One billion synthetic 3D-antibody-antigen complexes enable unconstrained machine-learning formalized investigation of antibody specificity prediction

Philippe A. Robert[1,☯,#], Rahmad Akbar[1,☯], Robert Frank[1], Milena Pavlović[2], Michael Widrich[3], Igor Snapkov[1], Maria Chernigovskaya[1], Lonneke Scheffer[2], Andrei Slabodkin[1], Brij Bhushan Mehta[1], Mai Ha Vu[4], Aurél Prósz[5], Krzysztof Abram[6], Alex Olar[7], Enkelejda Miho[8], Dag Trygve Tryslew Haug[9], Fridtjof Lund-Johansen[1], Sepp Hochreiter[3,10], Ingrid Hobæk Haff[11], Günter Klambauer[3], Geir K. Sandve[2], and Victor Greiff[1,#]

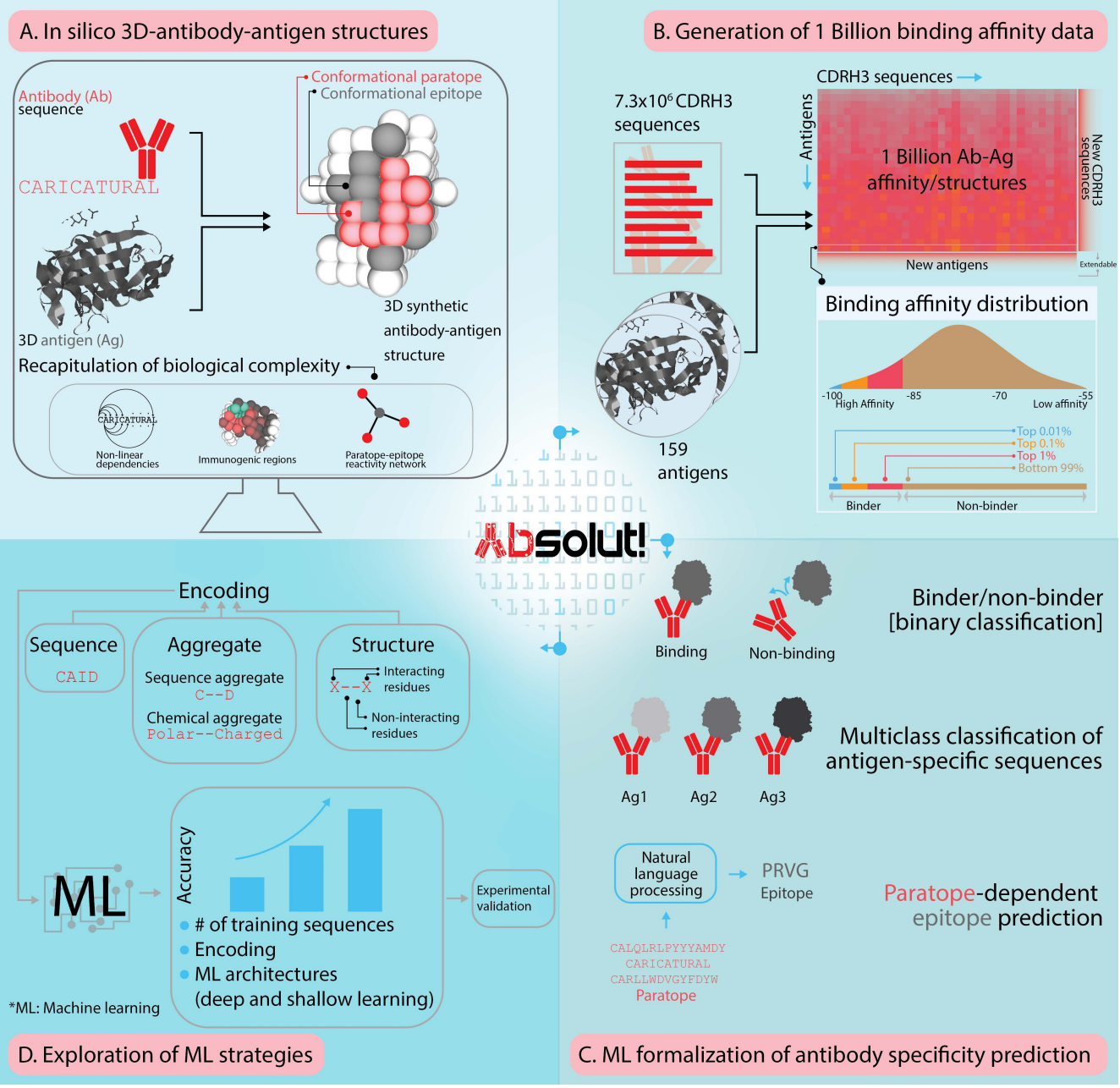
[1]Department of Immunology, University of Oslo, Oslo, Norway
[2]Department of Informatics, University of Oslo, Oslo, Norway
[3]ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning, Johannes Kepler University Linz, Linz, Austria
[4]Department of Linguistics and Scandinavian Studies, University of Oslo, Oslo, Norway
[5]Danish Cancer Society Research Center, Translational Cancer Genomics, Copenhagen, Denmark
[6]The Novo Nordisk Foundation Center for Biosustainability, Autoflow, DTU Biosustain and IT University of Copenhagen, Copenhagen, Denmark
[7]Eötvös Loránd University, Department of Complex Systems in Physics, Budapest, Hungary
[8]Institute of Medical Engineering and Medical Informatics, School of Life Sciences, FHNW University of Applied Sciences and Arts Northwestern Switzerland, Muttenz, Switzerland
[9]Department of Philosophy, Classics, History of Arts and Ideas, University of Oslo, Oslo, Norway
[10]Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria
[11]Department of Mathematics, University of Oslo, Oslo, Norway
[☯]Equal contribution, [#]Correspondence: philippe.robert@ens-lyon.org, victor.greiff@medisin.uio.no

Abstract

Machine learning (ML) is a key technology to enable accurate prediction of antibody-antigen binding, a prerequisite for *in silico* vaccine and antibody design. Two orthogonal problems hinder the current application of ML to antibody-specificity prediction and the benchmarking thereof: (i) The lack of a unified formalized mapping of immunological antibody specificity prediction problems into ML notation and (ii) the unavailability of large-scale training datasets. Here, we developed the *Absolut!* software suite that allows the parameter-based unconstrained generation of synthetic lattice-based 3D-antibody-antigen binding structures with ground-truth access to conformational paratope, epitope, and affinity. We show that Absolut!-generated datasets recapitulate critical biological sequence and structural features that render antibody-antigen binding prediction challenging. To demonstrate the immediate, high-throughput, and large-scale applicability of Absolut!, we have created an online database of 1 billion antibody-antigen structures, the extension of which is only constrained by moderate computational resources. We translated immunological antibody specificity prediction problems into ML tasks and used our database to investigate paratope-epitope binding prediction accuracy as a function of structural information encoding, dataset size, and ML method, which is unfeasible with existing experimental data. Furthermore, we found that *in silico* investigated conditions, predicted to increase antibody specificity prediction accuracy, align with and extend conclusions drawn from experimental antibody-antigen structural data. In summary, the Absolut! framework enables the development and benchmarking of ML strategies for biotherapeutics discovery and design.

Running title: Unconstrained *in silico* exploration of antibody specificity

Keywords: Machine learning, deep learning, antibody, antigen, specificity, epitope, paratope, structure



Graphical abstract | The software framework Absolut! enables (A,B) the generation of virtually arbitrarily large numbers of *in silico* 3D-antibody-antigen structures, (C,D) the formalization of antibody specificity as machine learning (ML) tasks as well as the exploration of ML strategies for paratope-epitope prediction.

Highlights

- Software framework Absolut! to generate an arbitrarily large number of *in silico* 3D-antibody-antigen structures
- Generation of one billion *in silico* antigen-antibody structures reflecting biological layers of complexity that make ML predictions challenging
- Immunological antibody specificity prediction problems formalized as machine learning tasks for which the in silico complexes are immediately usable as benchmark.
- Exploration of machine learning architectures for paratope-epitope interaction prediction accuracy as a function of neural network depth, dataset size, and sequence-structure encoding

Introduction

Antibodies are effector proteins of the adaptive immune system recognizing foreign molecules (antigens) such as microbes, cancer, autoantigens, or vaccine particles. Antibody therapeutics have led to impressive medical breakthroughs in infection, cancer, and autoimmunity (*1*, *2*). The interface on the antibody that establishes the binding contact with the antigen is called paratope. In the majority of cases, the paratope is part of the antibody's complementarity determining region 3 (CDRH3) (*3*–*5*). The antigen part of the 3D-binding interface is called the epitope (*6*, *7*).

The prediction of the recognized 3D epitopes (conformational epitopes) by an antibody is crucial for long-standing problems in computational antibody design (*8*–*13*) and vaccine engineering (*14*). 3D-antibody-antigen complexes resolved at the atomic level, for instance by crystallization, represent the gold standard for describing antibody binding but are time and cost-intensive to generate. Currently, there exist only $\approx 10^3$ non-redundant antibody-antigen structures (*15*, *16*), which is orders of magnitudes smaller than the diversity of antibody sequences ($>10^{13}$, (*17*)). Furthermore, affinity values remain unavailable for the majority of both experimentally determined antibody-antigen structures and 1D antigen-specific antibody sequence datasets (e.g., library-based screening approaches, (*18*)). The lack of data combined with the extreme complexity of antibody-antigen binding (*19*) and protein-protein CDR3 docking (*20*) has rendered antibody-antigen binding prediction an unresolved problem.

Machine learning (ML) represents an increasingly used approach for antibody-antigen binding prediction (*3*, *10*, *18*, *21*–*32*) given its capacity to infer the hidden nonlinear rules underlying high-complexity protein-protein interaction (*30*, *33*–*38*). Recent reports suggest that binding prediction based on antibody sequence or structure may be feasible as long as sufficiently large antigen(epitope)-specific antibody datasets become available (*15*, *16*, *39*–*43*). Currently, ML applications in antibody-antigen binding prediction are heterogeneous and developed on very small and incomplete-knowledge datasets (joint information on paratope, epitope, affinity is unavailable, size of datasets usually <10 000 antibodies). Indeed, ML methods encompass, among others, sequence-based paratope prediction (*3*, *5*, *22*, *44*–*46*) paratope-epitope-linked prediction (*3*, *30*, *45*) and vary in the in- or exclusion of structural information in their encodings. Therefore, there exist many possible ML tasks that map to the immunological problem of antibody-antigen binding prediction. Thus, in addition to the lack of large-scale antibody-antigen binding data, the lack of a formalized mapping of antibody specificity prediction problems from immunological into ML notation hinders the development and benchmarking of ML strategies for antibody biotherapeutics discovery and design.

Here, we provide a 3D-antibody-antigen binding simulation framework to address the antibody-antigen binding data availability problem for ML method development and formalization. The generation of synthetic antibody-antigen structures is based on the energetically optimal binding structure in a 3D lattice, recapitulates a range of physiological properties of antibody-antigen binding, and allows for the exploration of various types of negative control samples that are largely infeasible to create experimentally. We generated the *in silico* binding of 6.9 million murine CDRH3 sequences to 159 antigens (1B antibody-antigen binding pairs). For each CDRH3 sequence and antigen pair, we examined on average 6.8 million binding poses by exhaustive docking to elucidate the energetically optimal binding structure. Our framework facilitates the formalization of a map that connects immunological problems of antibody-antigen binding to ML notation. For three use cases on this map, we investigated the extent to which 1D-sequence and 3D-structural information is required for achieving high prediction accuracy of antibody-antigen binding: (i,ii) binary and multi-class classification of binding, and (iii) paratope-epitope prediction (*8*, *9*). We base our investigation on the premise that a successful ML antibody binding strategy for experimental

datasets should also perform well on synthetic datasets if operating on the same encoding, and therefore can be benchmarked using such synthetic datasets. We found that *in silico* investigated conditions predicted to increase antibody specificity prediction accuracy align with and extend conclusions drawn from experimental antibody-antigen structural data. Although the definitive proof of our premise can only be determined upon the availability of larger experimental affinity-annotated 3D-antibody-antigen datasets, we argue that even a simplified model of antibody-antigen interactions in 3D space entails non-linearities and positional dependencies reminiscent of natural complexity, and is thus relevant in benchmarking the most suitable combinations of encodings and ML strategies.

Results

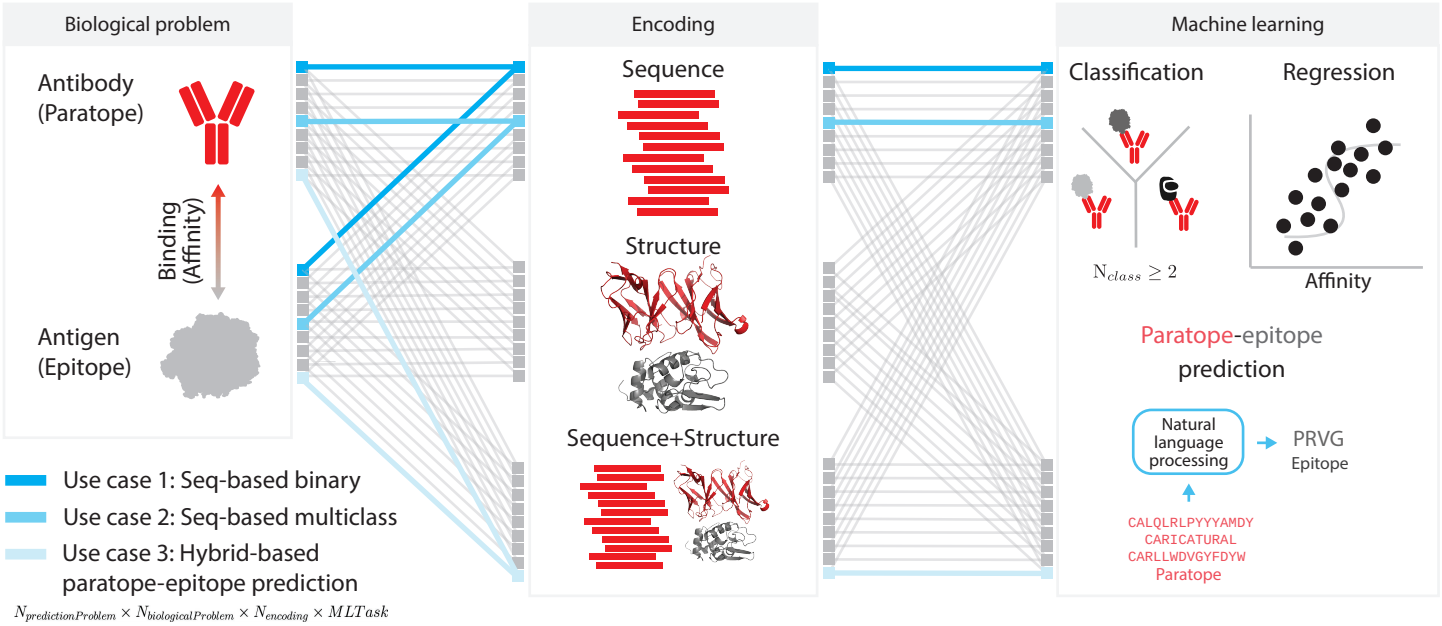
Formalizing and benchmarking antibody specificity prediction problems as ML tasks requires simulated 3D-antibody-antigen data

The biological problem of predicting antibody specificity refers to identifying which antibody sequence(s) bind to which antigen(s), and vice versa (Figure 1A), and remains an unresolved challenge (*10*). The biological formulation of the antibody-antigen prediction problem may involve either complete antibody or antigen sequences or structures, or parts thereof (paratope, epitope). Thus, there is a large variety of possible problem formulations and assumptions (Figure S1), and there is a lack of a formal framework that enables a unified formulation of antibody-antigen binding tasks.

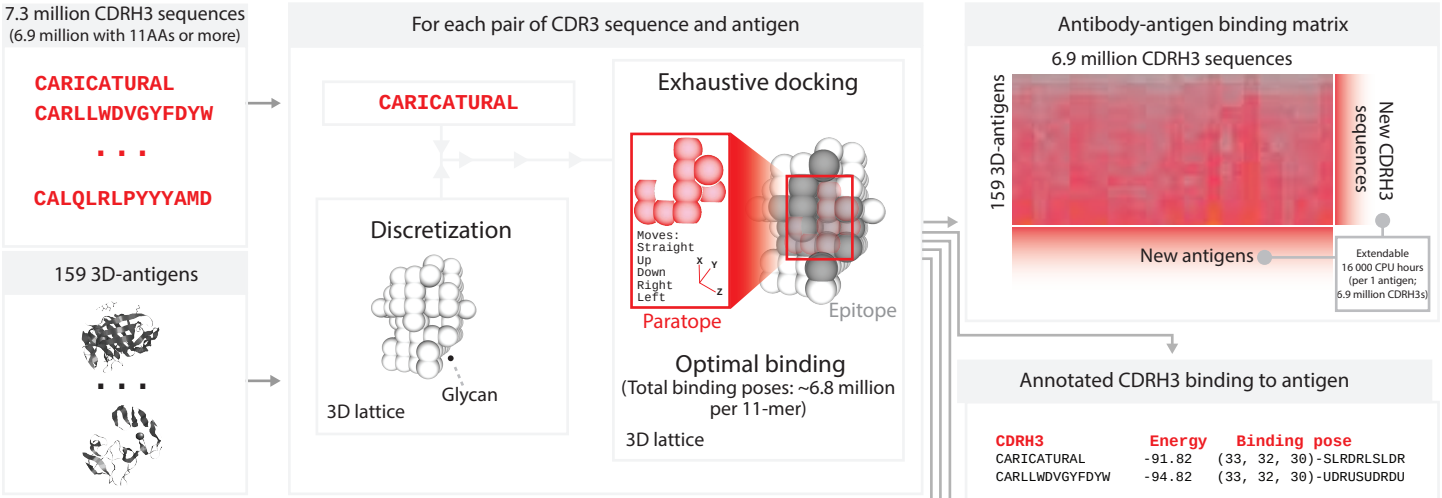
Simulated data is a crucial tool to precisely and meaningfully define different antibody-antigen binding problems, which requires explicit datasets with all necessary levels of annotation that are not available in experimental data so far. Commonly used data encodings can be divided into sequence-based and structure-based ones, while hybrid formalizations leverage both sequence- and structure-based encodings. Therefore, to define and compare ML approaches on the same basis, it is critical that simulated data (i) recapitulates 3D features of experimental antibody-antigen binding (especially for defining paratopes and epitopes); (ii) enables the generation of large-scale datasets, and (iii) allows the integration of sequence and structural information into hybrid encodings.

The majority of ML formalized tasks fall into three main categories: (i) Classification: prediction of whether an antibody sequence binds to an antigen, which may be divided into binary and multi-class/multi-label classification problems. A binary classification (prediction) is suited to predict whether an antibody is a binder or not (two classes, Figure 1A). In multi-class classification, an antibody may be a mutually exclusive element of one of at least three classes describing the binding to one antigen (for example low-affinity, medium-affinity, high-affinity classes); or the binding to different epitopes on an antigen; or even the binding to different antigens (Figure 1A). In multi-label classification (prediction), an antibody may belong to more than one class (e.g., a cross-reactive antibody that binds to more than 1 antigen at once). (ii) Regression: Prediction of the affinity of an antibody sequence to a target antigen, or the magnitude of developability parameters, which can be seen as a regression problem. (iii) Paratope-epitope prediction: Prediction of which residues of an antibody-antigen complex are involved in their binding interface (paratope or epitope prediction, Figure 1A). These problems involve the reconstruction of complex data structures describing features of the binding interface, beyond the scope of labels (classification) or continuous values (regression), and can be addressed using, e.g., Natural Language Processing (NLP) architectures (treating features as text).

A Formalization of antibody-antigen binding prediction



B Antibody-antigen binding data generation using Absolut!



C Recapitulation of biological complexity

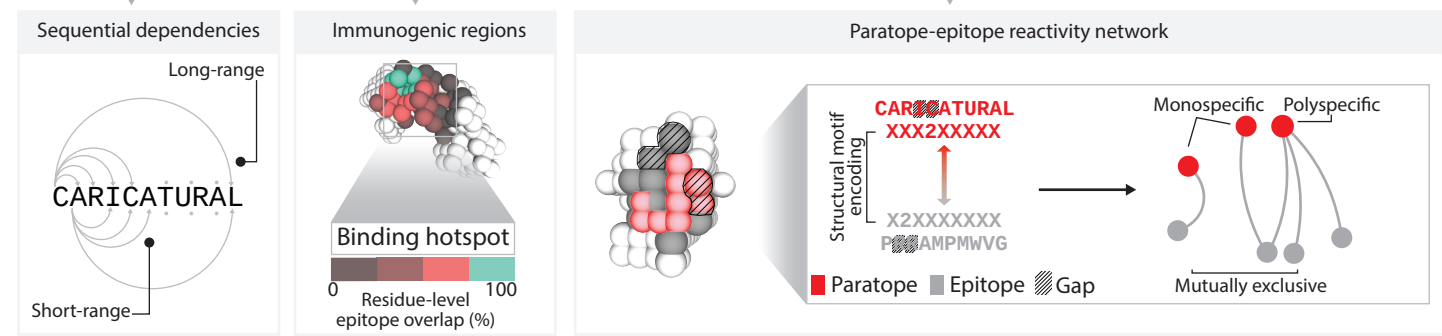


Figure 1 | Formalization of antibody-antigen binding and pipeline for the high-throughput generation of 3D antibody-antigen structure dataset.

