model (*61*), clonal expansion (*1*), or a combination thereof. Additionally, LIgO offers the flexibility for immune signals to take any shape when defined through a custom function that indicates the signal's presence in an AIR. Such functionality was lacking in previous simulation tools (Supplementary Table 1). Although LIgO offers limitless possibilities for defining immune signals, it remains the user's responsibility to define an immune signal that comprehensively captures biological complexity and reflects underlying assumptions of an AIRR-ML model to be validated. Particular care is taken to ensure that immune signals do not break underlying AIR biology, such as AIR germline-originating subsequences and the generation probability distribution. We demonstrated that

signal implantation coupled with importance sampling may achieve a similar pgen distribution to rejection sampling while keeping computational efficiency. The flexibility of the presented approach allows users to define immune signals according to the state of understanding in the field at any point in time. Furthermore, helper functions guide the user as to which signal definitions are possible within a reasonable simulation time window.

In the future, large-scale antigen-specific AIRR data both on the receptor and repertoire level will be necessary to understand immune signals better. More specifically, on the repertoire-level, the dynamics of repertoire structure as a function of antigen exposure require further investigation (*82*). On the receptor level, we need to better understand the long-range dependencies that govern paratope-epitope interaction, using structural (*3*) and deep-mutational scanning screens (*14*). Although LIgO does neither leverage HLA-background information for TCR repertoire simulation (*83*) nor the influence of IG/TRB germline gene background on AIRR diversity (*61*, *84–87*), such influence may be modeled similarly to immune states with respect to the underlying causal model as recently shown by Pavlović and colleagues (*46*).

While LIgO encompasses many necessary features for comprehensive and exhaustive AIRR-ML simulation, possible improvements are: (i) integrate structure-based features enabling implicit and nature-like inclusion of long-range dependency signals (*66*). (ii) Combine LigO with sequencing read simulators (*88*, *89*) to incorporate PCR or sequencing errors. This type of error simulation is essential for studying artificially induced diversity versus biological clonal expansion and batch effects (*46*, *90*). (iii) Moreover, realistic simulation of cross-reactivity (i.e., the simulation of multiple immune signals per sequence) and chain pairing require further biological insights. (v) Finally, simulations on the repertoire level may be refined to render the simulation of public clone occurrence more native-like (*62*), as well as by (vi) allowing for the simulation of somatic hypermutation (*91–93*) and (vi) associated phylogenetic lineage trees (*55*, *94–97*). There is a potential risk that an AIRR-ML model trained on simulated data may only learn the artifacts of the simulation framework, thereby impacting the applicability of ML model-related insights to real-world scenarios. This may be addressed in the future by analyzing how ML results on synthetic data transfer to experimental data as recently demonstrated by Robert et al. (*66*). Further work is needed on the biological understanding of immune signals for improved simulation (*38*, *63*). Moreover, current speed bottleneck computations, such as those of generation probability, may be alleviated by novel encoding algorithms (*98*)), thus improving overall simulation speed.

Future progress in AIRR-ML research is required in areas as diverse as evaluating the impact of sample size on prediction accuracy (*33*, *63*), negative dataset definition (*44*, *72*, *73*), and unbiased estimation of prediction accuracy (*74*, *99*). For all of these use cases, LIgO simulations may be employed for benchmarking and developing interpretable AIRR-ML methods.

Methods

Simulation of background AIRs

LIgO supports two sources for background AIRs — (i) simulation of synthetic AIRs using a V(D)J recombination model, or (ii) processing experimental or any other set of AIRs provided by the user. The chosen source of AIRs will also be used during the simulation of immune-signal-specific AIRs.

To simulate naive synthetic AIRs, LIgO uses the OLGA tool (*68*) according to a predefined V(D)J recombination model (OLGA model), but, in principle, any other V(D)J recombination simulator such as IGoR (*51*) or immuneSIM (*52*) can be used. Briefly, OLGA describes a V(D)J recombination using a stochastic process, whose parameters include the distribution of V genes (or alleles), a conditional probability of J genes (or alleles) given a V gene (or allele), and others. OLGA produces AIRs in the format of a V gene (or allele) name, CDR3 sequence (nucleotide and amino acid), and a J gene (or allele) name, for example, TRBV3*01 + CASSLGGVGYEQYF + TRBJ2-7*01. LIgO supports the usage of the default OLGA models with parameters estimated using IGoR (*51*). This includes human AIRs (IGH, IGL, IGK, TRA, TRB) and mice AIRs (TRA, TRB). Alternatively, the user may specify a custom OLGA model. For example, a user can simulate mosaic deletion patterns if they assign (conditional) probabilities of zero to several gene alleles (*100*).

The second option for the simulation of background AIRs is a user-provided set of annotated AIRs in the following formats — AIRR Community standard format (*101*), MIXCR (*102*), 10x Genomics (*103*), Adaptive Biotechnologies ImmunoSEQ (*2*, *104*), or VDJdb formats (*105*). In this case, LIgO will iterate through every AIR once until the set of AIRs is empty. If the user-provided set of AIRs is not large enough to complete the simulation, LIgO will inform the user that a larger pool of background AIRs is needed to complete the simulation.

Representation of motifs

LIgO allows for defining motifs in three different ways (Fig. 1C) — k-mers, gapped k-mers, and PWMs (positional weight matrices). K-mers and gapped k-mers might be more convenient for initial proof-of-concept simulations as they are more straightforward to define than PWMs.

1. A *k-mer* is defined as a nucleotide or amino acid subsequence of length k, for example, AAA. Theoretically, a k-mer can be an entire CDR3 region. LIgO can also introduce several mutations to a k-mer defined by a Hamming distance range, i.e., the number of allowed mutations from the original k-mer to the mutated k-mer. For example, a k-mer AAAA with additional hamming distance range from 1 to 2 corresponds to a set of 4-mers that contain two or three As and the other positions may be any nucleotide or amino acid.

2. A *gapped k-mer* is defined as a nucleotide or amino acid subsequence that contains k nucleotides or amino acids and gap(s). In LIgO, the gap is allowed only in one position and a gapped k-mer is parameterized with a subsequence, gap position, and gap length range. For example, a gapped 3-mer AA/A with a gap range from 1 to 3 encodes the subsequences AA-A, AA--A, and AA---A, where a gap, denoted with "-", corresponds to one arbitrary nucleotide or amino acid. Similarly to k-mers, LIgO supports the definition of hamming distance with a gapped k-mer.

3. A *PWM of length k is* defined as a joint probability distribution of k independent discrete distributions over nucleotides or amino acids. Every PWM column describes a distribution for a given position of a motif. A k-mer is a special case of a PWM where the values in every column are zero except for one nucleotide or amino acid that is 1. A gapped k-mer is a PWM whose values in every column are either 0 or 1 in the positions where there is no gap, and uniformly distributed among nucleotides or amino acids in the gap positions.

A critical concept for LIgO simulation is defining whether a given AIR contains a motif. If a motif is a k-mer or a gapped k-mer, an AIR must include this (gapped) k-mer as a substring of the CDR3 region. A (gapped) k-mer or a PWM search in an AIR is implemented using BioNumPy (*106*) enabling efficient representation and operations with AIRs. Additionally, the search for (gapped) k-mers can accommodate mismatches, allowing for hamming distance consideration. If a motif is a PWM, LIgO constructs a regular expression based on the PWM and checks if an AIR matches the regular expression. For every subsequence matching the regular expression, LIgO calculates a PWM score, which equals its log-likelihood under the distribution described by PWM. Only those subsequences with a PWM score above a user-defined threshold are considered a match.

PWMs and (gapped) k-mers can describe various motifs in a sequence. However, using only one PWM, we cannot introduce dependencies between different positions in a subsequence because, by definition, the discrete distributions corresponding to every motif position are independent. Therefore several PWMs or (gapped) k-mers may have to be combined to describe an immune signal and capture the biological complexity. For example, if one were to define a motif A(A/C)(A/C) where only one C is allowed, one would have to combine two k-mers AAC and ACA.

Representation of immune signals

Immune signals (Fig. 1C) combine motifs with positional dependencies and immune-specific information. LIgO implements two types of immune signals: (i) a set of motifs that are (optionally) restricted with the V and J gene or allele names and the position(s) in the CDR3. For example, a set of motifs {AAA, CCC, A--CC} restricted by IGHV1*01 and IGHJ4, which occur only on the IMGT positions 108 or 110 (*107*). In this case, an AIR contains an immune signal if it contains any of the motifs on a particular CDR3 position and the correct V and J gene, if the IMGT position(s) or V/J genes are defined. (ii)

The second way to define an immune signal is a custom function indicating the signal's presence in an AIR. The function takes the amino acid and the nucleotide AIR sequence, its V and J gene and returns true if the AIR contains the desired signal and false otherwise. Such custom functions allow users to define any immune signal. For example, for an antigen-specific ML classifier that takes a sequence as input and produces the signal label as output (*14*, *72*, *108*), the user could write a custom function to call the classifier and define an immune signal based on the classifier's output.

Representation of immune events

Immune events (Fig. 1C), such as diseases, vaccinations, or allergies, elicit immune responses which change an individual's adaptive immune receptor repertoire. AIRRs with the same immune event might share similar immune signals involved in the immune response. More formally, LigO defines an immune event as a set of immune signals and their proportion in an AIRR. For example, we can define an AIRR

with T1D (type 1 diabetes) experience as containing 1% of receptors with corresponding signals. To reflect biological complexity, LIgO-simulated immune repertoires may contain multiple immune events.

For a receptor-level simulation, an AIR belongs to an immune event if it contains any of the signals for the immune event. Implementation-wise, the immune event for a single AIR is the next level of abstraction after the immune signal and is directly assigned to an AIR. It does not influence the AIR sequence in any way, but logically, it should follow the immune signals that are considered to belong to a given immune event. Since it does not influence the simulation, there might be as many immune event labels as desired. For example, we can regard an immune event T1D as a set of two immune signals {signal1 = "AAA", signal2 = "GGG"}. In this case, an AIR (TRBV13*01+ CASSAGGGAFYGYTF + TRBJ1-2*01) will be signal2-specific because it contains "GGG". The user may then choose to explicitly assign the "T1D" label to the receptor. From a practical standpoint, a user would define a group of receptors with a particular set of simulation parameters and labels.

Generalization of the immune signal concept to paired-chain AIRR data

Both BCRs and TCRs are (generally) composed of two protein chains — heavy and light chains for BCRs and alpha/beta or gamma/delta chains for TCRs. Until recently, most studies focused on the heavy chain of BCRs and beta or delta chains of TCRs (*109*, *110*). However, the pairing information is important for studying AIR diversity (*111*) and antigen binding (*112*). Recently developed single-cell AIRR-seq technologies enable accurate identification of paired AIRs (*113*, *114*). Currently, it remains unclear whether the pairing rules that combine two chains in one receptor follow a uniform distribution or not, and until now, there are no studies that were able to infer the pairing rules (*115*–*118*). Furthermore, there is little evidence of major structural constraints in chain pairing (*119*).