(A) Formalization of the biological problem of antibody-antigen binding prediction. The biological formulation of the problem may involve either antibody or antigen, subsets thereof (paratope, epitope), sequence or structure or any combination thereof. The choice of data encoding can be divided into sequence-based and structure-based ones, while hybrid formalizations leverage both types of datasets (Figure S1). On the ML side, predicting whether an antibody sequence binds to an antigen may be broadly grouped into classification and regression. The former may involve two categories: binary classification or more multi-class or multi-label prediction. The latter may be used to predict the affinity of an antibody sequence to a target antigen. Furthermore, specific problems of antibody-antigen binding may be, for instance, to predict which residues of an antibody or an antigen are involved in the binding. (B) Antibody-antigen binding data generation pipeline using the Absolut! framework: the PDB structure of an antigen is transformed into a lattice representation where the CDRH3 sequences can be tested (6.8 million binding poses per each 11-mer of each CDRH3 sequence and antigen, Figure S2) for identifying the energetically optimal binding structure. The antigen discretization step (see Methods) transforms the antigen PDB structure into a 3D lattice using LatFit (*47*, *48*), where antibody and antigens are defined by a starting position and a string of letters representing moves on the lattice (Straight, Up, Down, Left, Right, see Figure S3A). The presence of glycans on the protein is parsed from the original PDB and included in the lattice representation. For each optimal antigen-binding structure, the 3D paratope, 3D-epitope and affinity values are obtained and recorded. Using 7.3 million experimental CDRH3 sequences (*49*), we thus produced a database (antibody-antigen binding matrix) of 6.9 million (CDRH3s of length 11 or more) x 159 of antibody-antigen binding pairs with 3D paratope+epitope+affinity resolution. (C) The generated dataset encompasses different levels of biological complexity of antibody-antigen binding in terms of (i) sequential dependencies, (ii) binding hotspots ("immunogenic regions"), showing clusters of epitopes that share residues. Turquoise residues are contained in all epitopes binding this region, while levels of red show how often a residue is contained in an epitope. (iii) The paratope-epitope reactivity network can show the organization of binding pairs according to a specific encoding and can be compared to atomistic experimental 3D-antibody-antigen binding data (see Figure 2 for exploration of the biological complexity of the synthetic antibody-antigen binding data).

Of note, all aforementioned tasks may also be formulated starting from the antigen perspective and predict their binding to predefined antibodies (bidirectionality of antibody-antigen binding prediction). Therefore, similar antibody-antigen binding prediction problems refer to a set of different biological and ML problem formulations, and current publications each study one formulation separately. Each formulation requires different types of dataset structures (class definition, encoding selection and class balancing) and encodings, rendering it challenging to directly compare their effectiveness. A common, large synthetic dataset that can be processed into immunology and ML-task adapted training datasets would enable the comparison of ML formulations for antibody-antigen prediction problems.

1 billion Absolut!-generated *in silico* antibody-antigen structures as a basis for ML benchmark and ML antibody-antigen binding task formulation

Currently available experimental datasets lack sufficient granularity for developing and benchmarking novel immune-adapted ML methods (e.g., small training datasets, lacking paratope-epitope information and/or affinity). Furthermore, the methodical exploration of formalized ML tasks for antibody specificity prediction requires ground-truth complete-knowledge datasets (i.e., datasets containing validated properties to be learned), which are typically non-existent in the experimental realm so far. Therefore, we here developed the Absolut! framework that enables on-demand generation of large synthetic datasets for 3D-antibody-antigen binding generated under the same, deterministic settings at a tractable computing cost.

The Absolut! pipeline simulates the binding of antibody sequences (CDRH3) to antigens (from PDB) *in silico* (Figure 1B), using a lattice representation of protein interactions. Both the antigen and the antibody CDRH3 loop are represented as 3D chains of consecutive amino acids (AAs) in a Euclidean lattice (Figure S2A, Methods), and only neighboring but not covalent AAs interact with a strength defined by the Miyazawa-Jernigan energy potential (a statistical interaction potential inferred from experimental protein structures) (*50*). The lattice representation enables the enumeration of all possible antibody binding structures (Figure S2B, Methods).

Generation of synthetic 3D-antibody-antigen binding complexes with Absolut! encompasses several steps: First, Absolut! creates a lattice representation of the antigen, a step called discretization (Figure 1B), by minimization of the dRMSD between the original PDB structure and its many possible lattice counterparts using the software LatFit (*48*) (Figure S3A, Methods). The lattice antigen is refined by blocking "aberrant geometries" and by modeling topological hindrance from protein glycosylation as "inaccessible positions" on the surface of the synthetic antigen (Figure S3A,B).

We generated a library of synthetic 3D-antigens by discretizing the PDB structure of 159 antigens from crystallized antibody-antigen complexes (*15*) and optimized the lattice resolution (distance between neighboring AAs) to reach an average RMSD resolution of 3.5Å compared to the original PDB (Figure S3C–F), which is near the the median resolution of PDB antigens (2.7Å, Figure S4C). The 159 antigens ranged between 27 to 568 AAs with a median of 159 AAs, and represent a large variety of pathogenic antigens as well as self-proteins (Figure S4–B), see Table S1 for the full antigen list. The discretization step allows the generation of antigens that capture realistic 3D antigen shapes and we confirmed that the discretization step also preserves the surface AA composition of the antigens (Figure S7A).

Second, Absolut! enables the calculation of the energetically optimal binding of a CDRH3 sequence to a lattice-discretized antigen. Absolut! enumerates all possible binding poses of the CDRH3 to the antigen (*51*), calculates their binding energy (scoring function) using the Miyazawa-Jernigan energy potential (*50*), and returns the best pose as the "binding structure" of the CDRH3 (Figure S2B, Methods), a step we termed "exhaustive docking" (Figure 1B). Each binding structure provides 3D-information on paratope, epitope and affinity. Exhaustive docking explores a large conformational ensemble of binding poses for each CDRH3 (*13*) and ensures that the energetically optimal binding is always found – as opposed to commonly used docking methods that follow optimization or heuristic strategies and have no guarantee of finding a global optimal solution. In view of the aims to define ML problems and to simulate benchmark data, the advantage of exhaustive docking is twofold: (i) it prevents ML from learning docking-tool-specific biases instead of the physical antibody-antigen binding rules, and (ii) screens the entire epitope space.

We calculated the binding of a database of 7.3 million unique CDRH3 antibody sequences obtained from experimental data (murine naïve B-cell receptor repertoires) (*49*) to the 159 discretized antigens (Figure 1B, Figure S4B). Using experimental CDRH3 sequences ensures that antibody sequences possess, analogously to the antigen, physiological AA composition and positional dependencies. The CDRH3 were assessed for binding by sub-peptides of size 11 AAs for the purpose of computational tractability (Figure 1B), and only 6.9 million CDRH3s of size 11 or more were considered. The generated database contains 1.1 billion antibody-antigen binding structures with conformational paratope, conformational epitope and affinity resolution (159 x 6.9×10^6 affinity matrix, Figure 1B). This dataset is several orders of magnitude larger than what is currently feasible to generate experimentally (*18*).

The Absolut! antibody-antigen binding dataset is not only ultra-large but also extendable. Human CDRH3 (*52*) can be used instead of murine ones, and showed similar binding behavior in terms of affinity distribution and epitope convergence (Figure S4H,I). In terms of computational requirements to extend the database, each 11-mer of a CDRH3 sequence required on average the enumeration of 6.8 millions binding poses per antigen (Figure S4D), depending on the antigen size (Figure S4E). The average exhaustive docking time was 7.9 seconds per CDRH3 and 16000 CPU hours per antigen for a total of 6.9 million CDRH3 with linear dependence on antigen size (Figure S4F). This is a fairly moderate computational requirement in view of the 6.9 million CDRH3 times 6.8 million docking poses for each antigen, showing the wide applicability and accessibility of Absolut!. Therefore, the Absolut! software allows the user to

generate custom datasets by discretizing new antigens or annotating custom CDRH3 sequences enabling the on-demand generation of large synthetic datasets for 3D-antibody-antigen binding generated under the same, deterministic settings, with accessible computing resources.

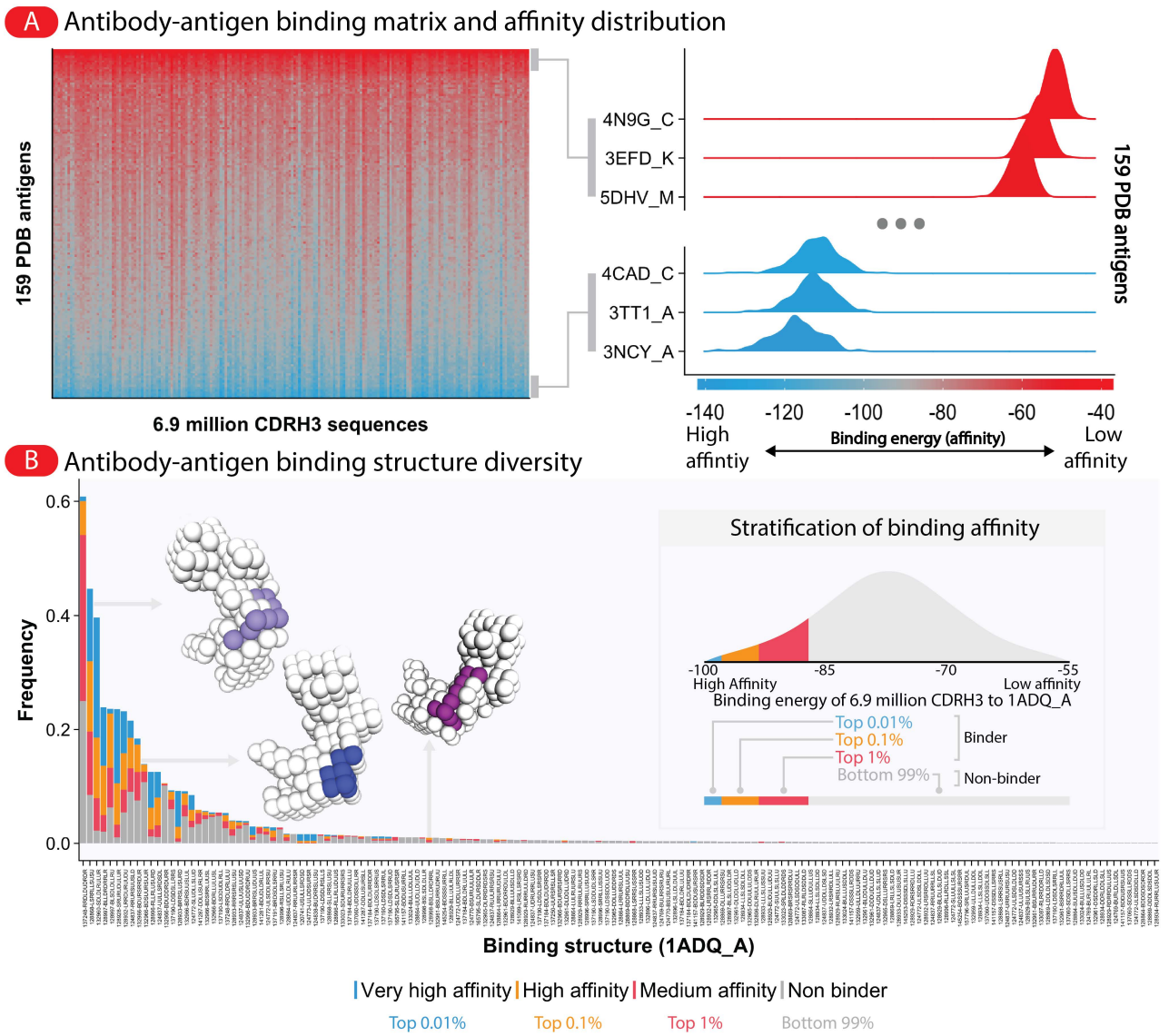
Finally, we validated that the Absolut!-generated data reflects to a certain extent the biological complexity encountered in experimental antibody-antigen binding (Figure 1C, see details in Figure 2). Furthermore, we confirmed that the inclusion of glycans on lattice antigens impacted the binding affinity of the CDRH3 sequences (Figure S4G) as observed experimentally (*53*, *54*). Additionally, for ML analyses, Absolut!-generated data enables extracting a large amount of information on interacting residues that can be used for encoding paratope, epitope and paratope-epitope interaction features (Figure S5), and mirroring the diversity of used encodings in ML studies, from binary vectors (*22*) to 3D distance representations (*55*, *56*).

Altogether, the Absolut!-generated database of *in silico* antibody-antigen binding complexes enables the unconstrained generation of antibody-antigen datasets designed specifically for different ML problems.

Absolut!-generated datasets reflect multiple levels of the biological antibody-antigen binding complexity

To confirm that the Absolut! database is suitable to explore ML strategies for antibody-antigen binding prediction at both the sequence and structural level, we explored to which extent the database reflects different levels of the biological complexity of antibody-antigen binding. The database contains the binding energy and structural conformation of 6.9 million CDRH3 binding to 159 antigens (Figure 2A, left). The distribution of binding energies (affinities) of the CDRH3 sequences to different antigens (Figure 2A, right) shows a wide range of affinities.

In the Absolut! framework, antibody sequences bind to an antigen with a large diversity of binding structures, as shown as an example, for antigen 1ADQ_A (Figure 2B). The binding of a CDRH3 is assessed on all contiguous subsequences of 11 AAs (see *Methods*). We define "binder" to an antigen as the 11-mer sequences within the top 1% affinity threshold (1% lowest binding energies of CDRH3 sequences for a given antigen, Figure 2B, inset, see Methods). This threshold yielded on average 53 000 unique 11-mer binder (CDR3-derived subsequences) per antigen (Figure S8A). The number of binders is arbitrarily extendable by processing additional CDRH3 sequences using Absolut!. Given that we obtain a distribution of affinities for a set of CDRH3 sequences or 11-mers, any other thresholds to delineate binders from non-binders (or more subtle affinity classes) may be chosen. Specifically, grouping CDRH3 sequences (or 11-mers) by affinity class may be relevant for exploring different definitions of negative (non-binder) samples in datasets, which are more or less distant from the top binder threshold, thereby enabling the testing of computational methods for antibody-antigen binding at different levels of granularity. Among the 159 antigens, the top 1% binders used on average 62 different binding structures binding on average 17 distinct epitopes on each antigen, while sequences along all affinity classes explored on average 374 different binding structures defining on average 64 different epitopes and two binding hotspots per antigen (Figure S4J–L). Certain binding structures were used by sequences belonging to different affinity classes (Figure 2B, one bar with 4 colors reflects a single binding structure containing all four binding classes) which is in line with experimental observations (*57*–*59*). Therefore, this framework reflects a substantial level of the structural complexity of antibody-antigen binding.


A Antibody-antigen binding matrix and affinity distribution
159 PDB antigens
6.9 million CDRH3 sequences
4N9G_C
3EFD_K
5DHV_M
4CAD_C
3TT1_A
3NCY_A
159 PDB antigens
-140
-120
-100
-80
-60
-40
High affintiy
Binding energy (affinity)
Low affinity
B Antibody-antigen binding structure diversity
Frequency
0.6
0.4
0.2
0.0
Stratification of binding affinity
-100
-85
-70
-55
High Affinity
Low affinity
Binding energy of 6.9 million CDRH3 to 1ADQ_A
Top 0.01%
Top 0.1%
Top 1%
Bottom 99%
Binder
Non-binder
Binding structure (1ADQ_A)
Very high affinity
High affinity
Medium affinity
Non binder
Top 0.01%
Top 0.1%
Top 1%
Bottom 99%

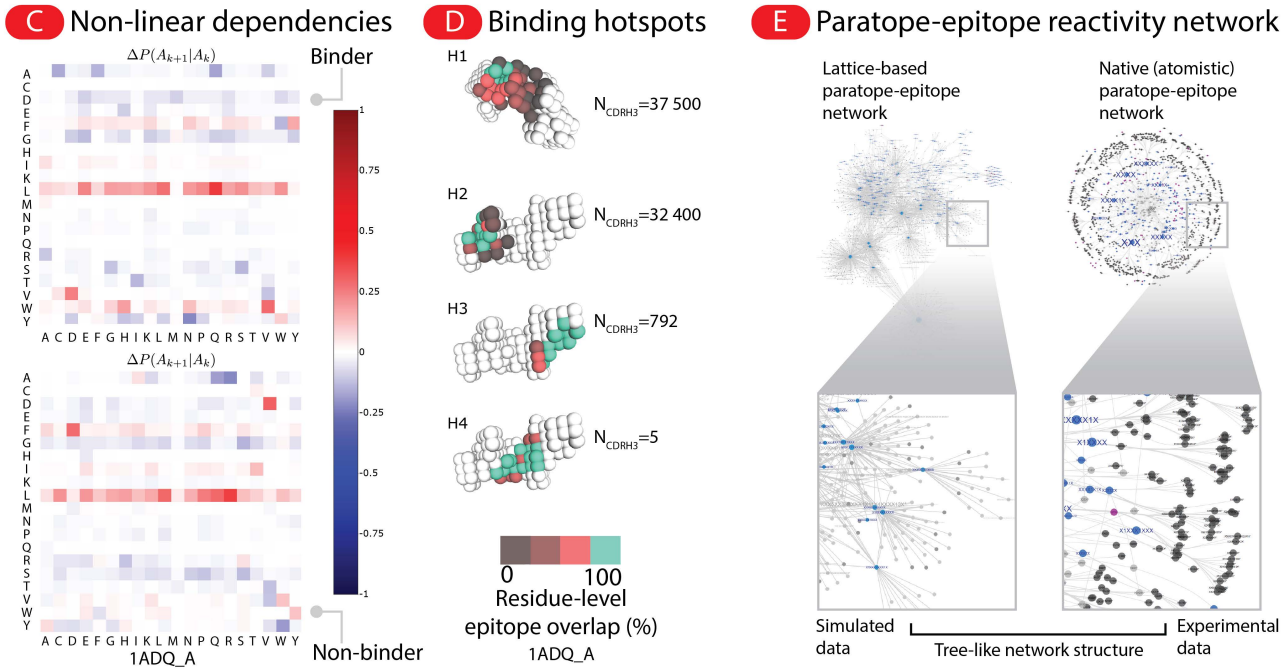

C Non-linear dependencies
$\Delta P(A_{k+1}|A_k)$
Binder
A C D E F G H I K L M N P Q R S T V W Y
$\Delta P(A_{k+1}|A_k)$
1ADQ_A
Non-binder
D Binding hotspots
H1
N_{CDRH3}=37 500
H2
N_{CDRH3}=32 400
H3
N_{CDRH3}=792
H4
N_{CDRH3}=5
0
100
Residue-level epitope overlap (%)
1ADQ_A
E Paratope-epitope reactivity network
Lattice-based paratope-epitope network
Native (atomistic) paratope-epitope network
Simulated data
Tree-like network structure
Experimental data

Figure 2 | The Absolut! dataset reflects granular levels of the complexity of antibody-antigen binding.
(A) Affinity of the 1 113 000 000 antibody-antigen pairs in the Absolut! database, shown as an affinity matrix of 159 antigens and 6.9 million CDRH3 sequences (left panel). Each heatmap tile represents the binding energy (affinity) of a CDRH3 sequence against an antigen. The right panel illustrates the affinity distributions of the top and bottom three antigens in terms of median affinity. (B) Diversity of binding structures in the dataset for antigen 1ADQ_A (see Methods). [Inset] Affinity annotation in (A) allows custom-stratification of CDRH3 sequences into binders and substratification thereof (the top 1% of affinity sorted CDRH3 sequences for each antigen separately) and non-binders (the bottom 99% of affinity sorted CDRH3 sequences). [Main panel] Distribution of 3D-binding structures with respect to each binding class as defined in the inset. The x-axis represents the antibody binding structures to this antigen, written as a 6-digit number representing a starting 3D position in the lattice, followed by a list of moves in 3D-space (see Figure S3A or *Methods* for details). Different CDRH3 sequences may converge to the same binding structure albeit with different affinities (i.e., one bar with 4 colors reflects a single binding structure containing all four binding classes). An affinity class could be distributed across multiple binding structures (i.e., blue bars: very high-affinity sequences, are found across multiple structures in the x-axis). Only 159 of the total 842 identified binding structures are shown for legibility, only 65 of them are used by the top 1% binding sequences. (C) Positional dependencies are illustrated by comparing the observed conditional probabilities of an AA A_k at position k, knowing the AA A_{k-1} at the previous position ($p(A_{k+1}|A_k)$, top), and vice-versa ($p(A_k|A_{k+1})$, bottom), between binders and non-binders ($\Delta p(A_{k+1}|A_k) = p(A_{k+1}|A_k)$ (binder) - $p(A_{k+1}|A_k)$ (non-binder). Positional dependencies at longer range, are shown in see Figure S6 for AAs within a distance up to 5 and 11 AAs, instead of only 2 AAs here). Red squares denote a case where two consecutive AAs are more highly dependent in binders than in non-binders, while blue squares show more negatively dependent consecutive AA, in binders than in non-binders. (D) Binding hotspots identified for antigen 1ADQ_A, named H1 to H4 (see Methods and Figure S18). (E) The paratope-epitope binding network (reactivity network) constructed with the Absolut! dataset (left) shares a complex topology with the oligo- and polyspecific characteristics also found in experimentally determined reactivity networks (*3*, *15*). Both networks represent paratope-epitope matching according to a predefined structural encoding, here as gapped structural interaction motifs, for instance, epitope X1X2X3XX82X3XXXXXXX and paratope XXXX1XX1XXX where X denotes an interacting residue and the numbers refer to the number of non-interacting residues in-between (see Methods and (*3*)). The Absolut!-generated network contains 6092 unique paratope-epitope pairs, with 2572 unique epitopes and 324 unique paratopes with this encoding. Epitopes are shown in blue and purple, and paratopes appear in light or dark gray. Specific epitopes (purple) are defined as being bound by only one paratope in the dataset (all binder sequences use this paratope-epitope pair), while specific paratopes (light gray) are only bound by one epitope.

The positional dependencies of binder sequences to 1ADQ relative to non-binder sequences are shown as the differential chance of having a certain AA knowing the AA in another position (Figure 2C for two adjacent positions A_k and A_{k+1}, and Figure S6 for any position within a range of 5 or 11 AAs as well as AA positional frequencies). Some AAs showed higher frequencies independently of the position, while some AAs were enriched only following specific AAs (red isolated dots), thus showing positional dependencies. Therefore, the properties of Absolut! binding sequences to an antigen harbor complex positional dependencies reminiscent of the natural properties of antibody-antigen binding (*3*, *32*).

Clustering the binding sequences to 20 different antigens showed that very similar antibody sequences in terms of Levenshtein distance (LD) can bind different antigens and sequence-wise very different antibodies can bind the same antigen (Figure S11F), which is a current challenge for ML prediction tasks applied to experimental antibody-antigen binding (*32*) and is likely a consequence of the complex positional dependencies between AAs at different positions incorporated in Absolut! datasets (Figure 2C). The latent space representation of binding sequences to each of the 20 antigens (Figure S11G–H) did not manage to cluster antigen-specific sequences, but rather clustered similar sequences together, showing that sequence similarity is a suboptimal measure for determining the extent of shared binding properties.

To better understand the properties of the *in silico* binding database, we have developed an algorithm to cluster the modes of antibody binding into, hereafter called, "binding hotspots" (see Methods, Figure S18). Specifically, the epitopes associated with binder sequences are clustered into binding hotspots mirroring immunogenic regions of an antigen (*60*). We show in particular the identified binding hotspots for the top 1%-affinity binder sequences to antigen 1ADQ_A (Figure 2D). Each cluster is defined by a core set of

shared epitope positions between different epitopes (turquoise) and surrounding antigen residues with different intensities of red depending on how many epitopes contain these residues. White areas denote antigenic regions that are more challenging to bind with high affinity, as no predicted binding from the top 1% binders had its epitope in these regions from the 6.9 million CDRH3 sequences. The number of binding sequences to each epitope ranged from 5 to 37 500 CDRH3 sequences for antigen 1ADQ_A (Figure 2D), suggesting the Absolut! database mirrors the hierarchy of more or less immunogenic domains within the same antigen (*61*). The binding hotspots of all 159 investigated antigens are shown in Figure S7.

To validate that binding hotspots represent immunogenic regions of the antigens, we asked whether the experimentally observed PDB antibody-antigen crystal structures) coincided with Absolut! binding hotspots (Figure S7B-E), and assessed how many residues in the antibody crystal structure fall within binding distance to a hotspot residue in the lattice antigen. In 75% of the experimental complexes, the paratope was in contact with hotspot core residues (partially due at times to the hotspots being large or the antigen being small), and in 85% of the complexes, the hotspot side residues were in contact with the experimental antibody structure (Figure S7D). We observed that the lattice framework favored concave pockets and that the predicted hotspots of natural antibody-antigen complexes with concave pockets overlap better with the *in silico* counterpart (Figure S7E). Therefore, although the lattice system cannot predict exactly where the antibody binds, which is understandable due to the simplification of binding angles in the lattice system, we can still observe a non-negligible overlap between predicted and experimental bindings, suggesting that several biologically relevant structural properties of antibody-antigen binding are well captured to already benchmark ML methods on these complex properties.

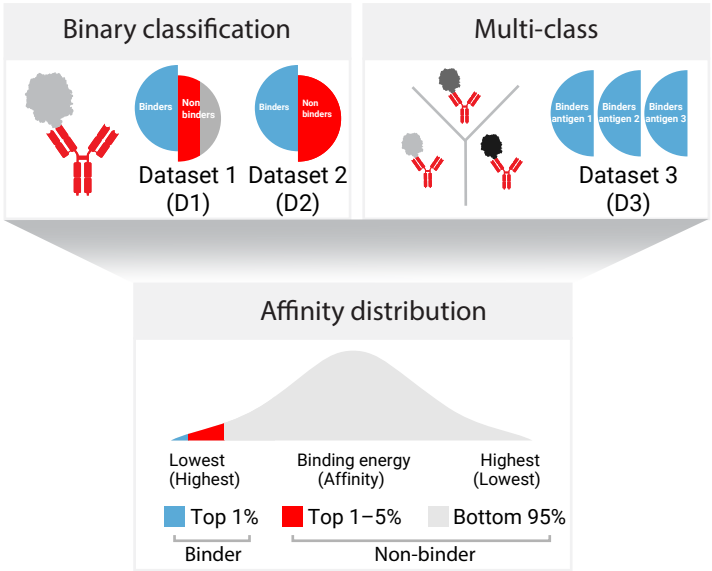
ML tasks often predict which antibody sequence binds to which antigen, according to specific encodings. The reactivity network of Absolut! links 53 000 binder sequences on average to each antigen. Here, we compare the reactivity networks of Absolut! (Figure 2E, left) and experimentally determined binders (Figure 2F, right, generated from 825 experimentally-determined antibody-antigen pairs (*3*)) according to such encoding. In both synthetic and experimental data, we observed a complex organization (tree-like structure) and mutually exclusive polyreactivity of certain binding modes (motifs).

Taken all together, although simplifying antibody-antigen binding, Absolut! encompasses substantial levels of structure- and sequence-based complexity levels, a subset of which are encountered in experimental antibody-antigen binding highlighting the relevance of Absolut!-generated data for developing and benchmarking antibody-antigen binding prediction methods.

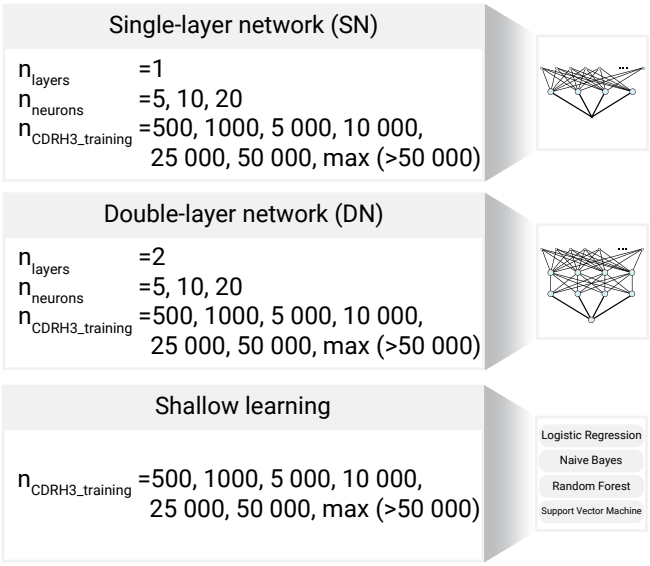
Absolut! allows prospective evaluations of ML strategies with custom-designed datasets

To demonstrate Absolut!'s usefulness in assessing the efficacy of different binary and multi-class classification ML strategies (Figure 1A), we compiled three datasets (Figure 3A), using the binding affinity of all 11-mers derived from CDRH3 sequences against a set of 159 antigens from the Absolut! database (Figure 2A). Nota bene, we do not aim to optimize the architectures and hyperparameters of the machine learning models. The investigations shown merely showcase the utility of Absolut! in creating custom-designed datasets with varying specifications that can be used to explore and train a wide range of ML methods for antibody specificity research.

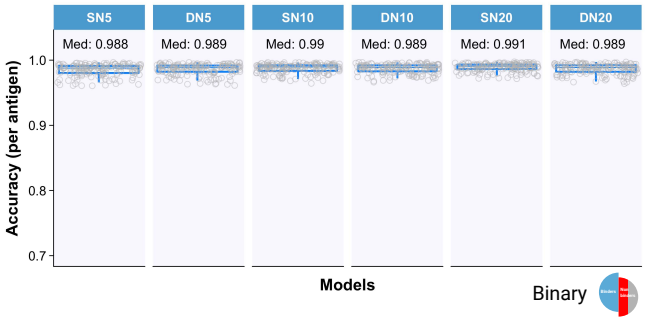
A ML task and dataset specification



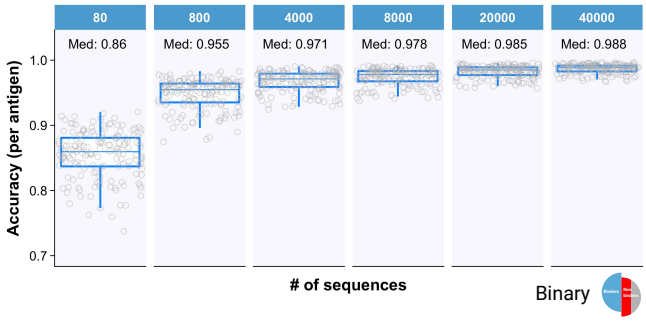
B Deep and shallow learning models



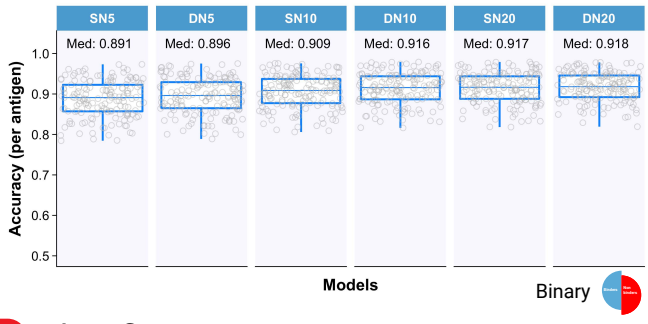
C Classification accuracy on D1



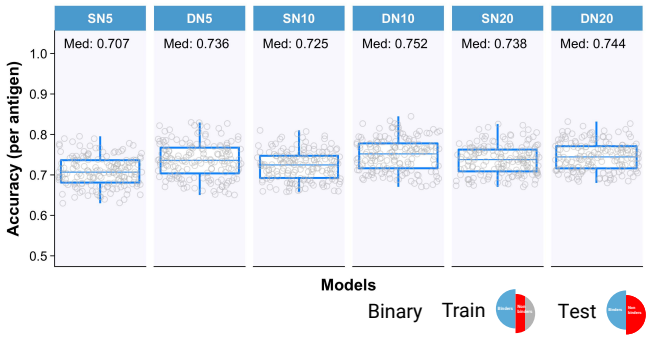
D Classification accuracy vs training size on D1



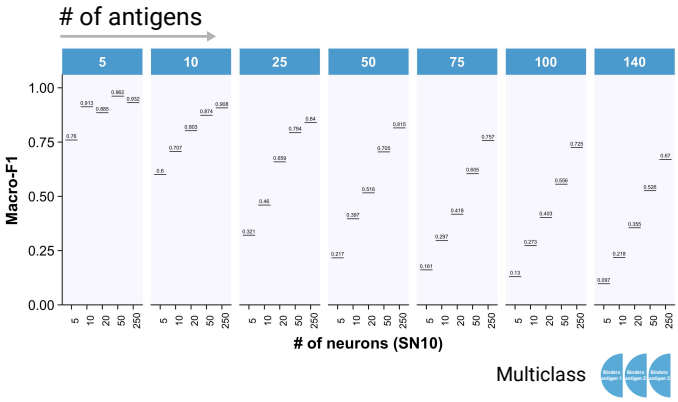
E Classification accuracy on D2



F Classification accuracy on D2; training on D1



G Classification accuracy on D3



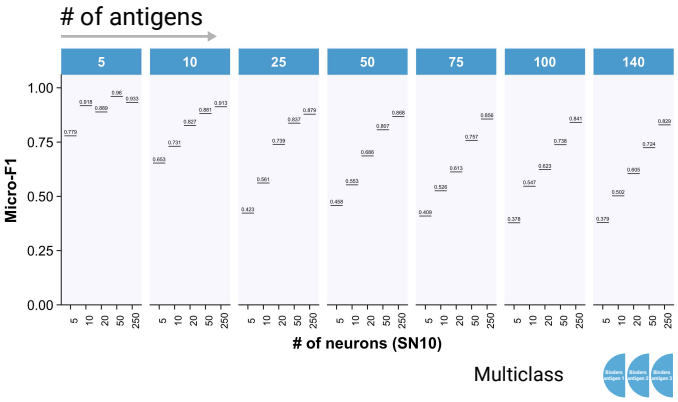


Figure 3 | Classification of binding and non-binding antibody sequences with machine learning.
(A) Design of datasets for the ML task of classifying a sequence as binding in a binary (binding, non-binding) or multi-class (binding to multiple antigens possible) setting. Binding is assessed for sequences of size 11 AAs derived from CDRH3 sequences (see Methods Figure S8). For each antigen separately, we defined binders as 11-mer sequences within the top 1% affinity threshold (blue), non-binders as 11-mer from the top 1–5% (red), and the bottom 95% (grey). Datasets 1 (D1) and 2 (D2) have been created for binary classification for each antigen separately, and D3 for multi-class classification for a set of $n_{antigens}$ antigens (Figure S8). D1 was generated with 50% binders and the other 50% is a mixture of 1–5% non-binders and bottom 95% non-binders (blue versus red+grey), while D2 contains 50% binders and 50% 1–5% non-binders (blue versus red), rendering this dataset a priori more challenging to classify since binders and non-binders are closer affinity-wise. In both datasets, binders and non-binders were sampled as 11-mers taken from CDRH3 sequences of the same length distribution, to limit potential AA composition bias (see Methods). D3 was generated by pooling sequences that were among the top 1% binders across multiple antigens, labeled with the name of their antigen, discarding 11-mers that bound multiple antigens to fit a multi-class classification problem. We also discarded antigens representing variants of the same protein, leaving 142 non-redundant antigens, i.e., different labels (see Table S1 for the list of discarded antigens). Due to the removal of cross-reactive sequences, D3 is imbalanced, because the amount of binder sequences to an antigen that was cross-reactive (with another selected antigen) were highly variable, as seen by the number of available sequences for D3 (Figure S8C). (B) Model architectures considered: Feed-forward neural network with a single layer (SN), feed-forward neural network with double-layers (DN) of equal number of neurons (5–20 neurons; in this Figure, SN5 indicates a single-layer network with 5 neurons and DN5 indicates a double-layer network with 5 neurons), and four shallow learning methods namely logistic regression (LR), Naive Bayes (NB), random forest (RF), and support vector machine (SVM) (see Figure S9 and Figure S10). The ML models were trained on the two tasks: binary classification (binder or non-binder) on D1 and D2 for each antigen separately, and multi-class classification (prediction of an antibody's target out of $n_{antigens}$ targets) on D3. $n_{training}$=80 to 40 000 sequences were used for D1 and D2 datasets as training dataset while $n_{testing}$=40 000 sequences were always used for testing, to avoid a bias of test size. For D3, $n_{training}$=200 000 sequences were used for training and $n_{testing}$=100 000 sequences for testing (or scaled-down when less than 300 000 sequences were available, see Methods for details). (C–F) Accuracy of binary classification prediction of the considered model architectures per antigen, defined as the percent of exact prediction (since D1 and D2 are perfectly balanced, this is equivalent to balanced accuracy [BACC: (Sens + Spec)/2]). One point represents the accuracy to the dataset (D1 or D2) of one antigen (each point is the average of 10 independent replicates, 159 points in total). Median accuracies among the antigens are shown in each panel. Prediction accuracies of shallow architectures and conditions are shown in Figure S9, including a control with shuffled labels (i.e., where the causal link between paratopes and epitopes was removed) to quantify prediction accuracy on randomized data. (C) All models, trained on 40 000 sequences from D1 irrespective of their network architectures, yielded accurate predictions for the binary classification task with median accuracy values ranging between 0.988–0.991 (see Figure S9 for shallow learning). (D) Prediction accuracy improved as a function of the number of training sequences (here shown for the DN10 architecture). The model trained on the least number of sequences ($n_{training}$=80) yielded the lowest accuracy (0.86) whereas the model trained on the largest dataset ($n_{training}$=40 000) yielded the highest accuracy (0.988). (E) Prediction accuracy (trained on 40 000 sequences) was slightly lower on D2 (compared to D1in (C)) with median accuracy values ranging between 0.891–0.918. (F) Models trained on D1 and tested on D2 yielded notably lower performance with median accuracy values ranging between 0.704–0.744. (G) Multi-class prediction accuracy improved as a function of model complexity (number of neurons) and decreased as a function of the number of antigens (classes). We quantified the accuracy as macro (left) and micro-F1 (right) scores (*62*) due to the class imbalance of D3 and visualized the resulting distributions as boxplots. 10 independent replicates are shown as boxplot for each condition and show very little variation. Macro-averaging weights each class equally whereas micro-averaging gives the same weight to each sequence (see Methods). Specifically, the macro-F1 values for the model with the least number of classes ($n_{antigens}$=5) ranged between 0.76–0.962 whereas the model with the most number of classes ($n_{antigens}$=140) yielded macro-F1 values between 0.097–0.67. Micro-F1 values for $n_{antigens}$=5 ranged between 0.79–0.96 and micro-F1 values for $n_{antigens}$=140 ranged between 0.379–0.829. Baseline shuffled (expected) accuracy ranged between 0–0.02 and 0–0.03 for macro-F1 and micro-F1 respectively (Figure S10).

Binary classification predicts if an antibody sequence binds to an antigen. Datasets 1 (D1) and 2 (D2) were generated for each antigen separately, and contain 40 000 of the top 1% affinity-sorted sequences as positive class (binders, Figure 3A, blue) and 40 000 sequences of the bottom 99% sequences (D1, Figure 3A, red and grey) or the top 1%–5% sequences (D2, Figure 3A, red) as negative class (non-binders). Details on D1 and D2 generation are shown in Methods and Figure S8A.

Multi-class classification aims at identifying which of N antigens (class) is bound by an antibody sequence. Dataset 3 (D3) was generated for a set of 5 to 140 randomly chosen antigens, by pooling the top 1% binder sequences to each antigen (Figure 3A), while discarding those sequences that are cross-reactive within these N antigens, in order to fall into a "multi-class" problem (one label per sequence only). The number of available sequences for each D3 ML training depends on the number of selected antigens and reached >500 000 monospecific sequences when pooling binders to 10 antigens or more (Figure S8B). As a comparison, including cross-reactive sequences in D3 would reach up to 1 250 000 sequences to all antigens (Figure S8C). As shown by the minimum and average number of available sequences per antigen (Figure S8D,E), some antigens only had a small number of mono-specific associated sequences and thus their class was smaller than other antigens. Therefore dataset D3 is imbalanced.

We compared the efficacy on both binary and multi-class classification of a set of shallow learning methods, namely Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM) and a single layer neural network (SN), as well as a double-layer network (DN) (Figure 3B). SN*n* denotes a single-layer neural network with *n* neurons in the hidden layer, while DN*n* denotes a double-layer neural network with two hidden layers of *n* neurons each. For brevity, we showed the results for SN and DN in Figure 3C–G and the remainder (LR, RF, NB and SVM together with AA composition and shuffling controls for SN) in Figure S9.

Irrespective of the SN or DN architecture of the model, we recorded high accuracy (0.982–0.99) for the binary classification use case on dataset 1 (Figure 3C). Subsequently, we examined the impact of the number of training sequences on prediction accuracy and found that >4 000 training sequences were sufficient to reach accuracy values >0.97 (Figure 3D, here shown are the accuracy values for the model DN10 as its predictive performance was found to be comparable to more complicated models SN20 and DN20 in Figure 3C). Prediction accuracy was slightly lower for SN and DN architectures on dataset 2 (0.823–0.918) underlining the impact of the choice of the negative dataset, i.e., the negative samples in the dataset, on the degree of difficulty of the ML task, and on the prediction accuracy (Figure 3E). All models trained on shuffled labels consistently reached 50% accuracy since D1 and D2 are balanced (Figure S9,C). Interestingly, models trained only on the AA composition of the sequences, reached a substantially high accuracy: shallow models reached 0.63 to 0.73 accuracy trained on 50 000 sequences from D1 (Figure S9) while all neural network architectures (SN5-20 and DN5-20) consistently reached 0.974 accuracy on D1 with 40 000 sequences on average across all antigens (Figure S9), which was very close to accuracies using the one-hot encoding (0.989, Figure 3). This suggests that discriminating the top 1% binders from all the remaining sequences is not a very difficult task per se, and was to be expected since very low-affinity sequences are likely very different from high-affinity binders. In comparison, models trained on the AA composition of 40 000 sequences from D2 reached 0.768 to 0.775 accuracy (Figure S9), which is lower than accuracies using the one-hot encoding (0.823–0.918, Figure 3E). We further used an out-of distribution approach to evaluate models that were trained on D1 on their capacity to predict binding in the D2 dataset of the same antigen (Figure 3F) (i.e., assessing accuracy on the distribution-shifted dataset (*63*), and found lower accuracy values (0.63–0.70), showing that training on D1 is not very informative to learn the sharp delineation between binders and non-binders.