The LIgO simulation framework is based on the simplifying assumption that an AIR is independently formed by two chains, which allows processing these two chains separately and defining an immune signal of a paired AIR as immune signals of each two receptor sequences. We define that a paired chain is part of an immune signal if and only if at least one chain contains the immune signal. A special case of a paired immune signal is a pair of two signals where one or even two AIRs are predefined (i.e., a corresponding motif is a whole receptor). In this case, all the paired receptors that are part of this paired immune signal will share a similar or same light chain. A paired immune event is defined as a pair of a specificity label and a set of paired immune signals.

Simulation of paired signals is performed similarly to the single-chain signal simulation described above. Each chain is independently simulated using either rejection sampling or signal implantation. After that, single chains are grouped into pairs per user specification.

Immune receptor annotation with immune signals

LIgO annotates an AIR as containing an immune signal or not, via checking if any of the immune signals defined in the simulation occurs anywhere in the AIR where it's allowed (per IMGT positions if they are specified). If one AIR contains two overlapping motifs from different signals, the AIR is labeled as specific to both signals and removed if two signals per receptor are not specified for that simulation batch. If one AIR contains two motifs from the same signal, they are not considered separately and the AIR is labeled as specific to that signal.

Simulation of immune-signal-specific receptors using rejection sampling

Rejection sampling enables simulation of event-specific AIRs without changing their sequence. The rejection sampling procedure is performed iteratively. During each iteration, a batch of background AIRs is selected either from the background V(D)J model or experimental data. By default, the batch size is 10^4 AIRs. If an immune signal contains signals which are restricted with a specific V or J gene and the simulation is defined by a V(D)J model, LIgO uses a skewed V(D)J model to simulate AIRs containing only the predefined V and J genes to speed up the simulation process. After that, each AIR from the batch is annotated with all immune signals. AIRs containing more than two immune signals are eliminated. AIRs containing one immune signal or a combination of two immune signals (as per user specification) are added to a pool of immune-signal-specific receptors. The process is repeated until the desired amount of receptors is not simulated for all immune signal(s) or the number of iterations exceeds the upper limit, which is 100 by default.

Simulation of immune-signal-specific receptors using signal implantation

Signal implantation is the fastest way to simulate immune-signal-specific receptors. However, this method is limited to immune signals which are defined as either a (gapped) k-mer or PWM. This limitation arises because implantation of other signal components such as a V gene or CDR3 length, may impact an AIR sequence structure. Once the user defines the desired amount of AIRs for each immune signal, LIgO obtains this specified amount of background receptors from either a recombination model or user-defined experimental data. After that, the corresponding (gapped) k-mer or a substring generated by a PWM is implanted to each AIR, respectively. Implantation can be performed at any position of the CDR3 region in such a way that the implanted subsequence does not increase the length of a given AIR. The implantation position can be uniformly distributed across the CDR3 or may be defined by the user according to a distribution across IMGT positions. For instance, positions in the center of the CDR3 might have a high probability of implantation, while positions in the conserved regions such as the beginning and the end of the CDR3 might exhibit negligible probability.

Generation probability evaluation

When a LIgO simulation is defined by an OLGA V(D)J recombination model (*51*, *68*), it enables the computation of generation probability for all AIRs involved in the simulation process with respect to this specific model. The generation probability (pgen) of an AIR estimates the probability of a given OLGA model to generate a given AIR (*68*). Pgen is calculated as the cumulative probability across all potential recombination scenarios capable of producing this AIR. While generation probability can provide valuable insights into AIRR analysis (*51*, *61*, *120*, *121*), it is worth noting that pgen computation might lead to an increase in overall simulation time due to the slower nature of OLGA's pgen calculation.

LIgO does not support pgen evaluation if a simulation is defined via user-provided experimental data. In this case, the corresponding recombination model is not specified since experimental data may be merged from multiple individuals and thus align with distinct recombination models (*61*).

Importance sampling

If a LIgO simulation is defined by an OLGA V(D)J recombination model, LIgO supports an importance sampling approach, which preserves the pgen distribution of a simulated AIRR close to the background pgen distribution. First, LIgO estimates the pgen distribution of background AIRs through a histogram of

logarithmically transformed pgens of a large set of background AIRs. To do so, LIgO generates a batch of background AIRs, and divides these AIRs into equal-width histogram bins (in log-scale), where the number of bins must be provided by the user. Next, an acceptance probability is assigned to each bin of the histogram. The acceptance probability is proportional to how often we observe background AIRs falling into the pgen range corresponding to the histogram bin. If there are any pgen bins without observed AIRs, they are set to a predefined minimum acceptance probability of 10^{-5}. All these acceptance probabilities are then normalized to sum to 1.

During the simulation of event-specific AIRs, LIgO generates an AIR either with rejection sampling or signal implantation and calculates the logarithm of its pgen. This log-pgen is then matched with the corresponding bin in the histogram and the acceptance probability associated with this bin. Subsequently, the AIR is accepted into the final dataset with a probability equal to the acceptance probability. This process is continued until the desired amount of event-specific AIRs has been simulated.

Assessment of simulation feasibility

To evaluate the simulation feasibility for a given set of parameters, LIgO generates a batch of background AIRs, with the default batch size being 10^5 AIRs. After that, each AIR is annotated with all immune signals participating in the simulation process and AIRs containing more than 2 immune signals are filtered out. The ratio of immune-signal specific receptors to the batch size estimates the immune signal frequency among the background receptors. If the estimated frequency of an immune signal is excessively low (less than 0.001%), then it is not recommended to use rejection sampling for simulating this immune signal. This is due to the number of background receptors that must be generated during rejection sampling to find at least one event-specific AIR is inversely proportional to the background frequency of this immune signal. Conversely, if the estimated frequency is too high (more than 10%), then such an immune signal may lack biological relevance and might cause an increased number of simulation iterations, if the user wants to add background receptors without any immune signals to the final dataset.

LIgO also estimates an immune signal co-occurrence matrix, which represents the conditional probability of observing one immune signal within an AIR given the presence of another immune signal. This conditional probability of an AIR containing immune signal 2 given the presence of immune signal 1 is calculated as a ratio of the number of AIRs within the original batch containing both immune signal 1 and immune signal 2 and the number of AIRs containing only immune signal 1. For users considering rejection sampling while ensuring each AIR holds no more than one signal, the co-occurrence matrix indicates which immune signals have high chances of co-existing in one AIR. These immune signals are recommended to be re-defined. Alternatively, users can use implantation instead of rejection sampling since it can restrict the number of immune signals to one for each AIR.

Clonal count simulation

In a repertoire-based scenario, LIgO offers an optional simulation of clonal counts for each AIR. These counts follow the Zeta distribution, which is a discrete analog of the power law distribution. The realization of the Zeta distribution is performed using the Zipf function from the scipy.stats library (*122*), which requires two parameters — *a* (shape parameter, where a≥1) and *loc* (shift parameter). For each immune signal and the background AIRs, the user needs to define *a* and *loc* parameters. By adjusting these parameters across different immune signals, one can simulate clonal expansion of AIRs containing these specific signals (*1*).

Identification of amino acid 3-mer frequency distribution patterns in CDR3 regions of synthetic IGHs and TRBs

To demonstrate the natural distribution of 3-mers within CDR3 regions, we used LIgO to simulate a dataset of 10^6 amino acid background TRBs using the default IGoR generation model (*51*, *68*), where each CDR3 has a restricted length of 15 amino acids. After that, we calculated how often each 3-mer appears on each CDR3 position among simulated TRBs using immuneML (*33*). We only considered 3-mers starting at the IMGT positions 105–115, because the positions 104 and 118 are conserved and thus deliberately excluded from the analysis. We also eliminated from the analysis all 3-mers that were present in less than 10 TRBs out of the total amount of 10^6 simulated TRBs. For each 3-mer we normalized the frequencies of this 3-mer across positions 105–115, ensuring that the total sum equals one. Normalized frequency vectors were then clustered using hierarchical clustering with complete linkage method and Euclidean distance between vectors. The hierarchical clustering dendrogram was split into 10 clusters and for each cluster, we calculated the mean frequency and 95% confidence interval among all 3-mers belonging to this cluster for all IMGT positions 105–115. Nine clusters out of ten (clusters 2-10) containing 48.69% of amino acid 3-mers in total have a peak of average frequency, i.e. average frequency > 0.4, at one of the IMGT positions 105–108 or 111–115 (Fig. 3). One cluster (cluster 1) containing 59.92% 3-mers has no average peaks.

The same analysis was also performed on 10^6 amino acid IGHs, see Supplementary Fig. 2 for more details. Three out of ten clusters (clusters 1, 3, 4, 79.98% of 3-mers) do not contain a frequency peak on average at any IMGT position, seven out of ten clusters (clusters 2, 5–10, 19.85% of 3-mers) have an average peak at one of the IMGT positions 105–108 or 111–115.

Signal implantation decreases the generation probability of a given receptor

We conducted a comparison of pgens before and after implantation across four different signals, where one signal is defined as a set of ten random amino acid k-mers of the same length; see Supplementary Table 2 for the list of used k-mers. We simulated four TRB datasets, each corresponding to a different k-mer size, specifically k = 2, 3, 4, and 5. For each k-mer, we simulated 100 signal-specific amino acid TRBs using implantation and the default IGoR model (*51*, *68*), resulting in a total of 1000 TRBs for each individual dataset. The implanting positions were constrained to ensure that each implanted k-mer does not affect the first and the last two amino acids of the original CDR3 sequence. Using the default IGoR model (*51*, *68*), we computed a pgen difference for each TRB by subtracting the log10-transformed pgen after implantation from the log10-transformed pgen before implantation. Receptors with a post-implantation pgen value of zero were excluded from the analysis, resulting in removing 0.8% of TRBs for 2-mers, 0.7% of TRBs for 3-mers, 2.4% of TRBs for 4-mers, and 3% of TRBs for 5-mers. The majority of TRBs exhibited a positive pgen difference, with percentages as follows: 95.1% for 2-mers, 96% for 3-mers, 97% for 4-mers, and 93.4% for 5-mers. The median pgen difference values were 2.46 for 2-mers, 3.04 for 3-mers, 4.14 for 4-mers, and 4.05 for 5-mers. This indicates that, on average, the pgen of a receptor after implantation may experience a reduction of between two and four orders of magnitude, depending on the length of the k-mer. Similarly, we calculated pgen difference when immune-signal-specific AIRs are generated using signal implantation coupled with importance sampling, where the background pgen distribution was estimated using 10^4 TRBs splitted into 30 histogram bins. In this scenario, the average pgen difference decreased compared to signal implantation performed on its

own. The median pgen difference values were 1.77 for 2-mers, 2.5 for 3-mers, 3.36 for 4-mers, and 3.46 for 5-mers.