As with sequence-based experimental datasets, D1 or D2 contain a mix of sequences that bind to different epitopes on the target antigen, and it is not clear whether epitope-specific sequences could be identified from such datasets. A first approach would be to naïvely assume similar sequences to bind the same epitope. We clustered 25 000 binder sequences to 1ADQ_A (positive class in D1) according to their LD and

could observe 10 clusters (Figure S11A), which did not correlate with the epitope or paratope encoding of these sequences, showing that similar sequences bound to different epitopes.

We used an integrated gradients analysis (*64*) as an attribution method to identify which positions were determinant in the classification of binder sequences in Dataset 1 (Figure S11B-C), as this technique has been used to interpret complex positional dependencies among binding sequences (*32*). The SN10 architecture was trained on D1 of an antigen (1ADQ_A, 40 000 sequences) and each sequence was linked with its gradient weights of the trained model, i.e., a measure on how each AA of this sequence quantitatively decided the output binding probability in the model. We then asked whether the gradient information could help clustering sequences with similar binding properties. The gradients allowed to cluster sequences according to their paratope and epitope encoding (Figure S11B), showing that by just training a model on binder and non-binders, the information learned by a simple neural network can be used not only to find new sequences with high affinity (*25*) but also to identify new sequences with desirable paratope or epitopes. The clustering was not due to a confirmation bias (by chance), as the clusters disappeared after shuffling the labels of each sequence (colors) (Figure S11B). Therefore, the integrated gradients provided learned information that was not available by analysis of similarity, and Absolut! can be used to benchmark the predictive power of attribution methods to cluster sequences by their binding properties.

In the context of multiple antigens, model predictions for the multi-class classification task were assessed using macro and micro-F1 scores (*62*) due to the imbalance of class sizes in D3. The micro-F1 returns an aggregate score for all sequences (samples) in the dataset (therefore more influenced by large classes), while the macro-F1 evaluates an average score per class thereby large and small classes are equally weighted (better describing the prediction for smaller classes when the dataset is imbalanced; see Methods). A robust accuracy across all classes is achieved when both micro and macro-F1 are high. Due to class imbalance, we focus on the macro-F1, as it is a more balanced metric, i.e., a high macro-F1 indicates that the prediction is robust across all classes independently of their sizes. Since SN architectures performed equally well as DN architectures on binary classification, we used SN architectures on multi-class classification with a range of model complexity (number of neurons 5–250). We found that the accuracy values for the multi-class classification decreased as a function of the number of classes (number of antigens) (Figure 3G). Conversely, the accuracy values improved as a function of model complexity (number of neurons) with the macro-F1 values for the largest model ($n_{neurons}$=250) ranging between 0.67–0.932. In comparison, shallow learning models yielded markedly lower macro-F1 accuracies (LR: 0.36–0.66; NB: 0.01–0.34; RF: 0.–0.15; and SVM: 0.35–0.69 for SVM, Figure S10A). Both macro-F1 and micro-F1 scores for the largest model ($n_{neurons}$=250) were higher than those of models trained on D3 with shuffled labels (macro-F1 < 0.02) or using only AA composition (macro-F1: 0.05–0.16, Figure S10B). We note that one-hot encoding yielded macro-F1 accuracies 5.8 to 13.4 times higher than using the AA composition (Figure S10B), indicating that the SN architecture successfully learned complex features of multi-class antigen binding beyond those contained in the AA composition.

Altogether, these investigations show that Absolut! is useful for testing different ML and attribution strategies on different dataset specifications, architectures, and scales. Importantly, it also allows us to examine the influence of different negative samples in the datasets such as in the case of datasets D1 and D2, as well as the construction of multi-class dataset D3.

Impact of structural data on paratope-epitope prediction accuracy

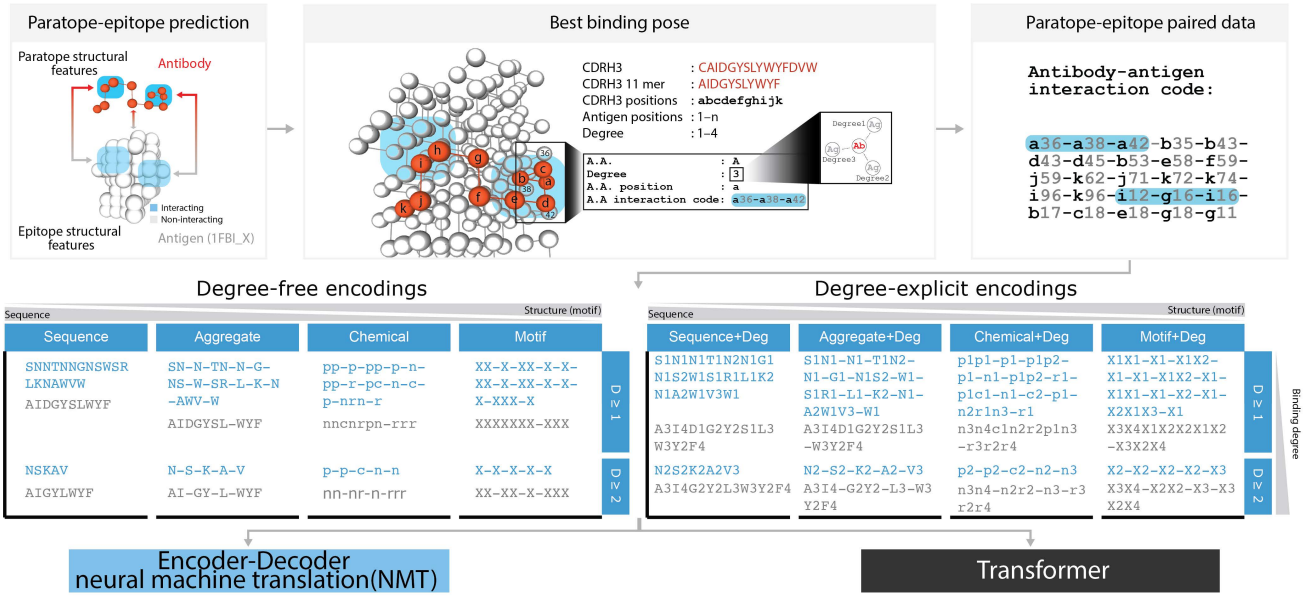
Since the rules of antibody-antigen binding lie within the paratope-epitope interface, we formulate the "paratope-epitope" prediction problem (Figure 4A), as the prediction of the epitope for a given paratope. Previous paratope, epitope or paratope-epitope ML prediction approaches (*8–10*) have used a diverse choice of input feature encoding leveraging both structure- (in some cases predicted for the paratope) and sequence-based features of antibody-antigen interaction, and have yet to investigate the impact of structural information on prediction accuracy. It is very important for the antibody discovery field to understand which levels of information are predictive for antibody-antigen binding, in order to better leverage or design experimental datasets. High accuracy of paratope-epitope prediction would mean that a chosen encoding contains sufficient information to learn binding rules from the dataset given a high enough training size. We have previously shown (*3*) that structural interaction encodings for both paratope and epitope prediction improve predictability. However, at a restricted quantity of experimental data (825 experimental antibody-antigen complexes), we were not able to exhaustively examine the full extent of the utility of such encodings (*3*).

The availability of large-scale structural antibody-antigen binding (complexes) in the Absolut! datasets enables thorough investigation of the contribution of structural, sequence, and hybrid structure-sequence encodings to the performance of ML prediction methods. We used our framework to propose mathematical expressions for antibody-antigen binding (Figure S5), including various encodings of paratopes and epitopes. To avoid ambiguity, we refer to "paratope" and "epitope" (Figure S5A) as the interacting residues (on the antibody and antigen, respectively) including their 3D position (conformational, 3D definition), as compared to the 1D definition of paratope or epitope (list of binding residues only).

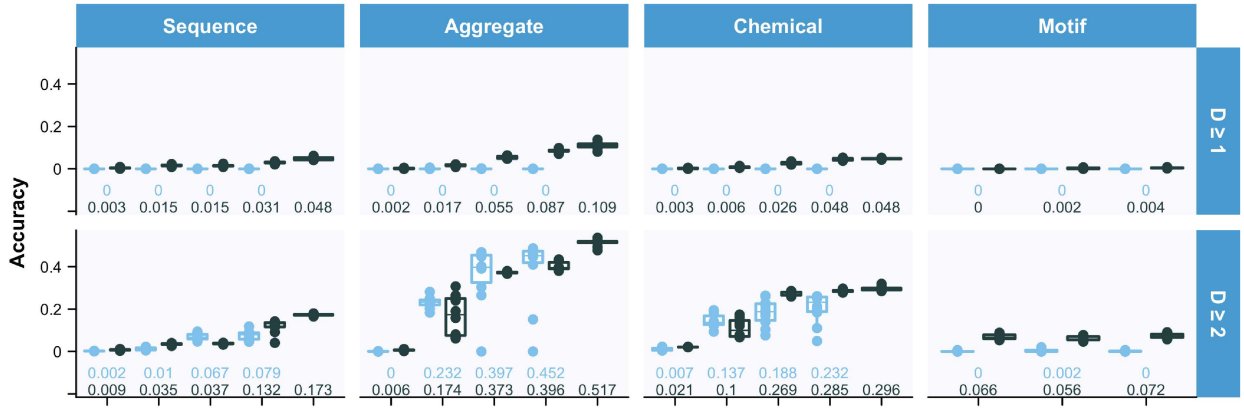
Paratope-epitope prediction may be formalized as either a multi-class prediction problem, where a paratope is labeled by the ID of the epitope, which requires a priori knowledge on all possible epitopes in the dataset; or as a sequence prediction problem, e.g., Natural Language Processing (NLP) problem, where a paratope needs to be "translated" into an epitope by text generation. Since the NLP formulation does not require to list all epitopes as classes, it allows predicting the binding epitope of new sequences (named as "neo-epitopes") and towards new antigens, which makes the NLP formulation a more difficult task than classification, but more useful for purposes such as antibody and vaccine design. We, therefore, decided to focus on the NLP paratope-epitope formalized problem (Figure 4A). A multi-label problem formalization would also be envisioned to learn the set of matching paratopes to an epitope.

We considered a diverse set of possible encodings suitable for NLP (Figure 4A), each differing in the amount of sequence and structural information contained. Degree-free encodings only contain paratope or epitope residues (involved in the binding), with or without indication of gaps ('-') for a group of non-interacting residues between them, and have already been used for ML prediction on experimental antibody-antigen structures (*3*). Residues can be encoded as the AA name (without gap: "Sequence", with gaps: "Aggregate"), their chemical property ("Chemical aggregate"), or just a "X" to denote that binding happens at this position ("Motif"). Each paratope residue can be annotated with its binding degree (i.e., how many binding partners on the antigen), termed "Degree-explicit" encoding, while "Degree-free" encodings do not contain any degree information (in earlier work, Akbar et al. (*3*) used motif, sequence, and aggregate encodings without including the notion of degree nor chemical properties in their encoding).

A Deep learning based paratope-epitope prediction



B Prediction accuracy (degree-free)



C Prediction accuracy (degree-explicit)

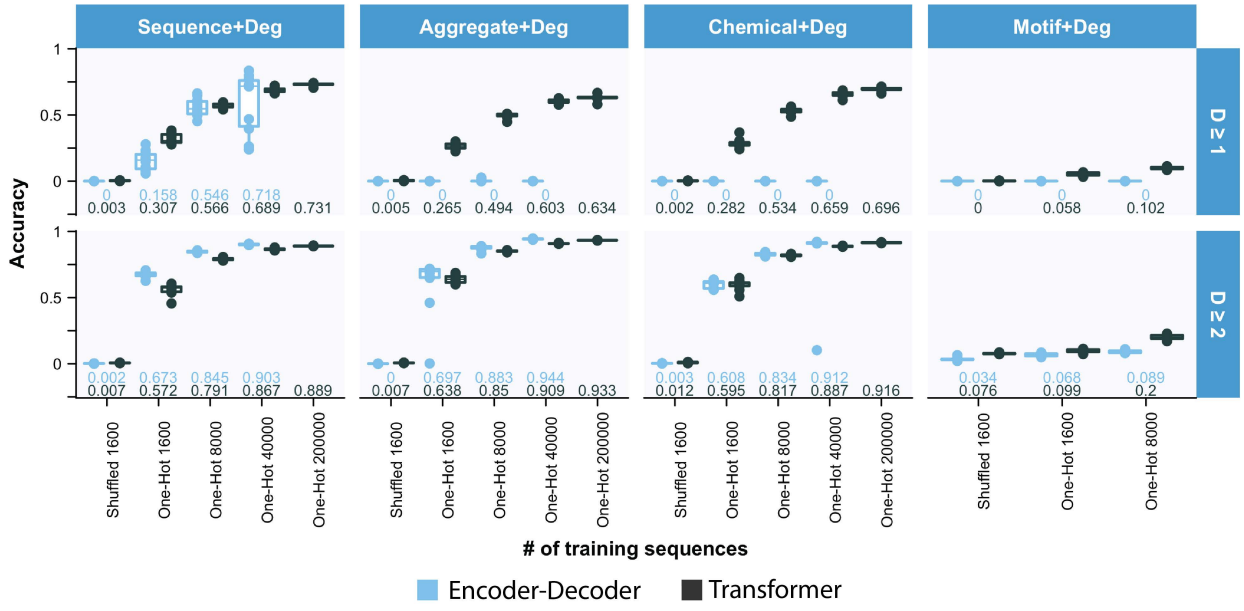


Figure 4 | ML prediction of paratope-epitope pairs involved in antibody-antigen binding.
(A) Problem formulation of paratope-epitope prediction: using a range of encodings of paratope and epitope pairs we aim to predict the epitope encoding from the paratope encoding. From an Absolut! antibody-antigen binding complex (the best binding pose), possible elements that can be incorporated into a paratope-epitope encoding are shown on one example binding (of CDRH3 sequence CAIDGYSLYWYFDVW to antigen 1FBI_X). The interaction code stores which residues on the CDRH3 (denoted by their position from a to l) bind to which residues on the antigen (numbers). From the interaction code, a list of encodings is proposed to describe both sequence and structural information on the paratope and epitope of a binding. Encodings contain the list of binding residues (i.e., paratope or epitope residues), possibly separated by gaps (stretches of non-interacting residues, noted as "-"). Residues can be encoded as their name ("Sequence" without gaps or "Aggregate", when using gaps), their chemical property ("Chemical", with p: polar, r: aromatic, n: non-polar, c: charged) or structural interaction motif where interacting residues are encoded as the character "X". Compared encodings can contain (degree-explicit) or not (degree-free) information on the degree of binding of each residue. Of note, the sequence encoding implicitly contains structural information, as non-binding residues are not included. We leveraged common natural language processing (NLP) architectures for the paratope-epitope prediction: an encoder-decoder with attention (*65*) and the transformer architecture (*66*) (as described in Figure S19). (B,C) Prediction accuracy (exact match of the predicted epitope to the test-set epitope) for both architectures as a function of a set of encodings, for degree-free (B) or degree-explicit (C) encodings, with (upper facet) or without (lower facet) residues binding with degree 1. Each point is an independent training instance and 10 different replicates per condition were performed. Learning on a paratope-epitope shuffled dataset is compared with learning on 1 600–40 000 (and 200 000 only for the Transformer architecture due to computational reasons) unique paratope-epitope pairs, while all conditions were tested on 100 000 unique pairs when available, or downscaled to reach 80% training and 20% testing (see Methods). The "motifs" encodings only allowed for 2 262 to 9 065 unique paratope-epitopes pairs (see Figure S12A), which are referred to as One-Hot (Max) conditions).

Finally, to disentangle the contribution of binding residues with only one binding partner (degree =1), we also considered encodings with only residues of binding degree 2 or more (Figure 4A lower facets). We also considered an encoding where the gaps are represented by their size and residues by an "X" ("Gapped Motifs", Figure S14A,B). Altogether, 19 encodings were compared that lie on the spectrum of containing more sequence information (Figure 4A, left) towards more structural information (Figure 4A, right), and including more (lower) or less (upper) information on the binding degree. Of note, each encoding defines a different biological paratope-epitope problem (Figure 1A, Figure S1). Nota bene, in all subsequently reported ML tasks, we encoded corresponding paratopes and epitopes with identical encodings (i.e., no mixing of encodings within one training dataset).

The same encoded epitope may be present in different paratope-epitope pairs depending on the encoding (Figure S12A, blue histograms). A single epitope was paired with a median of 24 841 paratopes using the sequence, degree-free encoding while this number went down to only 2 paratopes-epitope pairs per epitope in the motif, degree-free encoding. The variability of pairs per epitope in an encoding illustrates the imbalance of the prediction task between different epitopes. The same paratope was paired with a small number of different epitopes (Figure S12A, grey histograms), with a median of 3 paratope-epitopes pairs per paratope using the sequence encoding, to 10 epitopes per paratope in the motif, degree-free encoding. Of note, the ML formulation of learning paratope-epitope pairs is well justified if most of the time, one paratope only has one cognate epitope (which is the case for most encodings here). However, when a paratope encoding matches more than one epitope encoding (more than one possible answer), accuracy levels will be limited, and another ML formulation that accepts more than one exact answer might be better suited for future larger datasets. This asymmetry in paratope-epitope matching classes results from a large amount of CDRH3 sequences (6.9 million) assessed for binding to a small number of antigens (159), which occurred to have a small number of epitopes per antigen (immunodominance), as also observed in experimental datasets. We, therefore, decided to focus on the paratope→epitope prediction problem, for which each epitope is richly represented by many epitope-paratopes pairs, instead of the epitope → paratope prediction problem, which would require

extending the database with more antigens. Although paratope-epitope pairs are distinct in training and test datasets, the 'data leakage' (chance to observe an epitope in both datasets) is shown in Figure S12B,C). The data leakage problem is a non-trivial challenge to solve. On the paratope side, since similar antibody sequences can bind different antigens, one may want to keep them in the training dataset to describe binding to different antigens. Some studies separated paratope sequences such that 95% of similar sequences are only on one dataset (*67*), while most structural-based studies actually kept 99% similar paratopes in training and test datasets, in order to leverage structural homology with training entries (and it is not clear how they would perform if test paratopes would be more distant to the training ones). On the epitope side, one usually aims to predict if new sequences bind on the same antigens, and therefore the same epitope should be present in both training and testing. However, it could be envisioned to keep epitopes outside the training dataset in order to test the capacity of transfer learning. Thanks to the large Absolut! test dataset size, we were always able to find test-specific epitopes and investigate the accuracy separately on test-specific epitopes separately (see below).

We considered two commonly used NLP architectures using attention layers (Figure 4A and Figure S19), due to the existence of long-range positional dependencies (Figure S6): an encoder-decoder with a Bahdanau attention layer (*68*) and a transformer architecture (*66*). Degree-free encodings were processed as sentences of one-letter words (a residue or a gap), while degree-explicit encodings were processed with two-letter words, containing a gap, or a residue and its degree (Figure S19A) as processing with words of size one led to lower accuracies). Shallow architectures (SVM, LR, RF and NB; with one-hot encoding) provided baseline accuracies (Figure S16).

We assessed the performance on the paratope-epitope prediction problem of the two NLP architectures with the different proposed encodings (Figure 4A) and using 1 600 to 200 000 (transformer) or 1 600 to 40 000 (encoder-decoder, due to increased computational costs) unique paratope-epitope pairs for training. Performance was measured by the fraction of either exact prediction match (accuracy) (Figure 4B,C) correct predicted epitope size (Figure S13A) or the LD between predicted and target (test-set) epitope (Figure S13B,C). For each condition, a shuffled dataset control is included (i.e., where the causal link between paratopes and epitopes was removed) to quantify prediction accuracy on randomized data.

The accuracies of both NLP architectures were low using degree-free encodings including all binding residues irrespective of their degree (Figure 4B, upper facet, degree 1 or more). The encoder-decoder failed to predict the cognate epitope. In comparison, the transformer did not produce such repeats). The highest transformer-based accuracy was 0.109 with the aggregate encoding and 0.048 for the chemical encoding, with 200 000 training paratope-epitope pairs, compared to the shuffled control where accuracy remained under 0.003 for all encodings. Altogether, these encodings did not allow for the tested architectures to learn patterns in antibody-antigen binding.

When discarding residues with binding degree of 1, the degree-free encodings yielded higher prediction accuracies, and the encoder-decoder and transformer performed equally well (Figure 4B, lower facet), suggesting that degree 1 residues are not important for binding prediction. The best accuracy was achieved using the aggregate encoding, with 0.456 and 0.396 accuracies for the encoder-decoder and the transformer, respectively, with 40 000 training paratope-epitope pairs (shuffled accuracies were lower: 0 and 0.006 for the two architectures respectively). The transformer reached a 0.517 accuracy when trained on 200 000 sequences. Interestingly, the sequence encoding and the motif encodings yielded markedly lower accuracies using the transformer, 0.173 and 0.072 respectively, still higher than random (0.009 and 0.066 from the shuffled control, respectively).

Therefore, encodings that combine both sequence and structural information (the "Aggregate" and "Chemical" encodings), filtered on residues with degree two or more, are more predictive than sequence-only or structure-only ("Motif") encodings. Of note, the increase in prediction accuracies using the "Aggregate" encoding (compared to the "Sequence" one) was not due to a simplification of the task by a decrease in the number of possible output epitopes (2 128 epitopes associated with 16 276 paratopes each, on average for the "Sequence" encoding and 2 154 epitopes associated with 16 252 paratopes each, on average for the "Aggregate" encoding (Figure S12A))

Since discarding low-degree binding residues enhanced prediction accuracy, we asked whether including the binding degree in the encoding (degree-explicit encodings) would further improve accuracy. By retaining residues of degree 1 in the encoding (Figure 4C, upper facet), the transformer reached high accuracies for all the encodings (between 0.634 and 0.731 for 200 000 training pairs) except for the motif encoding (0.102 accuracy), in comparison to random accuracies between 0 and 0.005 depending on the encoding. The encoder-decoder, however, only performed well on the sequence encoding (0.689 accuracy on 40 000 training sequences), while it failed to predict the epitopes with the other encodings (accuracy 0).

Finally, discarding degree 1 residues in explicit degree encodings further increased the prediction accuracy (Figure 4C, lower facet), for all encodings except the motif one, and the encoder-decoder reached between 0.903 to 0.944 accuracy on 40 000 training pairs, slightly outperforming the transformer architecture that reached 0.867 to 0.909 on the same training dataset size.

In comparison, using "Gapped Motif" encodings lead to very poor accuracies (Figure S14C maximum 0.012) with both NLP architectures, showing that the size of gaps between binding residues does not contain sufficient information to predict paratope-epitope matching.

Therefore, we have shown that datasets of as low as 8 000 training paratope-epitope pairs can be leveraged to obtain high prediction quality (~0.8 accuracy) thanks to the inclusion of structural information inside sequence-based encodings, using gaps and binding degrees. The performance of the encoder-decoder and transformer measured by LD (Figure S13) led to analogous conclusions. Conditions where the encoder-decoder failed to predict the epitopes, were associated with wrong size of the predicted epitope (Figure S13A, due to infinite repeats). The shuffled controls show a different reference level for the LD depending on the encoding. Indeed, degree-explicit are longer while degree-filtered encodings are shorter (Figure 4A), therefore leading to different possible LD values. The highest improvement in LD compared to the shuffle control was achieved with the aggregate, degree-1-filtered, and degree-explicit encoding, going from LD=0.508 under shuffling down to LD=0.019 with the encoder-decoder trained with 40 000 epitope-paratope pairs (Figure S13B).

Altogether, the performance of NLP architectures on the Absolut! dataset validated our previous results (*3*) on experimental antibody-antigen structures, showing that the "Aggregate" encoding enables higher predictivity than the "Sequence" encoding alone on antibody-antigen structures from the AbDb dataset (*15*). Further, Absolut! extends those findings by predicting the impact of encodings on larger future datasets. Interestingly, structural information has recently been shown to be predictive and more data-efficient not only for the prediction of paratope-epitope binding (*3*) but also TCR-antigen binding (*69*).

Although the aforementioned accuracies reveal the capacity of the trained architecture to predict unseen paratope-epitope pairs (test set), we asked whether these architectures can predict paratope-epitope pairs when the epitope was not in the training dataset. Of note, although generated from different paratope-epitope pairs, the training and test datasets pairs share a substantial amount of epitopes (Figure S12B,C). To this end, we re-analyzed the accuracy and LD of predicted epitopes that were present in the training dataset (shared) separately to the ones that are new in the test dataset (test-unique), in (Figure S15A). The accuracy on test-unique epitopes was always zero (Figure S15B) and the LD was always very similar to the shuffled control, showing that neither architecture was able to predict epitopes that were completely absent from the training dataset (see Discussion). In case the trained architectures would manage to partially predict the epitope (for instance a sub-sequence of it), this should be observed by a decreasing LD on the test-unique epitopes. For degree-free encodings, the LD among test-unique epitopes was stable or sometimes increased with the training dataset size (for instance, the chemical, degree-free, degree-filtered encoding, went from 0.455 for the shuffled control to 0.537 for the transformer trained on 200 000 pairs) (Figure S15A). Such increasing LD means that learning on the training epitopes made the prediction worse for the test-unique epitopes. However, using the degree-explicit encodings, the LD decreased on test-unique epitopes in an encoding-specific fashion (e.g., using the chemical, degree-filtered, degree-explicit encoding, LD went from 0.521 in the shuffled group down to 0.482 using the encoder-decoder) (Figure S15A). For each test pair, we assessed the LD of the predicted epitope (compared to the target epitope) depending on how 'far' the target epitope was to the training dataset (defined as the minimum LD between the target epitope and all epitopes present in the training dataset) (Figure S17A,B). Shared epitopes between training and test datasets appear as LD=0 in the x-axis, while epitopes with LD>0 are test-specific. For all encodings that included degree 1 residues (two top rows), the LD of predicted epitopes was equally bad on test-specific target epitopes independently of their distance to the training epitopes (Figure S17A,B). However, degree-specific encodings, and especially "Chemical" and "Aggregate" degree-filtered encodings, showed a better LD of predicted epitopes when they were similar to those in the training data, suggesting that these particular encodings learned properties of neo-epitopes (sub-patterns), albeit without managing to generate the full epitope encoding.

Altogether, using Absolut!, we could give an estimation on the number of sequences (train + test) that are already sufficient to reach high prediction accuracy (5000 sequences for binary classification (Figure 3); 500 000 sequences for multi-class classification (Figure 3), and 10 000 paratope-epitope pairs for paratope-epitope prediction (Figure 4)). Further, we have shown that including structural information (Figure 4), albeit encoded in a simple form (gaps, binding degre), enables high prediction accuracy for paratope-epitope prediction, as a strategy to use moderately small experimental datasets in comparison to using sequence-based datasets only. These results confirm and extend previous findings on experimental antibody-antigen structures, that structural information increases paratope-epitope prediction accuracy (*3*). In summary, Absolut! enables the design of both structural and sequence-based features that meet specific criteria for the benchmarking of different ML tasks.

Discussion

Absolut! allows unconstrained generation of *in silico* 3D-antibody-antigen datasets for ML benchmarking

The prediction of antibody-antigen binding is a key challenge for the fields of antibody immunology as well as antibody and vaccine design (*10*, *37*, *70*). So far, it remains largely unfeasible to resolve this challenge using current experimental data given the scarcity of large-scale antibody-antigen binding datasets with 3D-paratope-epitope resolution (*10*). To overcome the lack of datasets to train ML prediction methods, we developed an antibody-antigen simulation tool (Absolut!) that allows virtually unconstrained generation of *in silico* antibody-antigen binding structures (Figure 1), and thereby imparts the capacity to generate datasets parametrized to an antibody-antigen prediction problem. Along with Absolut!, we provide a database of 1.03 billion antibody-antigen complexes (Figure 2, greifflab.org/Absolut/), obtained from evaluating the binding of 6.9 million CDRH3 to 159 antigens enumerating millions of binding poses per sequence and antigen. The diversity of binding modes explored is exemplified by the detection of on average 62 different antibody-antigen binding structures and 1 to 5 binding hotspots per antigen (Figure S4J–L) for the high-affinity (top 1%) CDRH3 sequences, reflecting experimental binding diversity (*71*). We have formalized biological definitions and characterizations of antibody-antigen binding *in silico* (Figure S5) and translated immunological antibody-antigen binding prediction use cases into ML problems (Figure 1A, Figure S1). Absolut! can generate *in silico* datasets of any size for each of these use cases and problems and with different encodings with only a moderate computational footprint (Figure S4F). As proof of principle for the usefulness of the Absolut! framework! for guiding ML approaches on experimental data, we showed that the inclusion of structural information, as already suggested for antibody-antigen crystal structures (*3*), reduces the number of antibody sequences required for antibody specificity prediction (Figure 3, Figure 4). These investigations are unattainable if attempted on currently available experimental data. As an orthogonal proof of Absolut! applicability to experimental settings, we have trained an ML generative model on 20 000 Absolut!-generated binder sequences to different antigens, and could use Absolut! to test that the newly generated sequences maintained high affinity while exploring a larger sequence space than the training dataset, which we confirmed on experimental data using experimentally-validated discriminative ML models (*72*). Therefore, by generating simulated data that reflect different forms of sequence signals of antigen specificity, Absolut! enables the *in silico* evaluation as well as the benchmarking of computational and ML strategies for antibody-antigen binding prediction prior to experimental exploration.

Absolut! enables ML studies on antibody-antigen binding datasets that exceed the size of existing ones by several orders of magnitude

To our knowledge, this is the first synthetic dataset for structural antibody-antigen binding. In comparison to available experimental datasets, state of the art paratope and epitope prediction tools were applied to 471 (*45*), 460 (*46*), 759 training + 173 testing (*67*), 1692 training + 92 testing (*29*), and 825 (*3*) experimental antibody-antigen binding structures; docking databases for antibody and antigen (whose structures are known both separately and in complex) have so far only accumulated 67 non-redundant antibody-antigen pairs. Although large-scale synthetic datasets such as those generated by Absolut! are best-suited for ML benchmarking, this approach has not been widely used, due to the trade-off between precision (coarse-grained versus atomistic resolution) and computational complexity (computationally greedy molecular dynamics versus fast rigid docking). Specifically, Glanville et al. (*73*), generated synthetic epitopes by (i) identifying a set of possible epitopes by docking 559 crystalized antibody structures towards an influenza A hemagglutinin (HA) structure, (ii) taking the top 10 000 docking poses (resulting in over 3 million unique epitopes/contact residue sets), (iii) expanded the epitope landscape by randomly

subsampling the AA residues in the contact residue sets. which was useful to study the prominence of epitopes in HA variants. However, there is no guarantee that a new antibody sequence would share structural properties with one of the 559 considered structures, and antibody flexibility was not considered (*19*). In comparison, using the Absolut! framework, we assessed for each of the 159 antigens and 6.9 million CDRH3 sequences, on average 6.8 million docking poses (including every possible flexible conformation of the CDRH3) by exhaustive search, with a guarantee of identifying the optimal binding for each CDRH3 sequence with access to both paratope and epitope information. Therefore, the main advantage of Absolut! is the deterministic and fast, and large-scale generation of structurally annotated antibody-antigen bindings for millions of antibody sequences to any antigen (protein structure) of interest, which is not yet feasible by molecular dynamics simulations or docking and surpassing current experimental datasets by several orders of magnitude.

The Absolut! framework reproduces non-trivial properties of antibody-antigen binding

Absolut! allows the symmetric study (paratope→epitope binding, epitope→paratope binding) of antibody-antigen binding by providing both paratope and epitope information (and affinity). For the purpose of developing computational and ML methods for the study of antibody-antigen binding, even coarse-grained models, such as the one presented here, are valuable as long as they (as this model does) provide features that overlap with experimental data as well as operate at least to a certain degree, within a physiological parameter space (Figure 2, Figure S6, Figure S7). In comparison to previous widely used more or less abstract coarse-grained representations of antibody-antigen binding (reviewed in (*74*)), ranging from probabilistic models (*75*, *76*), the shape-space model (*77*), to cubic ligand-receptors (*77*, *78*) or structural coefficients for antibody clones (*79*), Absolut! allows a substantially higher level of intrinsic structural complexity of antibody-antigen binding. Furthermore, Absolut! can generate custom-designed datasets related to many immunological problems and ML formulations (Figure S1, Figure 1), exemplifying and underlining the diversity, complexity, and applicability of Absolut! generated synthetic datasets.

Specifically, Absolut! recapitulates several critical features in terms of physiological parameters of the biology of native-like antibody-antigen binding structures. Discretized antigens are in the <3.5Å dRMSD vicinity of the respective native PDB structure and preserve the surface AA composition (Figure S3); generated binding structures follow an empirical, experimentally derived AA interaction potential between each pair of interacting AAs (the Miyazawa-Jernigan potential) (*80*); the paratope-epitope binding network follows native-like properties (Figure 2E); and possible structural conformations display a high combinatorial complexity (Figure 2B,D, Figure S4C–L). Epitope immunogenicity is captured by binding hotspots (Figure 2D), mirroring experimentally observed or predicted epitope clusters (*71*, *81*) and hotspots were overlapping with the experimental antibody structures most of the time (Figure S7E). Similar CDRH3 sequences were found to bind different antigens (Figure S11D–H), as found in experimental studies where binders and non-binders to the same antigen could be very similar (*32*), and very different sequences bound to the same antigen (Figure S11A–C, (*32*)). Previously, we have shown that the structural lattice model used in Absolut! (*51*), contains additional natural properties of the antibody-antigen immune response: cross-reactivity, accessibility and shielding effects, mutation-induced affinity jumps, and dynamics of affinity maturation (*51*). Furthermore, the Absolut! data structure allows for varied feature representation (e.g., structural information, Figure S5), all binding statistics are completely deterministic and thus fully analytically tractable and reproducible. Absolut! allows generating new validation sets suitable to test the validity of predicted patterns. Even under this simplified framework based on CDRH3 11-mers, an average of 6.8 million binding structures were explored for each 11-mer of each CDRH3 on an antigen, which allowed for identifying hundreds of high-affinity binding modes and several binding hotspots per antigen.

3D structure is intrinsically composed of long-range dependencies between distant AAs, since distant AAs in the antigen or antibody sequence may be neighbors in 3D space. Accordingly, we showed that the dependence between AA usage up to 5 to 11 AAs away between high-affinity binders and non-binders differ (Figure S6). By its coarse-grained representation of protein binding, Absolut! is not intended to precisely reproduce the exact location and structural geometries where one antigen binds, which is likely impossible in view of the complexity of antibody-antigen docking or binding prediction. Instead, we have shown that the Absolut! framework mirrors well immunogenic regions (Figure 2D) and physical rules behind antibody binding at high levels of complexity and is therefore relevant as a baseline ground truth dataset for biological problems in the immune receptor field, and it is useful to assess whether ML methods can predict binding within the many levels of complexity embedded into this framework. Absolut! is therefore a first decisive step towards generating parameterized custom-designed datasets for the benchmarking of ML methods for antibody-antigen-binding until either improved synthetic or suitably sized experimental datasets have become available.

Sequence-based classification of antigen-binding behavior can be improved by negative class design

Absolut!-generated datasets have allowed us to test the predictability of antibody-antigen binding as a function of dataset design, dataset size and sequence encoding. For binary classification, the definition of non-binder (negative class / negative samples) as top 1%–5% versus 1%–100% high-affinity sequences had a significant impact on prediction accuracy (Figure 3E,F), suggesting that the former strategy better delineates the boundary between binders and non-binders. Similarly in the context of enrichment-based antibody screening (*18*, *82*), where each enrichment step defines "better binders", our results indicate that the non-binders in a late enrichment step could represent a good negative class in comparison with the non-binders in the early enrichment steps. Therefore, Absolut! enables the generation of different negative samples in datasets with various controlled levels of difficulty, which is currently unfeasible experimentally. The sequence-based classification of binary antibody-antigen binding was successful with simple architectures and moderate data sizes (1000 to 5000 antibody sequences were sufficient to obtain 0.97 balanced accuracy using SN10, Figure 3D). In the context of multi-class classification, 200 000 sequences in total were sufficient to reach a macro-F1 accuracy of 0.67 with SN250 for predicting the binding to 140 antigens (Figure 3G). In all cases, DN networks did not provide a substantial advantage over SN architectures for classification tasks. In comparison, shallow learning methods reached 80% accuracy for binary (Figure S9A) and 0.39 macro-F1 for SVM with 140 antigens in multi-class classification (Figure S10). Furthermore, the VAE-generated latent space of monospecific binding sequences to different antigens did not cluster antigen-specific antibodies but rather by sequence similarity (Figure S11I,J), highlighting the complexity of binding rules that are not directly accessible by sequence similarity approaches. A next challenge would be to identify interpretable rules for predicting the target antigen of an antibody sequence. As a first attempt, we have shown that integrated gradients from trained SN models on D1 provide clustering of sequences binding to with the same paratope or epitope encoding, showing that structural rules are accessible from sequence datasets, although a preliminary knowledge of several modes of bindings might be required. Such rules would in turn allow for the generation of unlimited amounts of sequences binding to an antigen (of note, Absolut! allow to prospectively test the performance of predicted outputs from generative models (*72*), which is not possible with predefined datasets), and generating vaccine antigens with the desired immunogenicity or bound by desired antibodies. Assuming that such binding rules would be identified from an initial dataset (for instance a set of k-mers that would be predictive of binding to a specific antigen or epitope), Absolut! can then be used to generate new datasets suitable to test whether this rule is still valid on new datasets/in new conditions. Absolut! can thus prospectively test hypothetical binding rules inferred from small datasets. Assessing a model's applicability

domain and rigorous determination of model accuracy might arguably be more important for decision making than the particular modeling approach chosen (*83*). While there is no consensus on what an explainable ML should provide (*84*), several aspects of explainable AI that are desirable in drug design applications are met by the Absolut! framework due to its fully defined and deterministic simulations: (i) transparency, (ii) justification, i.e., elucidating why the answer provided by the model is acceptable, (iii) informativeness, i.e., providing new information to human decision-makers, and (iv) uncertainty estimation, i.e., the quantification of how reliable a prediction is (*83*).

Structural information improves sequence-based conformational paratope-epitope prediction

A major open question in the immune receptor ML field is to understand how beneficial it is to include structural information into the prediction task (*26*) and it has been challenging to combine sequence information from affinity enriched sequences, with structural binding datasets, which are intrinsically of different size and data type. For instance, structural prediction of paratope residues has allowed finding sequence-unrelated clonotypes binding the same target antigen (*53*).

Recent (linear) paratope or epitope prediction ML studies have shown that structural information does improve prediction accuracy. Pittala et al. predicted the (linear) paratope and epitope residues of antibody-antigen binding pairs whose binding structures are known, by adding an attention layer that includes information from the CNN of the other protein (*45*). The challenging part of sequence-based approaches is to convert an antibody sequence into potential structures (i.e., paratopes) to bind to an antigen, since the paratope structure is influenced by the antigen/epitope it binds likely due to CDRH3 flexibility/conformational sampling (*85–87*). Lu et al. (*46*) performed sequence-based paratope prediction by aligning antibody sequences with other antibody sequences whose antibody-antigen complex PDB structure was already known (not necessarily with the same antigen, n=460 total). The authors included the spatial arrangement of the paratope in these known complexes as multi-dimensional features for training a CNN followed by a geometric neural network. Schneider et al. (*67*) used Abodybuilder (*88*) to model the 3D structure of the antibody alone, using threading from known antibody structures, followed by a docking step to generate multiple binding poses and used an ML approach to provide a better ranking of the binding poses (as compared to the ranking provided by the docking scoring function). Interestingly, in both methods, including structures from alignment or threading to other known antibody-antigen binding structures was necessary to achieve high accuracy. Such antigen-agnostic prediction of the antibody structure before assessing the paratope/binding may bias the prediction to antibody-specific rules from the only crystallized antibody-antigen complexes instead of learning rules that well describe the structural compatibility between paratope-epitope pairs. Thus, the ideal antigen-aware and structure-equipped datasets to train models such as the one mentioned here can now be designed and generated in Absolut!.

Absolut! allows comparing the predictive potency of encodings with varying degrees of sequence and structural information prior to the availability of experimental data

Here, we present an approach to the paratope-epitope prediction problem, starting not from the antibody or the antigen sequence, but rather predicting the pairing between a paratope encoding and an epitope encoding, and using encodings that include a varying degree of structural information (Figure 4A). By using encodings that regroup multiple sequences under a shared pattern, this "encoded" approach allows embedding binding rules within the encoding, by keeping only relevant information describing the binding, and incorporating structural information on the structural "compatibility" between paratopes and epitopes. One epitope encoding value represents multiple epitopes on different antigens, and one paratope encoding value represents all the antibody sequences that bind with this paratope, possibly to

different epitopes. Of note, we kept the same level of encoding for the paratope and epitopes, and it would also be possible to test whether the epitope sequence can be predicted from a structural encoding, or vice-versa. We were able to delineate a hierarchy of, to a varying degree, predictive structural and sequence features into the paratope and epitope encodings (Figure 4B,C). The presence of the gaps in the encoding improved accuracy (compare "Sequence" and "Aggregate" encodings). Encoding AAs by their residue name or their chemical property did not significantly impact accuracy (compare "Aggregate" and "Chemical" encodings). Discarding binding residues of degree 1 in both the paratope and epitope provided a strong increase in prediction accuracy, and including the binding degree explicitly into the encoding further increased prediction accuracy. This suggests that the most informative embeddings /binding rules for antibody-antigen bindings should consider residues of binding degree 2 or more, the type of residue (chemical properties or their name), and the presence of gaps. In a previous study (*3*), we introduced the sequence, aggregate and motif encodings (without degree), as common rules for the binding of antibodies to different antigens, and shown that the sequence encoding provided low accuracies, while combining gaps with sequence (Aggregate) gave a higher prediction accuracy, on all non-redundant experimental antibody-antigen complexes known so far (AbDb database). Here, we validated these findings on the Absolut! dataset, and extended the predictions of how ML architectures would perform on bigger datasets, and including degree information. Therefore, using Absolut!, we assessed the predictiveness of interpretable features that are applicable to experimental antibody-antigen structures, as studied in (*3*).