Implantation coupled with importance sampling yields an overall similar pgen distribution to rejection sampling for signal-specific AIRs

We compared the generation probability distribution of signal-specific AIRs generated using three simulation approaches — rejection sampling (RS), implantation (I), and implantation coupled with importance sampling (I + IS). The immune signal was defined as a set of ten random k-mers of the same length, where k = 2, 3, 4, and 5; see the detailed description of the immune signals and the simulation of signal-specific AIRs using rejection sampling and signal implantation in the previous section. To simulate signal-specific AIRs using implantation coupled with importance sampling, we constructed a histogram of logarithmically transformed pgens using 30 bins and 10^4 background TRBs simulated with the default IGoR model (*51*, *68*). Next, we evaluated acceptance probabilities for each bin and continued the simulation until the number of accepted AIRs for each k-mer was 100. The pgen distribution of signal-specific AIRs generated using implantation coupled with importance sampling is overall closer to the rejection sampling pgen distribution, which is the ground-truth pgen distribution of signal-specific AIRs in logarithmic scale. The median of the log-pgen distribution for each method are: $3.5 \cdot 10^{-15}$ (I), $8.1 \cdot 10^{-13}$ (RS), $1.2 \cdot 10^{-12}$ (I+IS) for 2-mers; $1.9 \cdot 10^{-15}$ (I), $1.4 \cdot 10^{-13}$ (RS), $1.4 \cdot 10^{-13}$ (I+IS) for 3-mers; $1.4 \cdot 10^{-16}$ (I), $3.3 \cdot 10^{-15}$ (RS), $1.8 \cdot 10^{-14}$ (I+IS) for 4-mers; and $2.7 \cdot 10^{-16}$ (I), $5.6 \cdot 10^{-15}$ (RS), $1.9 \cdot 10^{-14}$ (I+IS) for 5-mers.

Use case 1: Out-of-distribution receptor-level simulation using LIgO

To demonstrate how different train-test split strategies may impact the prediction accuracy and the optimal model, we simulated a dataset of $4 \cdot 10^4$ IGH amino acid sequences in 10 replicates ($4 \cdot 10^5$ IGHs in total) using signal implantation and the default IGoR IGH V(D)J recombination model (*51*). Each dataset of $4 \cdot 10^4$ IGH sequences represented a set of AIRs obtained from four different individuals, where every individual contributed $5 \cdot 10^3$ signal-specific AIRs and $5 \cdot 10^3$ non-specific AIRs to the dataset. The immune signal was defined as a set of four similar 4-mers — AAAA, AAGA, AACA, and AANA, and we assumed that every individual carries only one signal 4-mer out of four. Altogether, one dataset contained 10^4 IGH sequences from person 1 ($5 \cdot 10^3$ IGH sequences with AAGA, $5 \cdot 10^3$ IGH sequences without any of four signal 4-mers), 10^4 IGH sequences from person 2 ($5 \cdot 10^3$ IGH sequences with AACA, $5 \cdot 10^3$ IGHs without without any of four signal 4-mers), 10^4 IGH sequences from person 3 ($5 \cdot 10^3$ IGH sequences with AANA, $5 \cdot 10^3$ IGH sequences without without any of four signal 4-mers), and 10^4 IGH sequences from person 4 ($5 \cdot 10^3$ IGH sequences with AAAA, $5 \cdot 10^3$ IGH sequences without without any of four signal 4-mers), where non-specific IGHs did not contain any of the four signal 4-mers.

We used a logistic regression model to perform a binary receptor-based classification task, i.e., classify AIRs into a signal-specific class or non-specific class. You can find more details about our usage of logistic regression in the "Machine Learning" methods section. The logistic regression model was trained and tested using the data described above and two train-test split approaches — (i) random train-test split and (ii) leave-one individual out train-test split. For the random train-test split strategy, $4 \cdot 10^4$ IGH sequences were randomly split into 75% train and 25% test subsets four times for the model assessment on the outer cross-validation loop, and every time within the inner cross-validation loop, threefold cross-validation was used for the model (e.g., hyperparameter) selection. For the leave-one-individual out procedure, we split 10^4 AIRs from one individual to the test set and the rest $3 \cdot 10^4$ AIRs to the train set for the model assessment (outer) cross-validation loop and this was repeated for every individual. In the

model selection (inner) cross-validation loop we again split 10^4 AIRs from one individual to the test set, and the rest $2 \cdot 10^4$ AIRs to the train set and repeated this procedure for every individual.

We benchmarked the logistic regression model from the scikit-learn package (*123*) using immuneML (*33*). We encoded the IGH sequences using 4-mers encoding, where each vector of 4-mer frequencies was normalized by scaling its variance to 1, and mean to zero across different IGH sequences. We used two train-test strategies and performed hyperparameter optimization for regularization parameters (L1 and L2), and the regularization constant (0.01, 0.05, 0.1, 1), where the optimized performance metric during model training was balanced accuracy. We repeated the logistic regression training 10 times for every dataset replica.

For both train-test split strategies, the optimal balanced accuracy was achieved on the L1 regularization and the regularization constant 0.1. However, the model trained using the random train-test split strategy, achieved a mean balanced accuracy of 0.98, and the model trained using the leave-one-individual out achieved a mean balanced 0.67. We examined the ability of both logistic regression models to recover the ground-truth implanted 4-mers. We considered the largest absolute coefficients of the optimal logistic regression models, and the random LR models recovered all four signal 4-mers, while the leave-one-individual out models only recovered three signal 4-mers (AAGA, AANA, AACA) out of four.

Use case 2: Limitations of conventional encoding schemes for repertoire-level binary classification when immune signals co-occur within the same AIR

We simulated 200 TRB repertoires (100 repertoires in each class and 10^5 amino acid AIRs in each repertoire) in six replicates (three replicates for two signal frequencies 0.01% and 0.1%). We defined an immune signal as a combination of two 3-mers GDT and SGL. Both GDT and SGL occur in the naive AIRs simulated via the default IGoR TRB model (*51*) at approximately 0.01% frequency and co-occur in the same AIR at approximately 0.00001% frequency. Using rejection sampling, we simulated positive class AIRRs containing 0.01% or 0.1% of signal-specific receptors (containing both GDT and SGL), and the negative class containing 0.001% and 0.01% signal-specific receptors, respectively. Additionally, AIRRs from the positive and the negative classes had similar frequency of each GDT and SGL separately, which made the classes indistinguishable for the classical k-mer encoding.

Using the simulated data described above, we compared two multiple instance learning (MIL) approaches — logistic regression and a custom approach that takes into account signal co-occurrence. The first ML method has been described extensively elsewhere (Kanduri et al. 2022). Briefly, an L1-regularized logistic regression model trained on k-mer abundance-encoded data (k=3 here) was used, whose hyperparameters are optimized through nested cross-validation. Note that this method treats each k-mer in isolation and does not consider combinations. The second ML method (custom MIL) first classifies each receptor (instance) as whether immune state-associated or not and then sums up the number of positive instances per repertoire. The number of positive instances per repertoire is used as the single feature for a logistic regression. The identification of positive instances relied on the presence of pairs of 3-mers (combinations) that are together over-represented in positive class labeled receptors relative to the negative class labeled receptors as assessed through a Fisher's exact test. For each 3-mer pair, the contingency table for Fisher's exact test is composed of (a) the number of receptors of positive AIRRs that contain the 3-mer pair, (b) the number of receptors of positive AIRRs that do not contain the 3-mer pair, (c) the number of receptors of negative AIRRs that contain the 3-mer pair, and (d) the number of receptors of negative AIRRs that do not contain the 3-mer pair.

Machine Learning

We utilized Logistic Regression for binary classification of signal-specific repertoires and receptors, which was similar to our previous studies (*33*, *35*). Briefly, we represented AIRs using k-mer encoding and with further standardization by subtracting the mean and dividing by the standard deviation. After that, we optimized LR hyperparameters, such as L1 and L2 regularization and regularization strength constant, using nested cross-validation. We evaluated performance of ML models using three metrics — precision ($\frac{TP}{TP+FP}$), recall ($\frac{TP}{TP+FN}$), and balanced accuracy ($\frac{precision+recall}{2}$), where TP is true positive, FP is false positive and FN is false negative. Model training, selection, and evaluation was performed using the immuneML platform (*33*) (version 2.2.3). See the Use cases methods sections for a more comprehensive description of classifiers used in every use case.

Code availability

LIgO is freely available as a Python package from PyPI (https://pypi.org/project/ligo/), on GitHub (https://github.com/uio-bmi/ligo), and as a Docker image (https://hub.docker.com/r/milenapavlovic/ligo).

Declaration of interests

V.G. declares advisory board positions in aiNET GmbH, Enpicom B.V, Specifica Inc, Adaptyv Biosystems, EVQLV, Omniscope, Diagonal Therapeutics, and Absci. V.G. is a consultant for Roche/Genentech, immunai, Proteinea, and LabGenius. P.M. holds shares in ImmuneWatch BV. The remaining authors declare no competing interests.

Funding

We acknowledge generous support by The Leona M. and Harry B. Helmsley Charitable Trust (#2019PG-T1D011, to VG), UiO World-Leading Research Community (to VG), UiO:LifeScience Convergence Environment Immunolingo (to VG, GKS, and IHH), EU Horizon 2020 iReceptorplus (#825821) (to VG), Research Council of Norway projects (#300740, 331890 to VG), a Research Council of Norway IKTPLUSS project (#311341, to VG and GKS), a Norwegian Cancer Society Grant (#215817, to VG), and Stiftelsen Kristian Gerhard Jebsen (K.G. Jebsen Coeliac Disease Research Centre) (to GKS), UiO:LifeScience Convergence Environment RealArt (to GKS, CK). This project has received funding from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 101007799 (Inno4Vac). This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and EFPIA (to VG). SG was supported by the Research Foundation Flanders (FWO) through an SB fellowship (1S48819N) and a travel grant for a research stay (V403821N). This work was carried out on immunohub e-Infrastructure funded by the University of Oslo and jointly operated by GreiffLab and SandveLab (the authors) in close collaboration with the University Center for Information Technology, University of Oslo, IT-Department (USIT).