In the context of the motif encoding, we found that shallow models could predict the epitope encoding (treated as a label as a multi-class task) to some extent, while the NLP architectures did not reach satisfying accuracies, likely due to the higher complexity of NLP text generation task, or the low dimension of encoded epitopes and paratopes (only two to five possible words, "X" and "-" for degree-free, or "X1", "X2", "X3", "X4" and "--" for degree-explicit), since NLP architectures have been developed on tasks with high number of words in the dictionary. However, for all the other encodings ("Sequence", "Aggregate" and "Chemical"), none of the shallow models reached more than 0.05 accuracy (Figure S16), showing that NLP architectures were more suitable to extract complex rules of antibody-antigen binding (Figure 4B,C). The encoder-architecture performed less well on encodings with degree 1 residues, likely because the encodings became too large with low numbers of possible words, leading to repeats due to loss of memory, and was outcompeted by the transformer in this condition. However, with degree 2, the encoded-decoder performed comparably to the transformer, although at a substantially higher computational cost as it uses recurrent neural networks (gated recurrent unit). Therefore, Absolut! allows comparing the efficiency of hybrid structure- and sequence-based encoding for their capacity to encode critical information that will be predictive for paratope-epitope prediction. The study of encodings allows identifying critical simple patterns that are most predictive, and therefore to find patterns that would have been missed by studying sequence similarity alone. Although we used manually-defined encodings (for the sake of straightforward interpretability), it would also be possible to use an embedding layer, either trained on millions of protein sequences such as the ESM1-b architecture (*89*), or a simple layer to generate internal representations of predictive patterns within smaller dimensions than the manual encodings. Knowledge of successful encodings, in turn, can help to design experimental datasets that are as complete as possible in the coverage of different values in such encodings, and avoiding generating biased or unbalanced data to only a few representations of antibody-antigen binding. It is reassuring that the best identified encodings here (in particular, the Aggregate encodings) have been validated to be also predictive of paratope-epitope matching on the set of experimental antibody-antigen structures (*3*).

Structural information in the encodings could potentially be extracted from sequence datasets in the future

The structural properties that have been used in the paratope and epitope encodings here (Figure 4A) are relatively simple (binding residues, degree of binding and gaps), and one may speculate that they may potentially be learned from sequence datasets, even without the knowledge of the antibody-antigen structures but with alternative types of datasets. For instance, the presence of gaps on the paratope could be observable on sequence datasets by assessing whether binding is impaired when the residues within the gaps are mutated (gap mutations should have less impact on the binding than binding positions, and one could expect a high entropy of binding sequences at these positions). If datasets where the binding antibody sequences to many single-point or double-point mutations antigens were available (using Deep mutational scanning like in (*32*)), a paratope residue of binding degree 1 would only be impacted by one point mutation on the antigen (although this mutation could also bind another paratope residue), while a paratope residue of binding degree 2 would be further impacted by a second mutation. Interestingly, the high accuracy of protein structure prediction by ML approaches (*56*, *88*) relied on evolutionary linked mutations between binding residues (stabilizing binding between two protein domains), given by multiple alignments of protein orthologs. Deep mutational scanning antibody- and antigen-binding datasets (*32*) resemble the above-mentioned evolutionary multiple sequence alignments and suggest that ML architectures could indeed leverage binding properties from it (*90*). The experimental identification of antibody sequence binding all single- and double-mutated antigens may not be available in the near future (*32*, *91*). However, Absolut! is able to generate many sequence datasets to mutated antigens *in silico*, and thereby allowing to predict what would be the smallest sequence dataset with a minimal number of mutated antigens to learn the paratope-epitope structural encoding from the sequence of the antibody and antigen only (of note, the structural encoding will always depend on the "compatible" paratope and epitope and this task cannot be learned separately for paratope alone, but always in the context of an antigen).

The challenge to find rules and encodings for high neoepitope prediction accuracy

Another major question in antibody discovery is to what extent rules learned from a set of antibodies binding to a set of antigens are applicable to other antibodies or antigens (generalization). For instance, antibody structure or paratope prediction studies typically allow similar paratope or epitope sequences to be in both the training and testing dataset, and therefore it is hard to disentangle whether their high prediction accuracy is due to the presence of a homologous antibody structure in the training dataset. In the context of paratope-epitope prediction, we have shown that several encodings could provide high accuracy when predicting the target epitope of new paratopes (Figure 4B,C). Since the dataset was built with paratope-epitope encoded pairs from 159 antigens to all the top 1% binder 11-mers, it means that antibody binding to different antigens and epitopes were governed by learnable shared patterns, and suggests the existence of universal rules for binding to all antigens. However, the high accuracy was due to the successful prediction of epitope targets that were already described in the training dataset (Figure S15, Figure S12B,C) while the accuracy of unknown epitopes during training (here termed neo-epitopes) was very poor (accuracy close to 0 and LD similar to shuffled controls). Certain encodings provided a slight decrease in LD between predicted and target neoepitopes suggesting that a sub-string or subpattern could be predicted for some neo-epitopes (Figure S15). The LD was slightly better for neoepitopes sharing subpatterns with training epitopes (close neoepitopes) than those more 'far away' from the training dataset (Figure S17). It remains to be tested whether improved LD to close neoepitopes would still be true after shuffling the neoepitope output values (i.e. non-specific LD improvement by chance), or whether specific binding rules could be predicted for these close neoepitopes, and which encodings better represented these shared rules between known and unknown epitopes. The Absolut! dataset contained a restricted

epitope space in comparison to the paratope space (due to the 159 antigens and the clustering of epitopes in immunogenic regions). The epitope landscape could be enlarged by using a more lenient affinity cut-off or by generating new bindings with perturbed antigens (by way of AA mutation or geometric perturbation for instance). Hyperparameter optimization of the two considered NLP architectures could further improve neoepitope prediction. The prediction of neoepitopes has also been proven to be difficult on the TCR binding prediction problem, for TCR sequences distant from the training dataset (*92*, *93*). Here, we went beyond the binary prediction of paratope and epitope residues, by showing that learning which epitope is compatible with which paratope is predictable, including the prediction of its structural features such as degree and gaps. Therefore, Absolut! also fosters the development of new methods before the data are available and enables the comparison of the predictive power of datasets with a varying richness of embedded structural information (in the paratope and epitope encoding). Since each paratope had very limited numbers of matching epitopes (most of the time only one epitope depending on the encoding, Figure S12A, grey histograms), we tested the simplest formulation of paratope-epitope prediction by learning paratope-epitope pairs independently, and the NLP architectures reached high accuracy probably by learning the most frequent epitope for each paratope. However, performing paratope-epitope prediction with datasets containing more antigens (mutated for instance) might require a more complex ML task formulation, since more answers will be possible for the same paratope (multiple epitopes for the same paratope, when a CDRH3 was binding different antigens with a different conformational epitope). Then, an accuracy of 100% will be impossible, and a formulation with predicting one of the possible epitopes as loss function, or using a Multiple Instance Learning formalism could help to learn from datasets with higher levels of cross-reactivity (*94*, *95*).

Protein-protein binding is different from antibody binding and Absolut! can help test transfer learning possibilities

Recent breakthroughs in ML-based protein structure prediction (*96*, *97*), conformational sampling (*98*) conformational trajectories (*96*), latent representation of protein structures (*99*), or protein-protein interaction (PPI) (*100*) might be transferred to antibody-antigen prediction in the future. One challenge will be the discrepancy between PPI binding and antibody-antigen binding. We have previously shown that the preferential AA usage of binding residue and the Ramachandran angles of the CDRH3 loops from antibody-antigen complexes are distinct from those observed in the binding interfaces of PPI datasets (*3*). PPI typically involves long-evolved binding structures, for which rigid docking is suitable, and for which large structural and evolutionary information is available (necessary for cutting edge protein structure predictions tools (*56*)), while antibody-antigen binding involves less-structurally determined antibody structure due to the flexibility of the CDR loops; the sampling of conformational ensembles (i.e., different paratopes for the same antibody when binding different antigens (*3*, *19*, *85*, *101–103*)), and the lack of evolutionary sequence data containing both antigen and antibody mutations to infer co-evolution of paratope-epitope binding AAs. A breakthrough would be achieved when PPI ML methods would be transferable to antibody-antigen binding, to combine both large PPI datasets and task-specific antibody-antigen binding datasets. We have shown that transfer learning from antigen-specific data-rich generative ML model ("high-N learning") to data-poor generative ML model ("low-N learning") trained on antibody sequence specific to a different antigen can enhance the quality of antibodies by the data-poor ML model (*72*). Absolut! is suitable to generate datasets mirroring those used in protein structure prediction or PPI prediction to test whether transfer learning would be achievable between PPI and antibody-antigen binding prediction. For instance, Absolut! could be used for generating multiple

sequence alignment datasets mirroring the evolution of antibody-antigen bindings, and predicting the minimal size or structure of such a dataset, allowing to optimize future experiments.

Limitations of the Absolut! framework in terms of reflecting the complexity of antibody-antigen binding

Although the Absolut! framework includes the above-mentioned levels of complexity, the main limitations of our model with respect to our current understanding of antibody-antigen binding are: (i) in contrast to the modeling performed in our framework, proteins do not conform to lattice representations based on fixed inter-AA distances and rigid 90-degree angles. By only accepting angles of 90 degrees, the lattice model strongly constrains the possible modes of bindings, to a necessary lower scale where an exhaustive search is feasible. (ii) Protein binding is a very complex process that depends on many additional factors not included in this model such as solvatation and 3D distribution of charges, and the possible flexibility of the antigen structure depending on the binding antibody. (iii) Only the CDRH3 loop (as opposed to VH-VL) was simulated, which is the most variable chain for antibody specificity (*3*, *4*), thereby neglecting the topological constraints with other CDR loops and their contribution to binding as well as the influence of the constant region on antibody binding (*104*). Indeed, we could show that Absolut! predicted binding structures were only partially overlapping with the experimental structural bindings (Figure S7B–E), and that the CDRH3 sequences preferred to bind concave areas of the antigen (Figure S7E). It is possible that peptides of the size of the CDRH3 alone would also bind in the same location, while the full antibody structure could be responsible for the differential binding.

Potential improvements for the Absolut! framework

Possible improvements of the Absolut! framework are: (i) within the lattice framework, it could be possible to force angle constraints within consecutive groups of three AAs (*105*), and thus, add an additional level of structural complexity on CDRH3 sequences binding an epitope: starting from a binding structure, a mutation could forbid certain angles used in that structure and constraint to bind with a different conformation. (ii) The 90 degrees angles can be refined to a larger set of angles. This will impact the lattice resolution and the antigen discretization might be redefined for higher lattice resolution. The angles between three consecutive AAs could be limited to specific angles. The AA coarse-grained resolution in Absolut! could be transformed into backbone carbons resolution, in which case library rotamers or Ramachandran regions of all AAs together could be used to restrict enumerated angles. A recent technique can enumerate all peptidic structures according to geometric constraints (*106*) and could be adapted. In this case, all possible binding structures would be enumerated from a list of possible "starting positions", and for each CDRH3 sequence, only the structures with Ramachandran-compatible angles would be kept and assessed for binding using a scoring function. (iii) Finally, since the framework structures between and after the CDRH3 are structurally conserved between antibodies, we could constrain the ends of a CDRH3 to have a specific distance, although we consider only sliding 11AAs among longer CDRH3s which means the geometric constraints would depend on the length of the CDR3, not only the 11-mer studied. Including structural constraints could enable simulating the CDRH3 and CDRL3 chains together, although the computational cost of evaluating each pair of compatible CDRH3 and CDRL3 structures without further optimizations seems unrealistic to achieve at the moment.

Absolut! can generate antibody repertoires with sequence-level antigen-specificity annotation

Although we have shown the usefulness of Absolut! on antibody-antigen binding prediction methods taking as input CDR3 sequences, the Absolut! framework also allows investigating the problem of antigen recognition at the repertoire level. Indeed, due to the low fraction of antibodies whose target has been identified, it remains a challenge to identify from antibody repertoire data predictive markers (CDR3

sequences or sequence patterns) that correlate with antigen recognition (*10*, *95*, *107*). The Absolut! framework can be used to generate synthetic antibody repertoires with sequence-level antigen-specific annotation, thereby enabling the investigation of ML methods for repertoire-based identification of immune-state associated antibody sequences (*10*, *95*, *108*, *109*). Absolut! may be used to investigate to what extent structural information may be important to consider for repertoire-based ML (*110*, *111*). Furthermore, these specificity-annotated datasets may be used for deeper investigations into modeling the ensemble specificity of antibody repertoires (*112*).

Conclusions

The advantage of the antibody-antigen simulation framework presented here is that it clarifies that the biological question of antibody-antigen prediction requires the formalization of the corresponding ML task as well as that it imparts the ability to create custom-designed datasets to train the ML models on. Indeed, we showed its immediate application to three different ML tasks all of which address the general problem of antibody-antigen binding prediction. Here, as proof of principle, we investigated to what extent training set size as well as structural information impact prediction accuracy. The presented workflow for data generation and ML benchmarking is also applicable to a large extent to TCR-epitope prediction (*69*, *113*, *114*). Naturally, the simulation and ML investigations here reported are far from exhaustive and will require follow-up studies. Specifically, beyond detailed benchmarking of already implemented encodings, investigations of graph-based encoding, physicochemical properties, full torsion angle, or 3D voxels, the addition of physics-based priors may be needed (*115*). Furthermore, comparative insights into different ML architectures such as Transformers, graph-based ML (*35*, *116*, *117*) as well as attribution strategies such as integrated gradients to map the models' prediction to the underlying biology are warranted (*115*). These follow-up investigations are especially important for addressing the transportability of trained ML architectures to other epitopes or antigens (*118*).

Materials and Methods

Representation of antibody-antigen binding

We used the Ymir framework that can represent binding between ligands and receptors in a 3D-lattice and has been used to model affinity maturation in germinal centers (*51*). Briefly, a protein AA chain is a set of consecutive (neighbor) AAs on the 3D lattice (Figure S2), meaning only 90 degrees angles are considered, and only one AA is allowed per lattice position. A protein chain is represented by its AA sequence, a starting point in space, and a list of relative moves in space to determine the next AA position (Straight (S), Up (U), Down (D), Left (L), Right (R)) according to a moving observer's coordinates (similar to a plane pilot) (*51*). From the starting point, the first move is defined from the coordinate system (Ox for straight, Oz for up, etc.), and the "observer" moves or turns. The next move is relative to the observer. "Backwards" (B) is only allowed as the first move (otherwise, it would collide with the previous position). For instance, "(25, 30, 28), BSLDUSUSRS, RGAYYGSSYGA" or "(26, 30, 27), DLURSLDDUL, YYGSSYGAMDY" are two protein structures of the size "10 moves" or 11 AAs. In particular, an antigen is a multi-chain Ymir protein (structure and AA sequence), and the antibody is represented either as an 11 AAs sub-sequence of its CDRH3 (sequence only) or as with a 3D conformation as an Ymir protein (structure and AA sequence).

Automatic discretization of antigens

The position of each AA within a protein's PDB structure of a protein (the antigen) should be transformed into 3D integer lattice positions in order to use the Ymir framework (Figure 1A). Intuitively, this process is similar to rescaling the PDB structure, finding the best rotation and translation, and converting each AA position into a possible integer position around it with the best fidelity to the original structure. We use the software LatFit (*48*) to perform this task, using its dRMSD (Root Mean Square Deviation, providing a measure of the distance between experimental and lattice positions) minimization algorithm. LatFit iterates through multiple possible reconstructions of the protein structure on the lattice. It starts from the first AA of the protein and assigns it to (0,0,0) in the lattice. It iterates through all possible neighboring positions for the next AA, thereby generating multiple starting lattice structures of size 2 AAs. Each structure is ranked according to its best fit to the underlying PDB positions, determined by the dRMSD, calculated on the best rotation/translation between this structure and the original PDB structure. LatFit iteratively keeps a list of N best-ranked structures of lattice positions for the first K AAs, and extends each structure with every possible position for the next AA, before ranking all the newly extended structures by their dRMSD and keeping only the N best ones. When residues are missing in the PDB file, or for instance when a protein loop was not structured enough to be reconstructed from a crystal or cryoEM, LatFit automatically starts a new chain starting with the next AA. Finally, since LatFit cannot parse residue insertions and can only process residues from a single chain in the PDB, we used pdb-tools (*47*) to rename AA insertions by shifting residue ID and merge all chains of interest into a single chain and adapted the LatFit C++ code to create new chains when necessary.

Altogether, the following LatFit parameters are considered during the discretization of the antigen:

i) *Type of discretization:* Which position do we consider for an AA (the position of the C Alpha "CA", the centroid center "CoM", or the fused center of positions of all the AA atoms "FuC" that we implemented);
ii) *Lattice resolution:* The distance between two lattice points in Å;
iii) *N:* The number of possible lattice chains that LatFit keeps at each iteration.

We compared the discretization types and lattice resolutions (Figure S3) and reached the conclusion that a lattice size of 5.25 Å using the "FuC" discretization is optimal to minimize the dRMSD quality of

discretization while avoiding an excessive number of empty cavities inside the protein when the lattice resolution is larger. As a tradeoff, values for N (retained lattice chain solutions) larger than 25 did not significantly improve the quality of discretization (dRMSD), although special cases of large proteins needed N=50 for LatFit to find a solution (PDB 3VI3 chains AB and 5JW4 chain A). Therefore, the discretization step preserves both the native antigen topology in 3D-space (typically <3.5Å, Figure S3) as well as the native surface AA distribution (Figure S7A).

Removal of aberrant geometries

Discretization artifacts might leave empty positions (cavities) inside the antigen, or generate topologies that would not be physically accessible by antibodies in vitro/vivo (Figure S3B). For instance, a free position surrounded by five AAs would be a dead-end if an antibody bound there. Furthermore, an antibody should not occupy a free position surrounded by four antigen AAs in the same plane (a "donut hole") because the antibody's tail would "pass through" the antigen. We used a recursive algorithm that identifies "aberrant geometries" and blocks access to all positions surrounded by four planar (donut) or five or six (dead-end) inaccessible positions until all such positions had been blocked. We added a visual inspection step for complicated topologies where the antibody may pass through the antigen (manual curation by adding blocking positions).

Assignment of glycans in the discretized antigens

Glycans show a high diversity of tree structures anchored on proteins as post-translational modifications and are of critical interest for antigen recognition because glycans on viral glycoproteins shield against antibody binding (*119*). However, the proteins used in PDB structures are often synthesized on expression systems that do not synthesize associated glycans, and crystal structures are more difficult to obtain in the presence of glycans (*120*). A glycan is typically a tree of sugar molecules anchored on a protein via a β-(1,4) linked N-acetylglucosamine (*NAG*) or N-acetylmuramic acid (*NAM*) stem. Here, we modeled the glycan by adding a "forbidden" position on the closest available lattice position to the stem position, around the discretized antigen (independently/after the LatFit discretization step was performed). The flexibility and weak binding of glycans (*121*) are therefore ignored. Starting from the PDB file, the following steps were carried out to assign glycan positions in the 3D lattice:

i) *identification of glycan stems:* Glycans are parsed from the PDB file by identifying NAM or NAG, and by finding which AAs are bound to them on the chains of interest. Since the NAM and NAG are relatively big (similar size to an AA), we decided to consider the barycenter position of all atoms of the NAM or NAG as one AA-like element and to discard the remaining sugar ramified structure that is normally flexible and do not necessarily preclude antibodies to be at the original positions of the ramifications.

ii) *identification of root AAs for the glycans in the discretized antigen:* Since the preprocessing of insertions by the tool pdb-tools shifts the AA IDs in the initial PDB, we keep track of the AA bound by each glycan stem by noting their position in space instead of their ID. After discretization, we retrieve where these AAs are (independently of their ID) positioned in space before and after discretization. The glycan position in the lattice should be attributed in the vicinity of this AA, and as close as possible to the real space position of the NAG or NAN barycenter.

iii) *definition of suitable positions to place glycans ("Atmosphere"):* In some cases, the discretization leaves empty space inside the antigen, which is not a suitable position for glycans that are typically located at the surface of proteins. We designed a recursive algorithm to identify "atmospheric positions" (Algorithm 1) in space that are defined as all positions that (i)

may be accessed from the exterior by 90-degree moves, and (ii) that are either of Hamming distance one (nearest neighbor) or distance two (non-nearest neighbor) from an antigen residue. This excludes buried empty lattice points. In the algorithm, [global] means the variable should be a global variable shared between all calls of the recursive search, to label previously visited positions.

Algorithm 1 Atmosphere: returns empty lattice points around (but not buried inside) an antigen.

```
procedure ATMOSPHERE(Set of antigen AA positions X)
    Step 1: Identify an antigen surface residue SP and an adjacent atmospheric point AP
    Pick randomly an antigen AA A_0(x_a, y_a, z_a) that is not on a lattice border ±maxX, ±maxY, ±maxZ.
    P ⟵ (x_a, y_a, maxZ)                                   ▷ A lattice border point
    while P ∉ X
        P ⟵ P − (0, 0, 1)
    end while
    SP ⟵ P                                                 ▷ First surface point
    AP ⟵ P + (0, 0, 1)                                     ▷ First atmospheric point
    Step 2: Recursive deep search of atmospheric points at distance 1 or 2 from the antigen (procedure below)
    Return RecursiveSearch(AP, Set of antigen AA positions X, ∅)
end procedure
procedure RECURSIVESEARCH(AP, Antigen AApositions X, [global] visited)
    S ⟵ ∅                                                  ▷ Empty set of postitions
    L ⟵ nearestNeighbors(AP)/V
    for each P_i ∈ L do
        visited ⟵ visited ∪ P_i   ▷ visited should be defined outside the function (global variable) such that
points visited by different sub-calls are not visited twice
        if non-nearestNeighbors(P_i) ∩X ≠ ∅) then
            S ⟵ S ∪ RecursiveSearch(P_i, X, visited)
        end if
    end for
    return S
end procedure
```

iv) *assignment of possible suitable free positions to each glycan:* to position a given glycan in the lattice, each glycan will be assigned a set of possible positions. Let A_i be the rooting antigen AAs for a glycan *i*, and its position (A_{ix}, A_{iy}, A_{iz}) in the lattice, from the discretization step. Let G_i be the original glycan floating position (NAG or NAN barycenter) and D_i the closest lattice integer position (rounding up all positions to integers). Among identified suitable positions (the "Atmosphere", the neighbors of A_i in the lattice are considered as potential positions for the glycan. If not, D_i and its neighbor lattice positions are considered. If there are still no such positions, its non-nearest neighbors are also considered (Algorithm 2, possible positions).

v) *greedy assignment of positions to glycans:* The possible positions for each glycan are sorted by their distance to the original position D_i and the glycans are iteratively assigned to a position in a greedy manner (Algorithm 2). For each glycan, its closest available position is assigned to it except if another glycan has only this choice (in which case the latter is assigned the respective position) or if another glycan has the same position with a smaller distance. The procedure is repeated until all glycans are assigned a lattice position. The greedy approach allows us to solve conflicts between spatially close glycans. Although it may be possible that there are not enough free positions for all glycans, in which case one position represents two or more glycans, we did not observe such a case for the antigens investigated.

Algorithm 2 Glycan assignment: for each glycan, the closest atmospheric point in the lattice is chosen, starting from glycans with smaller distances to atmospheric points.

```
Step1: Generation of suitable lattice positions for each glycan
procedure POSSIBLEPOSITIONS(Set of antigen AA positions X, glycan PDB positions G_{i∈[1..nG]}, position of AA A_i(a_ix, a_iy, a_iz) bound by Gi)
    Atm ⟵ Atmosphere(X)
    for i from 1 to n_G do
        L_i ⟵ ∅                                          ▷ List of possible positions for glycan i
        L_i ⟵ nearestNeighbors(a_i) ∩ Atm/X
        if L_i == ∅ then
            D_i ⟵ (round(G_ix), round(G_iy), round(G_iy))
            L_i ⟵ D_i ∪ nearestneighbors(D_i) ∩ Atm/X
            if L_i == ∅ then
                ListPositions ⟵ D_i ∪ non−nearestneighbors(D_i) ∩ Atm/X
            end if
        end if
    end for
    return couples (G_i, L_i)                            ▷ L_i is a set of lattice positions
end procedure

Step2: Greedy assignment of positions from the glycan with minimal distance
procedure ASSIGNPOSITIONS(couples (G_i, L_i), i ∈ [1..n_G])
    assignedPos ⟵ none, none, , ...                      ▷ Assigned lattice position for each glycan
    for i from 1 to n_G do
        Sort elements of L_i per distance to G_i (smallest first)
        while ∃i|assignedPos[i] == none                  ▷ i.e., not all glycans assigned yet
            for each glycan i with L_i ≠ ∅ do
                bestChoice ⟵ Li.head()
                if bestchoice ∉ ∪_{j≠i}{L_j.head().weight|L_j.head() <= bestChoice} then
                    assignedPos[i] ⟵ bestChoice
                    L_i ⟵ ∅
                else if bestChoice ∈ {L_{j≠i}|size(L_j) == 1) then
                    assignedPos[min({j})] ⟵ bestChoice   ▷ attributed to glycan with single choice
                    L_{min({j})} ⟵ ∅
                else
                    j ⟵ min index of j|L_j.head() == bestChoice
                    assignedPos[j] ⟵ bestChoice
                    L_j ⟵ ∅
                    Remove bestChoice from all other L_j
                end if
            end for
            if ∃i|assignedPos[i] == none and L_i == ∅ then
                Raise error: Glycan i could not get a position
                assignedPos[i] ⟵ impossible
            end if
        end while
    end for
    return assignedPos[]
end procedure
```

Calculation of CDRH3 binding energy for a binding structure

The affinity of an antibody (CDRH3) binding to an antigen depends on its binding confirmation (*51*). The antigen structure is pre-defined by the user (see Section above on discretization), and the structure of the antibody is not known a priori (the considered 11-mer AA subsequence). In the Ymir lattice framework (*51*), only neighboring but non-covalently bound AAs are considered interacting, and the strength of the interaction is defined by the Miyazawa-Jernigan (*80*) potential, an empirical, experimentally derived potential that gives a binding energy for each pair of interacting AAs (Figure S2A). For a particular conformation of the antibody on the antigen, by summing the binding energies of interacting AAs, three energies can be derived: i) the interactions between residues of the antigen (folding energy of the antigen), ii) the interaction between antibody residues (folding energy of the antibody) and iii) the interaction between residues of the antibody and the antigen (the binding energy). The total energy of the binding is the sum of the three interaction types. Since the antigen structure is assumed constant, its folding energy is constant and ignored. In order to find how an antibody CDRH3 sequence binds to the antigen, the antibody-antigen binding structure with minimum total energy (the most stable) will be defined as the "energetically optimal binding conformation of the CDRH3 to the antigen", and the binding energy will be recorded as "binding energy of this CDRH3 sequence to the antigen". It is this binding energy that is used for all downstream analyses. Of note, the most stable binding is not necessarily the one with the lowest binding energy, due to trade-offs between folding and binding energies.

Exhaustive search of the optimal binding (exhaustive docking)

The energetically optimal binding conformation is calculated using a previously developed algorithm (*51*) that performs an exhaustive search (Figure S2B). By enumerating all possible binding structures (which is feasible in the lattice), we have the guarantee to find the optimal binding, while usual docking approaches rely on optimization procedures which can always be stuck in local optima, and therefore called our approach "exhaustive docking". We decided to restrict the calculations to peptides of 11 AAs only as we could show that the lattice binding energy cannot directly be compared between binding sequences of different sizes (*51*) and that calculation for longer peptides was not computationally feasible. In brief, all possible 3D structures of 11-mers in the lattice are enumerated (independently of their AA content), provided they have a minimum predefined number of 11 contacts. Among the used antigen structures, an average of 6.8 million 11-mer structures was found per antigen (min 27 500, max 44.3 million) filtered according to a minimum of 11 contacts to the antigen (i.e., the sum of degrees of AAs on the antibody 11-mer). Allowing a lower number of contacts led to an increasingly higher number of structures, and we noted that found optimal structures always had many more contacts than 11 (22 on average), suggesting that this contact threshold could further be increased). Finally, when treating a CDRH3 sequence, all its consecutive 11-mers AA sequences (called "antibody slices") are derived and assessed for binding and folding energies according to all possible binding structures on the antigens. The optimal binding conformation of the 11-mer with the lowest binding energy is kept as the structural binding of the CDRH3. CDRH3s of less than 11 AAs are discarded (6 919 242 remaining out of 7 317 314, see next paragraph).

Antigen and CDRH3 data source for the generation of high-throughput structural antibody-antigen binding structures

Antigen dataset: In order to build the Absolut! database, we discretized 159 antigens sourced from the AbDb database, which collects all non-redundant PDB structures of antibody-antigens complexes, as a selection of therapeutically relevant antigens, as well as to allow for the comparison of our epitopes against experimentally known binding regions (Figure S7E). The list of antigens and their species is shown in (Table S1).

CDRH3 Sequence dataset: We used a total of 7 317 314 unique CDRH3 sequences of murine naive B cells (*49*). For comparison, we used 1 million healthy human sequences of naive B cell BCRs from a set of 22.6 million CDRH3 AA sequences (*52*). Although our pipeline is valid for any CDR/FR sequence, antibody binding was computed for a single binding site, the CDRH3 of the antibody heavy chain, because it is the most important antibody site for antigen specificity (*3*, *4*). Only the 6 919 242 CDRH3 of size 11 or more were used for binding calculations.

Absolut! computational consumption

The calculation of one binding energy/conformation of one 11-mer to one antigen takes on average one to two seconds on a CPU (depending on the number of structures in the exhaustive search), and 8 seconds per CDRH3 (Figure S4F). In total, 16 000 CPU hours were needed per antigen on average, and approximately 3 million CPU hours were required to generate the Absolut! database.

Clustering of antibody binding hotspots

From a structurally annotated list of CDRH3 sequences, i.e., the lattice binding structure and their associated epitope (set of bound antigen AA in the lattice), we define a binding hotspot as a set (cluster) of epitope residues shared by multiple binding structures on the antigen, in the following way (see Figure S5 for the definitions and Figure S18):

i) *Assign a list binding epitope residues to each CDRH3.* Each CDRH3 binding structure is translated into a list of positions bound on the antigen (the epitope). Additionally, to differentiate only "important" positions on the antigen, we first select residues on the CDRH3 that are a neighbor to at least *D* residues on the antigen, and then create the list of epitope residues A_i bound by these CDRH3 residues.

ii) *Separate sets of K antigen residues that explain the epitopes of all CDRH3s.* We aim to identify binding hotspots as antigen residues bound by many CDRH3s. For each antigen residue, we created the list of all CDRH3 sequences that contain A_i in their fingerprint. A set of K residues $\{A_i\}$ "covers" the CDRH3 sequences whose epitope contains all the A_i. The identification of binding hotspots translates into the following question: *"Can we find a minimal list of K-sets of antigen positions that cover all the CDRH3 sequences?"* This is a "set cover" problem, which is NP-complete (*122*). We decided to implement a greedy approximation solution by first taking the set of K positions that cover the most CDRH3s, then removing these CDRH3s and iteratively determining the set of K positions that cover the most remaining CDRH3s (Algorithm 1). Therefore, each set of K positions mirror the binding pattern of a group of CDRH3s.

iii) *Translate each K-set into a binding hotspot:* For each K-set, we analyze which residues are bound by this group of CDRH3s. The fraction of CDRH3 structures binding to each residue of the antigen is computed. The residues in the K-set are all bound by 100% of the CDRH3s. However, other positions might also be bound by 100% or a high enough percent and will be grouped together as an epitope.

Therefore, in our framework, *"a binding hotspot is a cluster of antigen residues bound by a group of **CDRH3**s with K common positions in their binding epitope"*. Furthermore, a CDRH3 will be named hotspot-specific if it contains the K common binding positions of this epitope. Note that a position can belong to two hotspots. It is not impossible that new CDRH3s would have structures overlapping two hotspots (although very unlikely), in which case they will be classified as binding to both hotspots.

Algorithm 3 Clustering of binding hotspots as sets of K positions on the antigen. The epitope of each binding i is taken as a set of N_i antigen positions binding antibody residues with minimal interaction degree D. All subsets of K positions of the N_i are generated. The K-sets that represent the most binding epitopes become binding hotspots.

```
procedure GETKSETS(Antigen AA positions X, bindings (CDR3_Seq R_i, 3D_structure S_i), K, D)

    Step 1: For each binding, select the list of interesting epitope residues, generates all K-subsets from them,
    and assign the K-subset back to the bindings
    SequencesPerKSet ⟵ empty dictionary of type {Set of K positions ⟶ list of CDR3 sequences}
    for each (R_i, S_i) do
        EpitopeResidues < −∅
        for each position P in S_i do
            if size(P.nearestNeighbours ∩ X) >= D then
                EpitopeResidues ⟵ EpitopeResidues ∪ (S.nearestNeighbours ∩ X)
            end if
        end for
        ListKSets ⟵ GenerateKSubsets(EpitopeResidues, K)                ▷ See function below
        for each set S in listKSets do                                  ▷ i.e., for each K-set
            SequencesPerKSet[S] ⟵ SequencesPerKSet[S] ∪ R_i             ▷ R_i is associated with all its K-set
        end for
    end for

    Step 2: Greedy selection of K-sets that explain most sequences
    bindingHotspots ⟵ emptylistofsets
    while SequencesPerKSet ≠ empty
        Sort the sets in SequencesPerKSet from highest number of sequences they contain
        Winner ⟵ SequencesPerKSet.head()
        bindingHotspots ⟵ bindingHotspots ∪ Winner
        for each set S in SequencesPerKSet do
            for each CDR3 sequence R ∈ SequencesPerKSet[Winner] do
                Remove R from SequencesPerKSet[S]
            end for
            if SequencesPerKSet[S] is empty then
                Remove S from SequencesPerKSet
            end if
        end for
    end while
    Return PoolWinners
end procedure

Tool function: generate all possible subsets of K elements from a set of N elements.
procedure GENERATEKSUBSETS({I_1 > 0, I_2 > 0, ...I_N > 0}, integer K)
    if K < 1 then
        return ∅                                                        ▷ End of recursion
    else if K == 1 then
        Return {I_1, I_2, ...I_N}
    else if size(S) < K then
        Return {−K + size(S), ..., −2, −1, I_1, I_2, ..., I_n}          ▷ Fills with negative numbers to reach K elements.
    else
        setList ⟵ empty list of sets
        for i from 1 to N do
            SubsetsWithoutIi ⟵ GenerateKSubsets(S/{Ii}, K − 1)
            for each subset G ∈ SubsetsWithoutIi do
                setList ⟵ Result + {Ii} ∪ G
            end for
        end for
        Return setList
    end if
end procedure
```

Feature extraction from annotated antibody-antigen binding structures

Definition of affinity classes (Figure 2B): Starting from 7 317 314 unique murine CDRH3 sequences (*49*), 6 919 242 CDRH3s of 11 or more AAs were kept and a total of 34 395 055 11-mers were generated (i.e., on average, 4.97 11-mers are generated per CDRH3). The same 11-mer can occur in different CDRH3 sequences, leading to 22 684 073 unique 11-mers. As explained above, the binding energy and structure of a CDRH3 to the considered antigen, are defined by its 11-mer with the lowest binding energy.

The Absolut! database allows generating datasets with antibody-antigen binding sequences represented as a CDRH3 or a 11-mer sequence. Binding thresholds and affinity classes were defined based on the CDRH3 sequences, for each antigen separately. Binder CDRH3 sequences were defined as top 1% binders and ranged between 67 761–81 255 unique CDRH3 sequences (median 70 024). This number varied between antigens, due to multiple CDRH3 sharing the same binding energy as the threshold energy. Other CDRH3 binding classes such as the top 1%–5% binders, or the bottom 95% non-binders were defined analogously (Figure 2B).

However, at the 11-mer level, top 1% binders were *not* defined as the top 1% of the 22 million unique 11-mers, because the binding energy of each CDRH3 was calculated among 4.97 different 11-mers on average, i.e., only keeping the energetically lowest 11-mer and discarding the other 3.97 11-mers. Therefore, an equivalent threshold would be around the top 0.2% best 11-mers. Instead of using a relative threshold proper to the 11-mers, we decided to define the "binder" or "top 1%" 11-mers based on the same absolute binding threshold as the top 1% CDRH3 sequences to the considered antigen, i.e, the list of 11-mers that would allow a CDRH3 to be a top 1% (CDRH3) binder to this antigen. We obtained between 28 486 and 97 618 unique 11-mers as top 1%" binders per antigen (median 53 753), as shown in Figure S8A. If the CDRH3 sequences did not share any 11-mer, we would have expected to obtain around 69 000 different binders for each antigen (exactly 1% of the number of CDRH3s). Lower numbers of top 1% 11-mers meant that the same 11-mer was isolated from multiple different CDRH3s, while higher numbers of top 1% 11-mers reveal a higher diversity of high-affinity 11-mers within the top 1% CDRH3s, i.e., that one CDRH3 contained more than one 11-mer over the binding threshold.

The strategy of defining affinity classes CDRH3 or 11-mers according to the same energy threshold allows interchangeably to use 11-mer or CDRH3 binders for an antigen to build ML datasets. Throughout the manuscript, we used the 11-mers derived from CDRH3 sequences, to work on sequences of equal size.

Binary classification of binders (Figure 3A–F)*:* The "top 1%" 11-mers were isolated for each antigen (median 53 753 unique sequences, as explained above), and 25 000 of them were taken and labeled as "binders". A random selection of 25 000 non-binders 11-mers (dataset 1, within the bottom 99% CDRH3 energies) or suboptimal binders (dataset 2, top 1%–5% CDRH3 binding energies) was taken. For both datasets, the non-binder 11-mers were sampled from CDRH3 of the same size distribution as the CDRH3 of origin of the binders, in order to avoid obvious AA composition bias, since CDRH3 sequences tend to start by "CAR" and shorter CDRH3s have, for instance, higher usage of C, A and R amino acids.

Multi-class classification of antigen target (Figure 3G–I)*:* Among the 159 antigens, several antigen structures originated from the same protein (but different PDBs, see antigens with "*" in Table S1). The binding structures of the CDRH3s to the remaining 142 non-redundant antigens were kept. When K antigens were randomly picked to perform classification, only the top 1% binders to at least one of these K antigens were kept (we discard 11-mers that bind none of them, and perform the classification among binders only). Further, since we opted for multi-class techniques, we discarded any 11-mer that bound more than one

antigen among the K-selected antigens, explaining the saturation in the number of available sequences as a function of the number of antigens in Figure 3H. In the future, multi-label techniques may also be tested without discarding these cross-reactive sequences.

Paratope-epitope prediction (Figure 4)*:* The structural annotation of all the binding 11-mers to the 159 antigens was transformed into epitope and paratope encoding, as described in Figure 4. For each paratope and epitope encoding, each encoded pair was kept only once.

Assessment of physiological relevance of binding structures

Surface residues in Figure S7A were identified using the module `findSurfaceResidues` in `Pymol ver. 2.1.0 (123)`. For Figure S7B, for each of the 159 antigen-antibody complexes that were used to generate the library of *in silico* antigens, the position of the antibody was kept unchanged during the discretization of the antigen using Latfit, therefore the lattice antigen is shown with the experimental structure of the antigen.

Machine learning

Data processing and training/test split: For the classification tasks (Figure 3A), 11-mer AA sequences were encoded as one-hot encoding (binary vector, dimension 220), as input features for the architectures shown in Figure 3B. Dataset 1 (binary classification) was generated with 40 000 unique binder 11-mers (Figure S8A), and the same amount of unique non-binders (defined differently in D1 and D2). The number of used training sequences is shown in the figure legends, while testing was performed on 40 000 sequences. Only in 9 antigens out of 159 antigens, 56 900 to 78 000 sequences (binder + non-binders) were available in total, and the training ratio was downscaled to 80% training and the remaining sequences for testing (see Figure S8A for the distribution of available binding 11-mers per antigen). The amount of available sequences for Dataset 3 is shown in Figure S8B. When at least 300 000 sequences were available, 100 000 sequences were used for testing and 200 000 for training. For the ML use case of 10 antigens or less, less than 300 000 sequences were available and 80% of them were used for training and 20% for testing. For the paratope-epitope prediction problem, a list of unique paratope-epitope encoding pairs was generated by taking the structural annotation of all 11-mers from the top 1% affinity class. Therefore, the same encoded paratope-epitope pair can represent the binding of multiple 11-mer sequences to different antigens and is only repeated once in the paratope-epitope dataset. The number of available unique paratope-epitope pairs is shown in Figure S12A. The number of used training pairs is shown in the figure legends (between 1 600 to 200 000 pairs) while 100 000 pairs not present in the training dataset were used for testing. For the "motif" encodings, only 2 262 to 9 065 pairs were available, and for the "Max" or "8000" condition, 80% of all available pairs were used for training and 20% for testing. The encoded paratope-epitope pairs are processed for NLP processing by cutting the paratope and epitope in words of size 1 (degree-free) or size 2 (degree explicit) encodings while gaps were considered as one word (see Figure S19A). Processed pairs were completed by adding a <start> and <end> word, tokenized word by word, and padding. Words of size 1 in degree explicit encodings led to lower accuracies and we restricted ourselves to words of size 2.

Encoder-decoder: We leveraged a model based on Neural Machine Translation (*65*) to "translate" an epitope to a paratope and vice versa at motif, sequence, and aggregate levels. The architecture comprises two components: encoder and decoder with gated recurrent units (see Figure S19). Within the decoder, a Bahdanau attention layer was used to capture relevant input-side information necessary for the prediction of the output. Utilizing the context vector, the decoder generates each paratope or epitope motif/sequence character by character. The dataset was split into 80% training and 20% test set. An

embedding layer was used to learn numerical representations of the input. The parameter combination: (i) embedding dimension (6) and (ii) number of units (hidden dimension) (512) was used to parameterize the models. Here, the embedding dimension is the length of the vector representing the input whereas the number of units is the number of cells in the GRU otherwise known as the length of the hidden dimension. The training procedure was carried out for 150 epochs with Adaptive Moment Estimation (Adam) optimizer (*124*) and was replicated ten times. 20 additional epochs were used where the loss function was letting the decoder predict the full output sentence instead of predicting the next word from the true previous word. The deep learning framework, `Tensorflow ver. 2.0` (*125*), was used to construct, train, and evaluate the models.