Supplementary materials

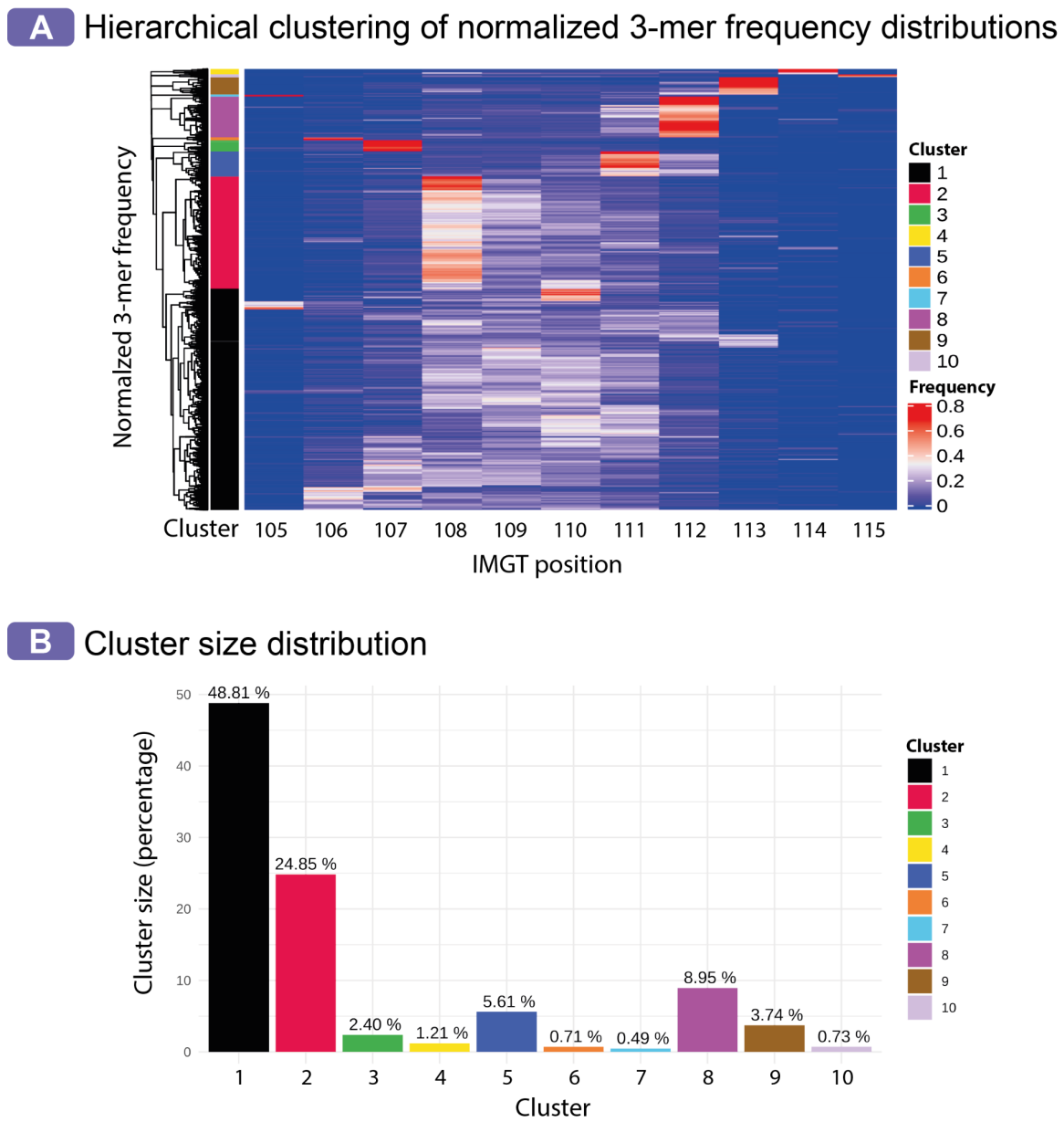
Supplementary Table 1: Comparison of LIgO with other AIRR simulation tools — IGoR/OLGA, immuneSIM, and simAIRR.

	LIgO	IGoR/OLGA	immuneSIM	simAIRR
General software description				
Reference	This work	(*51*, *68*)	(*52*)	(*62*)
Purpose	Simulation of AIRR with complex immune signals for ML benchmarking (both receptor and repertoire-based)	Simulation of a set of AIRs based on a V(D)J recombination model	Simulation a set of AIRs based on a V(D)J recombination model; simulating signals (gapped k-mer only) in individual AIRs	Simulation of antigen-experienced -like AIRRs by mimicking the relation between pgen and public AIR occurrence in real-world datasets
Software	Python package	C++, Python package	R package	Python package
Simulation				
User assistance	Yes	No	No	Yes
Support of both receptor and repertoire-level simulation	Receptor and repertoire level simulation	Receptor level only	Receptor level only, but supports clone counts	Repertoire level only
Simulation of single-chain and paired chain data	Yes	No	Yes	No
Immune signal complexity	Any signal complexity including (gapped) k-mers, PWMs, and user-defined signals	No	Set of k-mers	Any user-defined signals supplied in the form of full-length AIRs; strictly assumes that signal sequences are public
Simulation approach	Rejection sampling and signal implantation	No	Signal implantation	Implantation of full-length CDR3 amino acid sequences + V and J genes of AIRs into repertoire(s)
Simulates clonal expansion	Yes, somewhat (each signal and background set of	No	Yes, somewhat (AIR clone count distribution may be	No

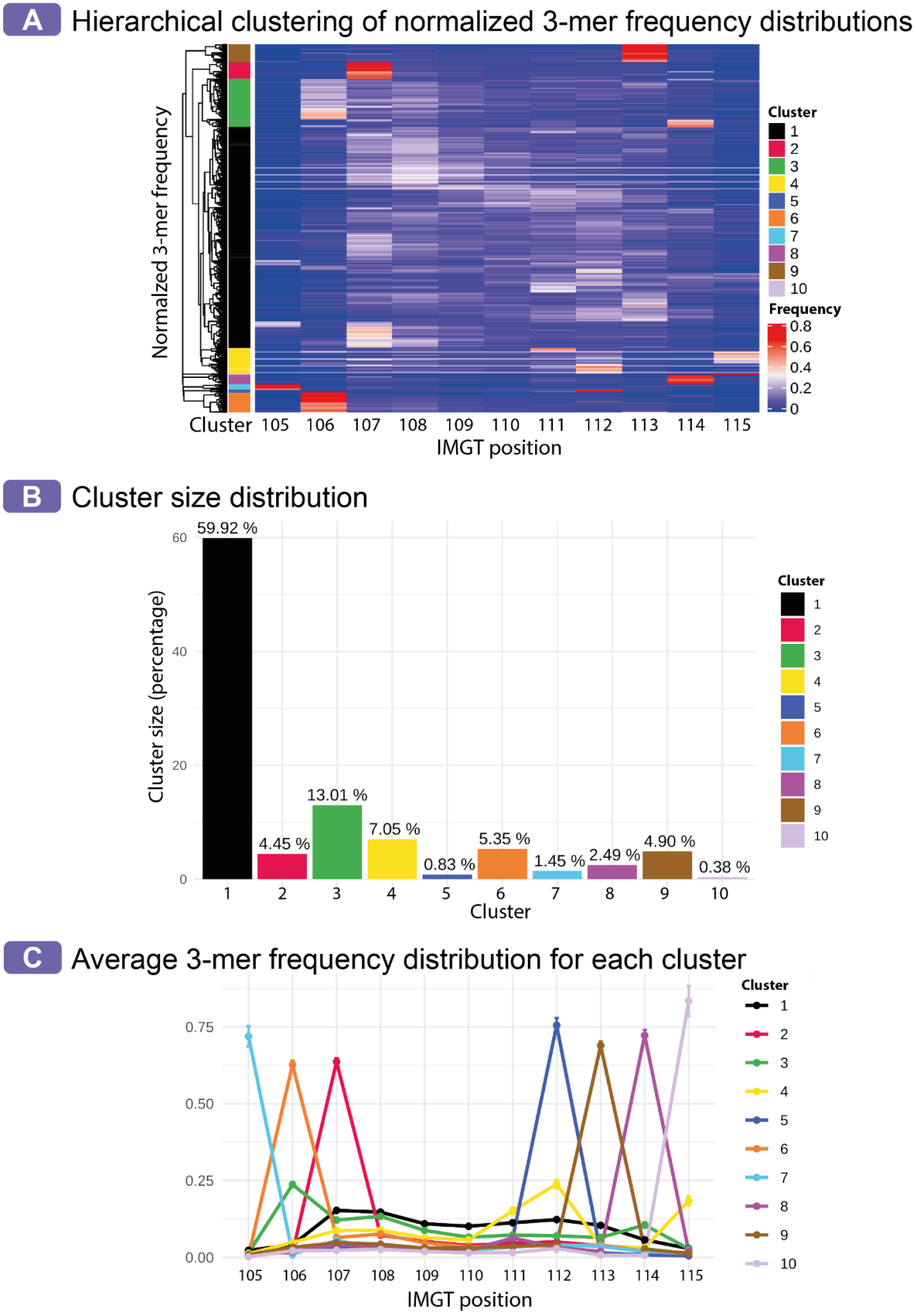
	AIRs may have its own clone count distribution)		simulated)	
Methods for preserving biological nativeness of simulated data				
Simulation based on experimental AIRR data	Yes	No, but a custom VDJ model can be used	No, but a custom VDJ model can be used	Yes
Preserves generation probability distribution	Yes, using importance sampling	Not applicable, the tools do not support repertoire-level simulation	No	No
Realistic frequencies of public AIRs in repertoire-level data when needed	No, but can be used in combination with simAIRR (see column 4 of this table)	Not applicable	Not applicable	Yes
Output				
Reporting of detailed information about immune signals and the simulation	Yes	No	Yes	No
AIRR-compliant output format	Yes	Yes, using pygor (https://github.com/statbiophys/pygor3)	Yes	No, only OLGA format

Supplementary Table 2 (relates to Fig. 3) List of random 2-mers, 3-mers, 4-mers, and 5-mers used for LIgO simulations.

2-mer	3-mer	4-mer	5-mer
FF	KTC	NDMH	PVWGS
TQ	VED	VHWR	LINHT
CW	FFC	HDPR	RTNHL
LT	FPY	NSNE	QHTSV
DQ	CTR	YLYW	WGSDT
KT	VMV	CHPQ	TKDKF
WE	GHT	CCYN	FWMPR
PM	KKY	VFDP	PYLCT
TC	NAM	CTQA	TRYHY
FG	MQP	VIYM	GRLGL

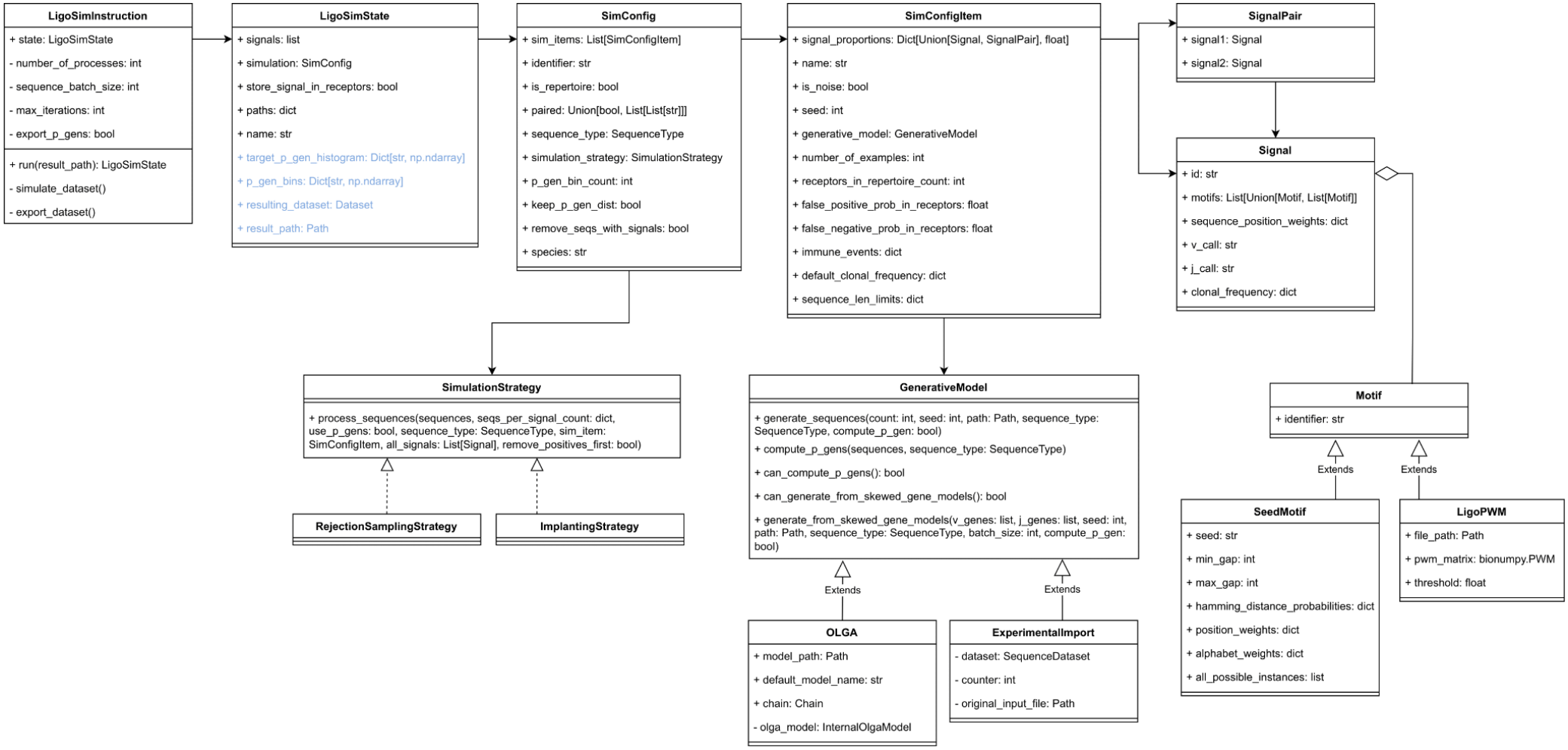


Supplementary Figure 1 (relates to Fig. 3) Analysis of 3-mer occurrence frequency distribution clusters based on 10^6 synthetic TRB CDR3 amino acid sequences of length 15aa. (A) Hierarchical clustering of 3-mer frequencies, where each row of the heatmap corresponds to the frequency of one 3-mer, and each column corresponds to one IMGT position. Each row (3-mer) of the heatmap is normalized such that the sum of frequencies of this 3-mer on all IMGT positions (105–115) equates to one. (B) Out of 8000 of all possible amino acid 3-mers, 48.81% of 3-mers (cluster 1) do not contain a frequency peak on average at any IMGT position, 48.69% of 3-mers (clusters 2-10) contain average frequency peak, i.e. average frequency > 0.4 at one of the IMGT positions 105–108 or 111–115, and 2.5% of 3-mers were eliminated from the analysis because they were found less than ten times in 10^6 simulated TRB sequences.



Supplementary Figure 2 (relates to Fig. 3) Analysis of 3-mer occurrence frequency distribution clusters based on 10^6 synthetic IGH CDR3 amino acid sequences of length 15aa. (A) Hierarchical clustering of 3-mer frequencies, where each row of the heatmap corresponds to the frequency of one 3-mer, and each column corresponds to one IMGT position. Each row (3-mer) of the heatmap is normalized such that the sum of frequencies of this 3-mer on all IMGT positions (105–115) equates to one. (B) Out of 8000 of all possible amino acid 3-mers, 79.98% of 3-mers (clusters 1, 3, 4) do not contain a frequency peak on average at any IMGT position, 19.85% of 3-mers (clusters 2, 5–10) contain an average frequency peak, i.e. average frequency > 0.4 at one of the

IMGT positions 105–108 or 111–115, and 0.17% of 3-mers were eliminated from the analysis because they were found less than ten times in 10^6 simulated TRB sequences. (C) We calculated the average frequency of all 3-mers at every IMGT position (105–115) within each cluster. One line represents one cluster, the points correspond to mean values, and the error bars correspond to the 95% confidence interval.



Supplementary Figure 3: Ligo UML class diagram. The attributes in blue are set automatically at runtime, the rest are provided by the user in the YAML specification or read from default values during parsing. The LIgO data/simulation model closely follows the YAML specification.

References

1. V. Greiff, P. Bhat, S. C. Cook, U. Menzel, W. Kang, S. T. Reddy, A bioinformatic framework for immune repertoire diversity profiling enables detection of immunological status. *Genome Med.* 7, 49 (2015).

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

3. R. Akbar, P. A. Robert, M. Pavlović, J. R. Jeliazkov, I. Snapkov, A. Slabodkin, C. R. Weber, L. Scheffer, E. Miho, I. H. Haff, D. T. T. Haug, F. Lund-Johansen, Y. Safonova, G. K. Sandve, V. Greiff, A compact vocabulary of paratope-epitope interactions enables predictability of antibody-antigen binding. *Cell Rep.* 34, 108856 (2021).

4. J. Ostmeyer, S. Christley, I. T. Toby, L. G. Cowell, Biophysicochemical Motifs in T-cell Receptor Sequences Distinguish Repertoires from Tumor-Infiltrating Lymphocyte and Adjacent Healthy Tissue. *Cancer Res.* 79 (2019), pp. 1671–1680.

5. E. L. Shrock, R. T. Timms, T. Kula, E. L. Mena, A. P. West, R. Guo, I.-H. Lee, A. A. Cohen, L. G. A. McKay, C. Bi, K. Null, Y. Leng, E. Fujimura, F. Horns, M. Li, D. R. Wesemann, A. Griffiths, B. E. Gewurz, P. J. Bjorkman, S. J. Elledge, Germline-encoded amino acid–binding motifs drive immunodominant public antibody responses. *Science*. 380, eadc9498 (2023).

6. R. A. Arnaout, E. T. L. Prak, N. Schwab, F. Rubelt, Adaptive Immune Receptor Repertoire Community, The Future of Blood Testing Is the Immunome. *Front. Immunol.* 12, 626793 (2021).

7. A. Vujkovic, M. Ha, T. de Block, L. van Petersen, I. Brosius, C. Theunissen, S. H. van Ierssel, E. Bartholomeus, W. Adriaensen, G. Vanham, G. Elias, P. Van Damme, V. Van Tendeloo, P. Beutels, M. van Frankenhuijsen, E. Vlieghe, B. Ogunjimi, K. Laukens, P. Meysman, K. Vercauteren, Diagnosing viral infections through T cell receptor sequencing of activated CD8+ T cells. *J. Infect. Dis.* (2023), doi:10.1093/infdis/jiad430.

8. V. Greiff, G. Yaari, L. Cowell, Mining adaptive immune receptor repertoires for biological and clinical information using machine learning. *Current Opinion in Systems Biology* (2020), doi:10.1016/j.coisb.2020.10.010.

9. R. Akbar, H. Bashour, P. Rawat, P. A. Robert, E. Smorodina, T.-S. Cotet, K. Flem-Karlsen, R. Frank, B. B. Mehta, M. H. Vu, T. Zengin, J. Gutierrez-Marcos, F. Lund-Johansen, J. T. Andersen, V. Greiff, Progress and challenges for the machine learning-based design of fit-for-purpose monoclonal antibodies. *MAbs*. 14, 2008790 (2022).

10. K. W. Wucherpfennig, P. M. Allen, F. Celada, I. R. Cohen, R. D. Boer, K. C. Garcia, B. Goldstein, R. Greenspan, D. Hafler, P. Hodgkin, E. S. Huseby, D. C. Krakauer, D. Nemazee, A. S. Perelson, C. Pinilla, Rol, K. Strong, E. E. Sercarz, Polyspecificity of T cell and B cell receptor recognition. *Semin. Immunol.* 19, 216–224 (2007).

11. P. A. Robert, A. L. Marschall, M. Meyer-Hermann, Induction of broadly neutralizing antibodies in Germinal Centre simulations. *Curr. Opin. Biotechnol.* 51, 137–145 (2018).

12. C. Garrett Rappazzo, M. L. Fernández-Quintero, A. Mayer, N. C. Wu, V. Greiff, J. J. Guthmiller, Defining and studying B cell receptor and TCR interactions. *The Journal of Immunology*. 211, 311–322 (2023).

13. C. T. Boughter, M. T. Borowska, J. J. Guthmiller, A. Bendelac, P. C. Wilson, B. Roux, E. J. Adams, Biochemical patterns of antibody polyreactivity revealed through a bioinformatics-based analysis of CDR loops. *Elife*. 9 (2020), doi:10.7554/eLife.61393.

14. D. M. Mason, S. Friedensohn, C. R. Weber, C. Jordi, B. Wagner, S. M. Meng, R. A. Ehling, L. Bonati, J. Dahinden, P. Gainza, B. E. Correia, S. T. Reddy, Optimization of therapeutic antibodies by predicting antigen specificity from antibody sequence via deep learning. *Nat Biomed Eng*. 5, 600–612 (2021).

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

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

17. A. Straub, S. Grassmann, S. Jarosch, L. Richter, P. Hilgendorf, M. Hammel, K. I. Wagner, V. R. Buchholz, K. Schober, D. H. Busch, Recruitment of epitope-specific T cell clones with a low-avidity threshold supports efficacy against mutational escape upon re-infection. *Immunity*. 56, 1269–1284.e6 (2023).

18. E. Dorigatti, F. Drost, A. Straub, P. Hilgendorf, K. I. Wagner, B. Bischl, D. H. Busch, K. Schober, B. Schubert, Predicting T Cell Receptor Functionality against Mutant Epitopes. *bioRxiv* (2023), p. 2023.05.10.540189.

19. V. Greiff, U. Menzel, E. Miho, C. Weber, R. Riedel, S. Cook, A. Valai, T. Lopes, A. Radbruch, T. H. Winkler, S. T. Reddy, Systems Analysis Reveals High Genetic and Antigen-Driven Predetermination of Antibody Repertoires throughout B Cell Development. *Cell Rep.* 19, 1467–1478 (2017).

20. V. Greiff, C. R. Weber, J. Palme, U. Bodenhofer, E. Miho, U. Menzel, S. T. Reddy, Learning the High-Dimensional Immunogenomic Features That Predict Public and Private Antibody Repertoires. *J. Immunol.* 199, 2985–2997 (2017).

21. A. Christophersen, M. Ráki, E. Bergseng, K. E. Lundin, J. Jahnsen, L. M. Sollid, S.-W. Qiao, Tetramer-visualized gluten-specific CD4+ T cells in blood as a potential diagnostic marker for coeliac disease without oral gluten challenge. *United European Gastroenterol J.* 2, 268–278 (2014).

22. R. K. Abbott, J. H. Lee, S. Menis, P. Skog, M. Rossi, T. Ota, D. W. Kulp, D. Bhullar, O.

Kalyuzhniy, C. Havenar-Daughton, W. R. Schief, D. Nemazee, S. Crotty, Precursor Frequency and Affinity Determine B Cell Competitive Fitness in Germinal Centers, Tested with Germline-Targeting HIV Vaccine Immunogens. *Immunity*. 48, 133–146.e6 (2018).

23. A. J. Brown, I. Snapkov, R. Akbar, M. Pavlović, E. Miho, G. K. Sandve, V. Greiff, Augmenting adaptive immunity: progress and challenges in the quantitative engineering and analysis of adaptive immune receptor repertoires. *Mol. Syst. Des. Eng.* 4, 701–736 (2019).

24. C. R. Weber, T. Rubio, L. Wang, W. Zhang, P. A. Robert, R. Akbar, I. Snapkov, J. Wu, M. L. Kuijjer, S. Tarazona, A. Conesa, G. K. Sandve, X. Liu, S. T. Reddy, V. Greiff, Reference-based comparison of adaptive immune receptor repertoires. *bioRxiv* (2022), p. 2022.01.23.476436.

25. G. Eraslan, Ž. Avsec, J. Gagneur, F. J. Theis, Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* 20, 389–403 (2019).

26. J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, S. Zhao, Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* 18, 463–477 (2019).

27. A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare. *Nat. Med.* 25, 24–29 (2019).

28. M. Wainberg, D. Merico, A. Delong, B. Frey, Deep learning in biomedicine. *Nat. Biotechnol.* 36, 829–838 (2018).

29. C. Xu, S. A. Jackson, Machine learning and complex biological data. *Genome Biol.* 20, 76 (2019).

30. T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, W. Xie, G. L. Rosen, B. J. Lengerich, J. Israeli, J. Lanchantin, S. Woloszynek, A. E. Carpenter, A. Shrikumar, J. Xu, E. M. Cofer, C. A. Lavender, S. C. Turaga, A. M. Alexandari, Z. Lu, D. J. Harris, D. DeCaprio, Y. Qi, A. Kundaje, Y. Peng, L. K. Wiley, M. H. S. Segler, S. M. Boca, S. J. Swamidass, A. Huang, A. Gitter, C. S. Greene, Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface*. 15 (2018), doi:10.1098/rsif.2017.0387.

31. M. W. Libbrecht, W. S. Noble, Machine learning applications in genetics and genomics. *Nat. Rev. Genet.* 16, 321–332 (2015).

32. J. G. Greener, S. M. Kandathil, L. Moffat, D. T. Jones, A guide to machine learning for biologists. *Nat. Rev. Mol. Cell Biol.* 23, 40–55 (2022).

33. M. Pavlović, L. Scheffer, K. Motwani, C. Kanduri, R. Kompova, N. Vazov, K. Waagan, F. L. M. Bernal, A. A. Costa, B. Corrie, R. Akbar, G. S. Al Hajj, G. Balaban, T. M. Brusko, M. Chernigovskaya, S. Christley, L. G. Cowell, R. Frank, I. Grytten, S. Gundersen, I. H. Haff, E. Hovig, P.-H. Hsieh, G. Klambauer, M. L. Kuijjer, C. Lund-Andersen, A. Martini, T. Minotto, J. Pensar, K. Rand, E. Riccardi, P. A. Robert, A. Rocha, A. Slabodkin, I. Snapkov, L. M. Sollid, D. Titov, C. R. Weber, M. Widrich, G. Yaari, V. Greiff, G. K. Sandve, The immuneML

ecosystem for machine learning analysis of adaptive immune receptor repertoires. *Nature Machine Intelligence*. 3, 936–944 (2021).

34. T. W. Kuijpers, M. T. Vossen, M.-R. Gent, J.-C. Davin, M. T. Roos, P. M. Wertheim-van Dillen, J. F. Weel, P. A. Baars, R. A. van Lier, Frequencies of Circulating Cytolytic, CD45RA CD27−, CD8 T Lymphocytes Depend on Infection with CMV. *The Journal of Immunology*. 170 (2003), pp. 4342–4348.

35. C. Kanduri, M. Pavlović, L. Scheffer, K. Motwani, M. Chernigovskaya, V. Greiff, G. K. Sandve, Profiling the baseline performance and limits of machine learning models for adaptive immune receptor repertoire classification. *bioRxiv* (2021), p. 2021.05.23.445346.

36. Pavlovic, Al Hajj, Greiff, Pensar, Using causal modeling to analyze generalization of biomarkers in high-dimensional domains: a case study of adaptive immune repertoires. *ICML 2022: Workshop* (available at https://openreview.net/pdf?id=pH7ZF604wDk).

37. A. Slabodkin, L. M. Sollid, G. K. Sandve, P. A. Robert, V. Greiff, Weakly supervised identification and generation of adaptive immune receptor sequences associated with immune disease status. *bioRxiv* (2023), p. 2023.09.24.558823.

38. M. Widrich, B. Schäfl, M. Pavlović, H. Ramsauer, L. Gruber, M. Holzleitner, J. Brandstetter, G. K. Sandve, V. Greiff, S. Hochreiter, Others, Modern hopfield networks and attention for immune repertoire classification. *Adv. Neural Inf. Process. Syst.* 33, 18832–18845 (2020).

39. M. F. Pradier, N. Prasad, P. Chapfuwa, S. Ghalebikesabi, M. Ilse, S. Woodhouse, R. Elyanow, J. Zazo, J. Gonzalez, J. Greissl, E. Meeds, AIRIVA: A Deep Generative Model of Adaptive Immune Repertoires. *arXiv [q-bio.QM]* (2023), (available at http://arxiv.org/abs/2304.13737).

40. P. A. Robert, R. Akbar, R. Frank, M. Pavlović, M. Widrich, I. Snapkov, M. Chernigovskaya, L. Scheffer, A. Slabodkin, B. B. Mehta, M. H. Vu, A. Prósz, K. Abram, A. Olar, E. Miho, D. T. T. Haug, F. Lund-Johansen, S. Hochreiter, I. H. Haff, G. Klambauer, G. K. Sandve, V. Greiff, One billion synthetic 3D-antibody-antigen complexes enable unconstrained machine-learning formalized investigation of antibody specificity prediction, , doi:10.1101/2021.07.06.451258.

41. A. Weber, J. Born, M. Rodriguez Martínez, TITAN: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics*. 37, i237–i244 (2021).

42. P. Moris, J. De Pauw, A. Postovskaya, S. Gielis, N. De Neuter, W. Bittremieux, B. Ogunjimi, K. Laukens, P. Meysman, Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. *Brief. Bioinform.* 22 (2021), doi:10.1093/bib/bbaa318.

43. D. Hudson, R. A. Fernandes, M. Basham, G. Ogg, H. Koohy, Can we predict T cell specificity with digital biology and machine learning? *Nat. Rev. Immunol.*, 1–11 (2023).

44. C. Dens, K. Laukens, W. Bittremieux, P. Meysman, The pitfalls of negative data bias for the T-cell epitope specificity challenge. *bioRxiv* (2023), p. 2023.04.06.535863.

45. G. K. Sandve, V. Greiff, Access to ground truth at unconstrained size makes simulated data as indispensable as experimental data for bioinformatics methods development and benchmarking. *Bioinformatics* (2022), doi:10.1093/bioinformatics/btac612.

46. M. Pavlović, G. S. Al Hajj, J. Pensar, M. Wood, L. M. Sollid, V. Greiff, G. K. Sandve, Improving generalization of machine learning-identified biomarkers with causal modeling: an investigation into immune receptor diagnostics. *arXiv [q-bio.QM]* (2022), (available at http://arxiv.org/abs/2204.09291).

47. V. Chen, M. Yang, W. Cui, J. S. Kim, A. Talwalkar, J. Ma, Best Practices for Interpretable Machine Learning in Computational Biology. *bioRxiv* (2022), p. 2022.10.28.513978.

48. E. I. Prakash, A. Shrikumar, A. Kundaje, "Towards More Realistic Simulated Datasets for Benchmarking Deep Learning Models in Regulatory Genomics" in *Proceedings of the 16th Machine Learning in Computational Biology meeting*, D. A. Knowles, S. Mostafavi, S.-I. Lee, Eds. (PMLR, 22--23 Nov 2022), vol. 165 of *Proceedings of Machine Learning Research*, pp. 58–77.

49. J. D. Romano, T. T. Le, W. La Cava, J. T. Gregg, D. J. Goldberg, P. Chakraborty, N. L. Ray, D. Himmelstein, W. Fu, J. H. Moore, PMLB v1.0: an open-source dataset collection for benchmarking machine learning methods. *Bioinformatics*. 38, 878–880 (2022).

50. J. Thiyagalingam, M. Shankar, G. Fox, T. Hey, Scientific machine learning benchmarks. *Nature Reviews Physics*. 4, 413–420 (2022).

51. Q. Marcou, T. Mora, A. M. Walczak, High-throughput immune repertoire analysis with IGoR. *Nat. Commun.* 9, 561 (2018).

52. C. R. Weber, R. Akbar, A. Yermanos, M. Pavlović, I. Snapkov, G. K. Sandve, S. T. Reddy, V. Greiff, immuneSIM: tunable multi-feature simulation of B- and T-cell receptor repertoires for immunoinformatics benchmarking. *Bioinformatics* (2020), doi:10.1093/bioinformatics/btaa158.

53. M. G. Woodcock, D. S. Bortone, B. G. Vincent, STIG: Generation and simulated sequencing of synthetic T cell receptor repertoires. *bioRxiv* (2020), p. 2020.02.28.969469.

54. Y. Safonova, A. Lapidus, J. Lill, IgSimulator: a versatile immunosequencing simulator. *Bioinformatics*, btv326 (2015).

55. A. Yermanos, V. Greiff, N. J. Krautler, U. Menzel, A. Dounas, E. Miho, A. Oxenius, T. Stadler, S. T. Reddy, Comparison of methods for phylogenetic B-cell lineage inference using time-resolved antibody repertoire simulations (AbSim). *Bioinformatics* (2017), doi:10.1093/bioinformatics/btx533.

56. J. Han, S. Masserey, D. Shlesinger, R. Kuhn, C. Papadopoulou, A. Agrafiotis, V. Kreiner, R. Dizerens, K.-L. Hong, C. Weber, V. Greiff, A. Oxenius, S. T. Reddy, A. Yermanos, Echidna: Integrated simulations of single-cell immune receptor repertoires and transcriptomes. *Bioinformatics Advances*, vbac062 (2022).

57. C. Sutherland, G. J. M. Cowan, AIRRSHIP: simulating human B cell receptor repertoire

sequences, , doi:10.1101/2022.12.20.521228.

58. X. Yang, Y. Zhu, H. Zeng, S. Chen, J. Guan, Q. Wang, C. Lan, D. Sun, X. Yu, Z. Zhang, Knowledge-based antibody repertoire simulation, a novel allele detection tool evaluation and application. *bioRxiv* (2021), p. 2021.07.01.450681.

59. K. Davidsen, B. J. Olson, W. S. DeWitt, J. Feng, E. Harkins, P. Bradley, F. A. Matsen, Deep generative models for T cell receptor protein sequences. *eLife*. 8 (2019), , doi:10.7554/elife.46935.

60. G. Isacchini, A. M. Walczak, T. Mora, A. Nourmohammad, Deep generative selection models of T and B cell receptor repertoires with soNNia. *Proc. Natl. Acad. Sci. U. S. A.* 118 (2021), doi:10.1073/pnas.2023141118.

61. A. Slabodkin, M. Chernigovskaya, I. Mikocziova, R. Akbar, L. Scheffer, M. Pavlović, H. Bashour, I. Snapkov, B. B. Mehta, C. R. Weber, J. Gutierrez-Marcos, L. M. Sollid, I. H. Haff, G. K. Sandve, P. A. Robert, V. Greiff, Individualized VDJ recombination predisposes the available Ig sequence space. *Genome Res.* (2021), doi:10.1101/gr.275373.121.

62. C. Kanduri, L. Scheffer, M. Pavlović, K. D. Rand, M. Chernigovskaya, O. Pirvandy, G. Yaari, V. Greiff, G. K. Sandve, simAIRR: simulation of adaptive immune repertoires with realistic receptor sequence sharing for benchmarking of immune state prediction methods. *Gigascience*, doi:10.1093/gigascience/giad074.

63. C. Kanduri, M. Pavlović, L. Scheffer, K. Motwani, M. Chernigovskaya, V. Greiff, G. K. Sandve, Profiling the baseline performance and limits of machine learning models for adaptive immune receptor repertoire classification. *Gigascience*. 11 (2022), doi:10.1093/gigascience/giac046.

64. O. Shemesh, P. Polak, K. E. A. Lundin, L. M. Sollid, G. Yaari, Machine Learning Analysis of Naïve B-Cell Receptor Repertoires Stratifies Celiac Disease Patients and Controls. *Front. Immunol.* 12, 627813 (2021).

65. N. Thomas, K. Best, M. Cinelli, S. Reich-Zeliger, H. Gal, E. Shifrut, A. Madi, N. Friedman, J. Shawe-Taylor, B. Chain, Tracking global changes induced in the CD4 T-cell receptor repertoire by immunization with a complex antigen using short stretches of CDR3 protein sequence. *Bioinformatics*. 30, 3181–3188 (2014).

66. P. A. Robert, R. Akbar, R. Frank, M. Pavlović, M. Widrich, I. Snapkov, A. Slabodkin, M. Chernigovskaya, L. Scheffer, E. Smorodina, P. Rawat, B. B. Mehta, M. H. Vu, I. F. Mathisen, A. Prósz, K. Abram, A. Olar, E. Miho, D. T. T. Haug, F. Lund-Johansen, S. Hochreiter, I. H. Haff, G. Klambauer, G. K. Sandve, V. Greiff, Unconstrained generation of synthetic antibody–antigen structures to guide machine learning methodology for antibody specificity prediction. *Nature Computational Science*. 2, 845–865 (2022).

67. D. B. Jaffe, P. Shahi, B. A. Adams, A. M. Chrisman, P. M. Finnegan, N. Raman, A. E. Royall, F. Tsai, T. Vollbrecht, D. S. Reyes, N. L. Hepler, W. J. McDonnell, Functional antibodies exhibit light chain coherence. *Nature*. 611, 352–357 (2022).

68. Z. Sethna, Y. Elhanati, C. G. Callan, A. M. Walczak, T. Mora, OLGA: fast computation of

generation probabilities of B- and T-cell receptor amino acid sequences and motifs. *Bioinformatics*. 35 (2019), pp. 2974–2981.

69. B. D. Corrie, N. Marthandan, B. Zimonja, J. Jaglale, Y. Zhou, E. Barr, N. Knoetze, F. M. W. Breden, S. Christley, J. K. Scott, L. G. Cowell, F. Breden, iReceptor: A platform for querying and analyzing antibody/B-cell and T-cell receptor repertoire data across federated repositories. *Immunol. Rev.* 284, 24–41 (2018).

70. A. Kovaltsuk, J. Leem, S. Kelm, J. Snowden, C. M. Deane, K. Krawczyk, Observed Antibody Space: A Resource for Data Mining Next-Generation Sequencing of Antibody Repertoires. *J. Immunol.* 201, 2502–2509 (2018).

71. J. P. Gygi, S. H. Kleinstein, L. Guan, Predictive overfitting in immunological applications: Pitfalls and solutions. *Hum. Vaccin. Immunother.* 19, 2251830 (2023).

72. A. Montemurro, L. E. Jessen, M. Nielsen, NetTCR-2.1: Lessons and guidance on how to develop models for TCR specificity predictions. *Front. Immunol.* 13, 1055151 (2022).

73. L. Deng, C. Ly, S. Abdollahi, Y. Zhao, I. Prinz, S. Bonn, Performance comparison of TCR-pMHC prediction tools reveals a strong data dependency. *Front. Immunol.* 14 (2023), doi:10.3389/fimmu.2023.1128326.

74. P. Meysman, J. Barton, B. Bravi, L. Cohen-Lavi, V. Karnaukhov, E. Lilleskov, A. Montemurro, M. Nielsen, T. Mora, P. Pereira, A. Postovskaya, M. R. Martínez, J. Fernandez-de-Cossio-Diaz, A. Vujkovic, A. M. Walczak, A. Weber, R. Yin, A. Eugster, V. Sharma, Benchmarking solutions to the T-cell receptor epitope prediction problem: IMMREP22 workshop report. *ImmunoInformatics*. 9, 100024 (2023).

75. I. Walsh, G. Pollastri, S. C. E. Tosatto, Correct machine learning on protein sequences: a peer-reviewing perspective. *Brief. Bioinform.* 17, 831–840 (2016).

76. S. Petti, S. R. Eddy, Constructing benchmark test sets for biological sequence analysis using independent set algorithms. *PLoS Comput. Biol.* 18, e1009492 (2022).

77. R. L. Warren, J. D. Freeman, T. Zeng, G. Choe, S. Munro, R. Moore, J. R. Webb, R. A. Holt, Exhaustive T-cell repertoire sequencing of human peripheral blood samples reveals signatures of antigen selection and a directly measured repertoire size of at least 1 million clonotypes. *Genome Res.* 21, 790–797 (2011).

78. B. Briney, A. Inderbitzin, C. Joyce, D. R. Burton, Commonality despite exceptional diversity in the baseline human antibody repertoire. *Nature*. 566 (2019), pp. 393–397.

79. Y. Katayama, T. J. Kobayashi, Comparative Study of Repertoire Classification Methods Reveals Data Efficiency of k -mer Feature Extraction. *Front. Immunol.* 13, 797640 (2022).

80. M. Cinelli, Y. Sun, K. Best, J. M. Heather, S. Reich-Zeliger, E. Shifrut, N. Friedman, J. Shawe-Taylor, B. Chain, Feature selection using a one dimensional naïve Bayes' classifier increases the accuracy of support vector machine classification of CDR3 repertoires. *Bioinformatics*. 33, 951–955 (2017).

81. Y. Sun, K. Best, M. Cinelli, J. M. Heather, S. Reich-Zeliger, E. Shifrut, N. Friedman, J. Shawe-Taylor, B. Chain, Specificity, Privacy, and Degeneracy in the CD4 T Cell Receptor Repertoire Following Immunization. *Front. Immunol.* 8, 430 (2017).

82. A. A. Abu-Shmais, M. J. Vukovich, P. T. Wasdin, Y. P. Suresh, S. A. Rush, R. A. Gillespie, R. S. Sankhala, M. Choe, M. Gordon Joyce, M. Kanekiyo, J. S. McLellan, I. S. Georgiev, Convergent Sequence Features of Antiviral B Cells. *bioRxiv* (2023), p. 2023.09.06.556442.

83. W. S. DeWitt 3rd, A. Smith, G. Schoch, J. A. Hansen, F. A. Matsen 4th, P. Bradley, Human T cell receptor occurrence patterns encode immune history, genetic background, and receptor specificity. *Elife*. 7 (2018), doi:10.7554/eLife.38358.

84. M. Pennell, O. L. Rodriguez, C. T. Watson, V. Greiff, The evolutionary and functional significance of germline immunoglobulin gene variation. *Trends Immunol.* 44, 7–21 (2023).

85. O. L. Rodriguez, Y. Safonova, C. A. Silver, K. Shields, W. S. Gibson, J. T. Kos, D. Tieri, H. Ke, K. J. L. Jackson, S. D. Boyd, M. L. Smith, W. A. Marasco, C. T. Watson, Genetic variation in the immunoglobulin heavy chain locus shapes the human antibody repertoire. *Nat. Commun.* 14, 4419 (2023).

86. A. Omer, A. Peres, O. L. Rodriguez, C. T. Watson, W. Lees, P. Polak, A. M. Collins, G. Yaari, T cell receptor beta germline variability is revealed by inference from repertoire data. *Genome Med.* 14, 2 (2022).

87. A. Peres, W. D. Lees, O. L. Rodriguez, N. Y. Lee, P. Polak, R. Hope, M. Kedmi, A. M. Collins, M. Ohlin, S. H. Kleinstein, C. T. Watson, G. Yaari, IGHV allele similarity clustering improves genotype inference from adaptive immune receptor repertoire sequencing data. *Nucleic Acids Res.* (2023), doi:10.1093/nar/gkad603.

88. W. Huang, L. Li, J. R. Myers, G. T. Marth, ART: a next-generation sequencing read simulator. *Bioinformatics*. 28, 593–594 (2012).

89. H. Gourlé, O. Karlsson-Lindsjö, J. Hayer, E. Bongcam-Rudloff, Simulating Illumina metagenomic data with InSilicoSeq. *Bioinformatics*. 35, 521–522 (2019).

90. A. O. Smirnova, A. M. Miroshnichenkova, Y. V. Olshanskaya, M. A. Maschan, Y. B. Lebedev, D. M. Chudakov, I. Z. Mamedov, A. Komkov, The use of non-functional clonotypes as a natural calibrator for quantitative bias correction in adaptive immune receptor repertoire profiling. *Elife*. 12 (2023), doi:10.7554/eLife.69157.

91. G. Yaari, J. A. Vander Heiden, M. Uduman, D. Gadala-Maria, N. Gupta, J. N. H. Stern, K. C. O'Connor, D. A. Hafler, U. Laserson, F. Vigneault, S. H. Kleinstein, Models of somatic hypermutation targeting and substitution based on synonymous mutations from high-throughput immunoglobulin sequencing data. *Front. Immunol.* 4, 358 (2013).

92. M. Safra, L. Werner, A. Peres, P. Polak, N. Salamon, M. Schvimer, B. Weiss, I. Barshack, D. S. Shouval, G. Yaari, A somatic hypermutation-based machine learning model stratifies individuals with Crohn's disease and controls. *Genome Res.* 33, 71–79 (2023).

93. Z. Sheng, C. A. Schramm, R. Kong, NISC Comparative Sequencing Program, J. C. Mullikin,

J. R. Mascola, P. D. Kwong, L. Shapiro, Gene-Specific Substitution Profiles Describe the Types and Frequencies of Amino Acid Changes during Antibody Somatic Hypermutation. *Front. Immunol.* 8, 537 (2017).

94. D. K. Ralph, F. A. Matsen 4th, Inference of B cell clonal families using heavy/light chain pairing information. *PLoS Comput. Biol.* 18, e1010723 (2022).

95. K. B. Hoehn, O. G. Pybus, S. H. Kleinstein, Phylogenetic analysis of migration, differentiation, and class switching in B cells. *PLoS Comput. Biol.* 18, e1009885 (2022).

96. K. B. Hoehn, J. A. Vander Heiden, J. Q. Zhou, G. Lunter, O. G. Pybus, S. H. Kleinstein, Repertoire-wide phylogenetic models of B cell molecular evolution reveal evolutionary signatures of aging and vaccination. *Proc. Natl. Acad. Sci. U. S. A.* 116, 22664–22672 (2019).

97. C. Zhang, A. V. Bzikadze, Y. Safonova, S. Mirarab, A scalable model for simulating multi-round antibody evolution and benchmarking of clonal tree reconstruction methods. *Front. Immunol.* 13, 1014439 (2022).

98. T. Konstantinovsky, G. Yaari, A Novel Approach to T-Cell Receptor Beta Chain (TCRB) Repertoire Encoding Using Lossless String Compression. *bioRxiv* (2023), p. 2023.01.30.526195.

99. P. Moris, J. De Pauw, A. Postovskaya, S. Gielis, N. De Neuter, W. Bittremieux, B. Ogunjimi, K. Laukens, P. Meysman, Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. *Brief. Bioinform.* 22 (2021), doi:10.1093/bib/bbaa318.

100. M. Gidoni, O. Snir, A. Peres, P. Polak, I. Lindeman, I. Mikocziova, V. K. Sarna, K. E. A. Lundin, C. Clouser, F. Vigneault, A. M. Collins, L. M. Sollid, G. Yaari, Mosaic deletion patterns of the human antibody heavy chain gene locus shown by Bayesian haplotyping. *Nat. Commun.* 10, 628 (2019).

101. F. Rubelt, C. E. Busse, S. A. C. Bukhari, J.-P. Bürckert, E. Mariotti-Ferrandiz, L. G. Cowell, C. T. Watson, N. Marthandan, W. J. Faison, U. Hershberg, U. Laserson, B. D. Corrie, M. M. Davis, B. Peters, M.-P. Lefranc, J. K. Scott, F. Breden, AIRR Community, E. T. Luning Prak, S. H. Kleinstein, Adaptive Immune Receptor Repertoire Community recommendations for sharing immune-repertoire sequencing data. *Nat. Immunol.* 18, 1274–1278 (2017).

102. D. A. Bolotin, S. Poslavsky, I. Mitrophanov, M. Shugay, I. Z. Mamedov, E. V. Putintseva, D. M. Chudakov, MiXCR: software for comprehensive adaptive immunity profiling. *Nat. Methods*. 12, 380–381 (2015).

103. G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, M. T. Gregory, J. Shuga, L. Montesclaros, J. G. Underwood, D. A. Masquelier, S. Y. Nishimura, M. Schnall-Levin, P. W. Wyatt, C. M. Hindson, R. Bharadwaj, A. Wong, K. D. Ness, L. W. Beppu, H. J. Deeg, C. McFarland, K. R. Loeb, W. J. Valente, N. G. Ericson, E. A. Stevens, J. P. Radich, T. S. Mikkelsen, B. J. Hindson, J. H. Bielas, Massively parallel digital transcriptional profiling of single cells. *Nat.*

Commun. 8, 14049 (2017).

104. S. Nolan, M. Vignali, M. Klinger, J. N. Dines, I. M. Kaplan, E. Svejnoha, T. Craft, K. Boland, M. Pesesky, R. M. Gittelman, T. M. Snyder, C. J. Gooley, S. Semprini, C. Cerchione, M. Mazza, O. M. Delmonte, K. Dobbs, G. Carreño-Tarragona, S. Barrio, V. Sambri, G. Martinelli, J. D. Goldman, J. R. Heath, L. D. Notarangelo, J. M. Carlson, J. Martinez-Lopez, H. S. Robins, A large-scale database of T-cell receptor beta (TCRβ) sequences and binding associations from natural and synthetic exposure to SARS-CoV-2. *Res Sq* (2020), doi:10.21203/rs.3.rs-51964/v1.

105. D. V. Bagaev, R. M. A. Vroomans, J. Samir, U. Stervbo, C. Rius, G. Dolton, A. Greenshields-Watson, M. Attaf, E. S. Egorov, I. V. Zvyagin, N. Babel, D. K. Cole, A. J. Godkin, A. K. Sewell, C. Kesmir, D. M. Chudakov, F. Luciani, M. Shugay, VDJdb in 2019: database extension, new analysis infrastructure and a T-cell receptor motif compendium. *Nucleic Acids Res.* 48, D1057–D1062 (2020).

106. K. Rand, I. Grytten, M. Pavlovic, C. Kanduri, G. K. Sandve, BioNumPy: Fast and easy analysis of biological data with Python. *bioRxiv* (2022), p. 2022.12.21.521373.

107. M.-P. Lefranc, C. Pommié, M. Ruiz, V. Giudicelli, E. Foulquier, L. Truong, V. Thouvenin-Contet, G. Lefranc, IMGT unique numbering for immunoglobulin and T cell receptor variable domains and Ig superfamily V-like domains. *Dev. Comp. Immunol.* 27, 55–77 (2003).

108. R. Akbar, P. A. Robert, C. R. Weber, M. Widrich, R. Frank, M. Pavlović, L. Scheffer, M. Chernigovskaya, I. Snapkov, A. Slabodkin, B. B. Mehta, E. Miho, F. Lund-Johansen, J. T. Andersen, S. Hochreiter, I. Hobæk Haff, G. Klambauer, G. K. Sandve, V. Greiff, In silico proof of principle of machine learning-based antibody design at unconstrained scale. *MAbs*. 14, 2031482 (2022).

109. J. Q. Zhou, S. H. Kleinstein, Cutting Edge: Ig H Chains Are Sufficient to Determine Most B Cell Clonal Relationships. *J. Immunol.* 203, 1687–1692 (2019).

110. G. Georgiou, G. C. Ippolito, J. Beausang, C. E. Busse, H. Wardemann, S. R. Quake, The promise and challenge of high-throughput sequencing of the antibody repertoire. *Nat. Biotechnol.* 32, 158–168 (2014).

111. D. S. Shcherbinin, V. A. Belousov, M. Shugay, Comprehensive analysis of structural and sequencing data reveals almost unconstrained chain pairing in TCRαβ complex. *PLoS Comput. Biol.* 16, e1007714 (2020).

112. J. Rossjohn, S. Gras, J. J. Miles, S. J. Turner, D. I. Godfrey, J. McCluskey, T cell antigen receptor recognition of antigen-presenting molecules. *Annu. Rev. Immunol.* 33, 169–200 (2015).

113. M. J. T. Stubbington, O. Rozenblatt-Rosen, A. Regev, S. A. Teichmann, Single-cell transcriptomics to explore the immune system in health and disease. *Science*. 358, 58–63 (2017).

114. S. Friedensohn, T. A. Khan, S. T. Reddy, Advanced Methodologies in High-Throughput

Sequencing of Immune Repertoires. *Trends Biotechnol.* 35, 203–214 (2017).

115. T. Dupic, Q. Marcou, A. M. Walczak, T. Mora, Genesis of the αβ T-cell receptor. *PLoS Comput. Biol.* 15, e1006874 (2019).

116. N. Jayaram, P. Bhowmick, A. C. R. Martin, Germline VH/VL pairing in antibodies. *Protein Eng. Des. Sel.* 25, 523–529 (2012).

117. B. DeKosky, Paired VH:VL Analysis of Naïve B Cell Repertoires and Comparison to Antigen-Experienced B Cell Repertoires in Healthy Human Donors. *Decoding the Antibody Repertoire* (2017), pp. 41–57.

118. B. J. DeKosky, O. I. Lungu, D. Park, E. L. Johnson, W. Charab, C. Chrysostomou, D. Kuroda, A. D. Ellington, G. C. Ippolito, J. J. Gray, G. Georgiou, Large-scale sequence and structural comparisons of human naive and antigen-experienced antibody repertoires. *Proc. Natl. Acad. Sci. U. S. A.* 113, E2636–45 (2016).

119. H. Tanno, T. M. Gould, J. R. McDaniel, W. Cao, Y. Tanno, R. E. Durrett, D. Park, S. J. Cate, W. H. Hildebrand, C. L. Dekker, L. Tian, C. M. Weyand, G. Georgiou, J. J. Goronzy, Determinants governing T cell receptor α/β-chain pairing in repertoire formation of identical twins. *Proc. Natl. Acad. Sci. U. S. A.* 117, 532–540 (2020).

120. M. V. Pogorelyy, A. A. Minervina, M. Shugay, D. M. Chudakov, Y. B. Lebedev, T. Mora, A. M. Walczak, Detecting T cell receptors involved in immune responses from single repertoire snapshots. *PLoS Biol.* 17, e3000314 (2019).

121. Y. Elhanati, Z. Sethna, C. G. Callan Jr, T. Mora, A. M. Walczak, Predicting the spectrum of TCR repertoire sharing with a data-driven model of recombination. *Immunol. Rev.* 284, 167–179 (2018).

122. P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*. 17, 261–272 (2020).

123. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine Learning in Python. *arXiv [cs.LG]* (2012), pp. 2825–2830.