Transformer: A transformer architecture was used as a comparison and implemented in `Tensorflow ver. 2.0`. Specifically, 8 heads, an embedding dimension of 16, and feed-forward layers of dimension 512 were used, together with a dropout rate of 0.1. The training dataset was separated into 40 batches of equal size and 250 epochs were used.

Shallow learning models for classification (including paratope-epitope prediction as a classification problem): Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR) models were defined as shallow models. All shallow learning models were parameterized with their baseline (default) parameters (see Table S2). We note that the shallow learning models, namely LR and SVM, yielded comparable performance (Figure S10) in comparison to the single-layer neural network Figure 3) and the performance may further be improved by hyperparameter optimization. Prior to training, the data was randomly split into training and test datasets (80:20), each AA was encoded as a one-hot vector (of size 20), consequently, each sequence was encoded as a one-hot vector of size 220 (11x20). Following training, the test dataset was used to obtain the model's accuracy. The ML framework, `Scikit-learn ver. 0.21.3` (https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), was used to construct, train and evaluate the models.

Performance metrics: Accuracy is defined as $\frac{tp+tn}{tp+fn+tn+fp}$ where tp is true positive, tn is true negative, fp is false positive and fn is false negative.

Macro-F1 is defined as

$$2 \times (Precision_M \times Recall_M)/(Precision_M + Recall_M),$$

Micro-F1 is defined as

$$2 \times (Precision_\mu \times Recall_\mu)/(Precision_\mu + Recall_\mu)$$

where macro averaging M is the per-class average and micro averaging μ is the cumulative average as defined in (*62*). Specifically, macro averaging splits the multi-class problem (*k* classes) into a binary classification for each class (k_i), measures the performance metric for each class and finally averages the results from all classes (average of *k* metrics). In contrast, micro averaging treats the entire dataset as one aggregate and measures one single metric (instead of *k* metrics). Thus, micro-averaging is more sensitive to imbalance in the dataset since the class with a large number of samples dominates the average. Macro- and micro-averaging will return an identical metric when the number of samples in each class is identical (balanced dataset).

*Integrated Gradi*ents (IG) (*64*) were computed with PyTorch (*126*) on the SN10 architecture to identify the importance of input dimensions towards the binder / non-binder prediction. The SN10 model (Figure 3) was trained on 40 000 sequences from dataset 1 for antigen 1ADQ_A (i.e., containing 50% binders and 50% non-binders to antigen 1ADQ_A). IG were only calculated on binder sequences from the training dataset (positive class), to identify which positions were predicted to be important for binding, or to reveal clustering of sequences based on their predicted IG. For each sequence, linear interpolations of 100 points between the one-hot encoded input and a baseline were created (we used the 'zero' baseline, i.e., a zero vector with the same dimensionality as the encoded input). Gradients for each interpolation were computed with respect to the output of the neural network. The IGs are computed by averaging over all 100 gradients. Thus, each sequence is associated with a 11x20-dimensional IG matrix.

VAE: We used a standard VAE implementation (*127*) using fully connected layers, batch normalization between layers (for numerical stability and to allow for higher learning rates - we used the Adam optimizer with learning rate 1e-04), and a module for sampling from standard normal distribution. The architecture is described in Table S3. The encoder takes one-hot encoded 11-mers (220 input units) while the decoder contains 220 output units. We trained the model with 233 605 samples. The VAE optimizes Evidence Lower Bound (ELBO), which consists of reconstruction error and the Kullback-Leibler (KL) divergence to keep the posterior probability close to the latent prior. The reparameterization trick was used to allow backpropagation through non-deterministic nodes in the neural network. Specifically, our implementation uses negative log-likelihood for reconstruction loss and KL divergence as a closed formula, and was built using the PyTorch deep-learning framework. Regarding parameter beta, we tried the following (1.05, 1.1, 1.2, 1.5, 2.0). The purpose of parameter beta was to encourage learning disentangled representation where each dimension of the latent space (orthogonal directions) corresponds to a single variation factor in the input space (*128*). We noticed that while increasing the value of parameter beta, the reconstruction loss increased as well on average (resulting in lower reconstruction quality, also described by Dupont et al. (*129*)), and did not observe any changes in the clustering. We, therefore, used beta=1.0.

ESM-1b embedding layer: In order to leverage biochemical attributes pre-learned on 250 million protein sequences, we added to the VAE the pre-trained ESM-1b Transformer embedding architecture described in (*89*). Per-sequence latent space representations of 11-mer sequences were generated using the default parameters and methods (*89*).

UMAP visualization: The latent space of the VAEs was clustered using UMAP with the following parameters: $n_{neighbors}$=(2, 5, 10, 20, 50, 100, 200), min_dist=(0.0, 0.1, 0.25, 0.5, 0.8, 0.99), $n_{components}$=(2,3), metric=('euclidean','correlation').

Graphics

Plots were generated using the R package ggplot2 (*130*) as well as the python packages seaborn (*131*) and matplotlib (*132*). PLots were arranged using `Adobe Illustrator 2020` (Adobe Creative Cloud 5.2.1.441). Logo plots were made with R package ggseqlogo (*133*) and networks using Cytoscape 3.0.

Hardware used

Computations were performed on the Norwegian e-infrastructure for Research & Education (NIRD/FRAM; https://www.sigma2.no) and a local server.

Code and data availability

The Absolut! package is freely available on https://github.com/csi-greifflab/Absolut/ and the binding database is available on https://greifflab.org/Absolut/.

Acknowledgements

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), a Research Council of Norway FRIPRO project (#300740, 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 LMS and GKS). This work was carried out on Immunohub eInfrastructure funded by 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). We acknowledge Thérèse Malliavin (Institut Pasteur, Paris, France) for her comments and suggestions that helped in the analysis of the results.

Declaration of interests

E.M. declares holding shares in aiNET GmbH. V.G. declares advisory board positions in aiNET GmbH and Enpicom B.V. VG is a consultant for Roche/Genentech.

References

1. R.-M. Lu, Y.-C. Hwang, I.-J. Liu, C.-C. Lee, H.-Z. Tsai, H.-J. Li, H.-C. Wu, Development of therapeutic antibodies for the treatment of diseases. *J. Biomed. Sci.* 27, 1 (2020).

2. R. A. Elsner, M. J. Shlomchik, Germinal Center and Extrafollicular B Cell Responses in Vaccination, Immunity, and Autoimmunity. *Immunity.* 53, 1136–1150 (2020).

3. R. Akbar, P. A. Robert, M. Pavlović, J. R. Jeliazkov, A compact vocabulary of paratope-epitope interactions enables predictability of antibody-antigen binding. *Cell Reports.* 34, 108856 (2021).

4. J. L. Xu, M. M. Davis, Diversity in the CDR3 Region of VH Is Sufficient for Most Antibody Specificities. *Immunity.* 13, 37–45 (2000).

5. V. Kunik, S. Ashkenazi, Y. Ofran, Paratome: an online tool for systematic identification of antigen-binding regions in antibodies based on sequence or structure. *Nucleic Acids Res.* 40, W521–W524 (2012).

6. D. J. Barlow, M. S. Edwards, J. M. Thornton, Continuous and discontinuous protein antigenic determinants. *Nature.* 322, 747–748 (1986).

7. G. N. Sivalingam, A. J. Shepherd, An analysis of B-cell epitope discontinuity. *Mol. Immunol.* 51, 304–309 (2012).

8. R. A. Norman, F. Ambrosetti, A. M. J. J. Bonvin, L. J. Colwell, S. Kelm, S. Kumar, K. Krawczyk, Computational approaches to therapeutic antibody design: established methods and emerging trends. *Brief. Bioinform.* (2019), doi:10.1093/bib/bbz095.

9. 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).

10. 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.

11. L. Csepregi, R. A. Ehling, B. Wagner, S. T. Reddy, Immune Literacy: Reading, Writing, and Editing Adaptive Immunity. *iScience.* 23, 101519 (2020).

12. S. Fischman, Y. Ofran, Computational design of antibodies. *Curr. Opin. Struct. Biol.* 51 (2018), doi:10.1016/j.sbi.2018.04.007.

13. P. Sormanni, F. A. Aprile, M. Vendruscolo, Third generation antibody discovery methods: in silico rational design. *Chem. Soc. Rev.* 47, 9137–9157 (2018).

14. D. R. Burton, What Are the Most Powerful Immunogen Design Vaccine Strategies?: Reverse Vaccinology 2.0 Shows Great Promise. *Cold Spring Harb. Perspect. Biol.* 9 (2017), doi:10.1101/cshperspect.a030262.

15. S. Ferdous, A. C. R. Martin, AbDb: antibody structure database-a database of PDB-derived antibody structures. *Database* . **2018** (2018), doi:10.1093/database/bay040.

16. J. Dunbar, K. Krawczyk, J. Leem, T. Baker, A. Fuchs, G. Georges, J. Shi, C. M. Deane, SAbDab: the structural antibody database. *Nucleic Acids Res.* **42**, D1140–D1146 (2014).

17. H. Wardemann, C. E. Busse, Novel Approaches to Analyze Immunoglobulin Repertoires. *Trends Immunol.* **38**, 471–482 (2017).

18. A. H. Laustsen, V. Greiff, A. Karatt-Vellatt, S. Muyldermans, T. P. Jenkins, Animal Immunization, in Vitro Display Technologies, and Machine Learning for Antibody Discovery. *Trends Biotechnol.* (2021), doi:10.1016/j.tibtech.2021.03.003.

19. A. Kanyavuz, A. Marey-Jarossay, S. Lacroix-Desmazes, J. D. Dimitrov, Breaking the law: unconventional strategies for antibody diversification. *Nat. Rev. Immunol.* **19**, 355–368 (2019).

20. M. Pedotti, L. Simonelli, E. Livoti, L. Varani, Computational Docking of Antibody-Antigen Complexes, Opportunities and Pitfalls Illustrated by Influenza Hemagglutinin. *Int. J. Mol. Sci.* **12**, 226 (2011).

21. S. Daberdaku, C. Ferrari, Antibody interface prediction with 3D Zernike descriptors and SVM. *Bioinformatics*. **35** (2019), doi:10.1093/bioinformatics/bty918.

22. E. Liberis, P. Veličković, P. Sormanni, M. Vendruscolo, P. Liò, Parapred: antibody paratope prediction using convolutional and recurrent neural networks. *Bioinformatics*. **34**, 2944–2950 (2018).

23. R. R. Eguchi, N. Anand, C. A. Choe, P.-S. Huang, IG-VAE: Generative Modeling of Immunoglobulin Proteins by Direct 3D Coordinate Generation. *bioRxiv* (2020), p. 2020.08.07.242347.

24. M. C. Jespersen, S. Mahajan, B. Peters, M. Nielsen, P. Marcatili, Antibody Specific B-Cell Epitope Predictions: Leveraging Information From Antibody-Antigen Protein Complexes. *Front. Immunol.* **10** (2019), doi:10.3389/fimmu.2019.00298.

25. G. Liu, H. Zeng, J. Mueller, B. Carter, Z. Wang, J. Schilz, G. Horny, M. E. Birnbaum, S. Ewert, D. K. Gifford, Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics*. **36**, 2126–2133 (2020).

26. C. Marks, C. M. Deane, How repertoire data is changing antibody science. *J. Biol. Chem.*, jbc.REV120.010181 (2020).

27. S. Friedensohn, D. Neumeier, T. A. Khan, L. Csepregi, C. Parola, A. R. G. de Vries, L. Erlach, D. M. Mason, S. T. Reddy, Convergent selection in antibody repertoires is revealed by deep learning. *bioRxiv* (2020), p. 2020.02.25.965673.

28. D. R. Ripoll, S. Chaudhury, A. Wallqvist, Using the antibody-antigen binding interface to train image-based deep neural networks for antibody-epitope classification. *PLoS Comput. Biol.* **17** (2021), doi:10.1371/journal.pcbi.1008864.

29. J. A. Ruffolo, J. Sulam, J. J. Gray, Antibody structure prediction using interpretable deep learning. *bioRxiv* (2021), p. 2021.05.27.445982.

30. A. Del Vecchio, A. Deac, P. Liò, P. Veličković, Neural message passing for joint paratope-epitope prediction. *ArXiV* (2021), (available at http://arxiv.org/abs/2106.00757).

31. A. Deac, P. Veličković, P. Sormanni, Attentive cross-modal paratope prediction. *J Comput Biol.* **26** (2019), doi:10.1089/cmb.2018.0175.

32. 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* (2021), doi:10.1038/s41551-021-00699-9.

33. J. Xu, Distance-based protein folding powered by deep learning. *Proc. Natl. Acad. Sci. U. S. A.* **116**, 16856–16865 (2019).

34. M. AlQuraishi, End-to-End Differentiable Learning of Protein Structure. *Cell Syst.* **8**, 292–301.e3 (2019).

35. F. Sverrisson, J. Feydy, B. Correia, M. Bronstein, Fast end-to-end learning on protein surfaces. *bioRxiv* (2020), p. 2020.12.28.424589.

36. H. C. S. Chan, H. Shan, T. Dahoun, H. Vogel, S. Yuan, Advancing Drug Discovery via Artificial Intelligence. *Trends Pharmacol. Sci.* **40**, 801 (2019).

37. H. Narayanan, F. Dingfelder, A. Butté, N. Lorenzen, M. Sokolov, P. Arosio, Machine Learning for Biologics: Opportunities for Protein Engineering, Developability, and Formulation. *Trends Pharmacol. Sci.* (2021), doi:10.1016/j.tips.2020.12.004.

38. R. J. L. Townshend, R. Bedi, P. A. Suriana, R. O. Dror, End-to-End Learning on 3D Protein Structure for Interface Prediction. *arXiv* (2018), (available at http://arxiv.org/abs/1807.01297).

39. M. B. Swindells, C. T. Porter, M. Couch, J. Hurst, K. R. Abhinandan, J. H. Nielsen, G. Macindoe, J. Hetherington, A. C. R. Martin, abYsis: Integrated Antibody Sequence and Structure-Management, Analysis, and Prediction. *J. Mol. Biol.* **429**, 356–364 (2017).

40. S. Mahajan, R. Vita, D. Shackelford, J. Lane, V. Schulten, L. Zarebski, M. C. Jespersen, P. Marcatili, M. Nielsen, A. Sette, B. Peters, Epitope Specific Antibodies and T Cell Receptors in the Immune Epitope Database. *Front. Immunol.* **9**, 2688 (2018).

41. J. R. Abella, D. Antunes, C. Clementi, L. E. Kavraki, Large-scale structure-based prediction of stable peptide binding to Class I HLAs using random forests. *Front. Immunol.* **11** (2020), doi:10.3389/fimmu.2020.01583.

42. M. A. Rangel, A. Bedwell, E. Costanzi, S. Ricagno, J. Frydman, M. Vendruscolo, P. Sormanni, Fragment-based computational design of

antibodies targeting structured epitopes. *bioRxiv* (2021), p. 2021.03.02.433360.

43. Y. Kang, D. Leng, J. Guo, L. Pan, Sequence-based deep learning antibody design for in silico antibody affinity maturation. *arXiv* (2021), (available at http://arxiv.org/abs/2103.03724).

44. P. P. Olimpieri, A. Chailyan, A. Tramontano, P. Marcatili, Prediction of site-specific interactions in antibody-antigen complexes: the proABC method and server. *Bioinformatics*. 29, 2285–2291 (2013).

45. S. Pittala, C. Bailey-Kellogg, Learning Context-aware Structural Representations to Predict Antigen and Antibody Binding Interfaces. *bioRxiv* (2019), p. 658054.

46. S. Lu, Y. Li, F. Wang, X. Nan, S. Zhang, Leveraging Sequential and Spatial Neighbors Information by Using CNNs Linked With GCNs for Paratope Prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2021) (available at https://ieeexplore.ieee.org/document/9439924).

47. J. P. G. L. Rodrigues, J. M. C. Teixeira, M. Trellet, Alexandre M J, pdb-tools: a swiss army knife for molecular structures. *F1000Res.* 7, 1961 (2018).

48. M. Mann, R. Saunders, C. Smith, R. Backofen, C. M. Deane, Producing High-Accuracy Lattice Models from Protein Atomic Coordinates Including Side Chains. *Adv. Bioinformatics*. 2012 (2012), doi:10.1155/2012/148045.

49. 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).

50. S. Miyazawa, R. L. Jernigan, An empirical energy potential with a reference state for protein fold and sequence recognition. *Proteins*. 36 (1999) (available at https://pubmed.ncbi.nlm.nih.gov/10409829/).

51. P. A. Robert, M. Meyer-Hermann, A 3D structural affinity model for multi-epitope in silico germinal center simulations. *bioRxiv* (2020), p. 766535.

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

53. S. A. Robinson, M. I. J. Raybould, C. Schneider, W. K. Wong, C. Marks, C. M. Deane, Epitope profiling of coronavirus-binding antibodies using computational structural modelling. *bioRxiv* (2021), p. 2021.04.12.439478.

54. Composition and Antigenic Effects of Individual Glycan Sites of a Trimeric HIV-1 Envelope Glycoprotein. *Cell Rep.* 14, 2695–2706 (2016).

55. D. E. Pires, D. B. Ascher, mCSM-AB: a web server for predicting antibody-antigen affinity changes upon mutation with graph-based signatures. *Nucleic Acids Res.* 44 (2016), doi:10.1093/nar/gkw458.

56. F. Ju, J. Zhu, B. Shao, L. Kong, T.-Y. Liu, W.-M. Zheng, D. Bu, CopulaNet: Learning residue co-evolution directly from multiple sequence alignment for protein structure prediction. *bioRxiv* (2020), p. 2020.10.06.327585.

57. R. S. Rudicell, Y. D. Kwon, S.-Y. Ko, A. Pegu, M. K. Louder, I. S. Georgiev, X. Wu, J. Zhu, J. C. Boyington, X. Chen, W. Shi, Z.-Y. Yang, N. A. Doria-Rose, K. McKee, S. O'Dell, S. D. Schmidt, G.-Y. Chuang, A. Druz, C. Soto, Y. Yang, B. Zhang, T. Zhou, J.-P. Todd, K. E. Lloyd, J. Eudailey, K. E. Roberts, B. R. Donald, R. T. Bailer, J. Ledgerwood, NISC Comparative Sequencing Program, J. C. Mullikin, L. Shapiro, R. A. Koup, B. S. Graham, M. C. Nason, M. Connors, B. F. Haynes, S. S. Rao, M. Roederer, P. D. Kwong, J. R. Mascola, G. J. Nabel, Enhanced potency of a broadly neutralizing HIV-1 antibody in vitro improves protection against lentiviral infection in vivo. *J. Virol.* 88, 12669–12682 (2014).

58. W. K. Wong, S. A. Robinson, A. Bujotzek, G. Georges, A. P. Lewis, J. Shi, J. Snowden, B. Taddese, C. M. Deane, Ab-Ligity: identifying sequence-dissimilar antibodies that bind to the same epitope. *MAbs*. 13, 1873478 (2021).

59. M. Golcuk, A. Hacisuleyman, B. Erman, A. Yildiz, M. Gur, Binding mechanism of neutralizing Nanobodies targeting SARS-CoV-2 Spike Glycoprotein. *bioRxiv* (2021), p. 2021.04.23.441186.

60. D. Angeletti, J. S. Gibbs, M. Angel, I. Kosik, H. D. Hickman, G. M. Frank, S. R. Das, A. K. Wheatley, M. Prabhakaran, D. J. Leggat, A. B. McDermott, J. W. Yewdell, Defining B cell immunodominance to viruses. *Nat. Immunol.* 18 (2017), doi:10.1038/ni.3680.

61. D. Angeletti, J. W. Yewdell, Understanding and Manipulating Viral Immunity: Antibody Immunodominance Enters Center Stage. *Trends Immunol.* 39 (2018), doi:10.1016/j.it.2018.04.008.

62. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* 45, 427–437 (2009).

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. *bioRxiv* (2021), p. 2021.05.23.445346.

64. M. Sundararajan, A. Taly, Q. Yan, Axiomatic Attribution for Deep Networks. *arXiv [cs.LG]* (2017), (available at http://arxiv.org/abs/1703.01365).

65. M.-T. Luong, H. Pham, C. D. Manning, Effective Approaches to Attention-based Neural Machine Translation. *arXiv* (2015), (available at http://arxiv.org/abs/1508.04025).

66. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need. *arXiv* (2017), (available at http://arxiv.org/abs/1706.03762).

67. C. Schneider, A. Buchanan, B. Taddese, C. M. Deane, DLAB - Deep learning methods for structure-based virtual screening of antibodies. *bioRxiv* (2021), p. 2021.02.12.430941.

68. D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* (2014), (available at http://arxiv.org/abs/1409.0473).

69. M. Milighetti, J. Shawe-Taylor, B. Chain, Predicting T cell receptor antigen specificity from structural features derived from homology models of receptor-peptide-major histocompatibility complexes. *bioRxiv* (2021), p. 2021.05.19.444843.

70. P. Rawat, D. Sharma, A. Srivastava, V. Janakiraman, M. M. Gromiha, Exploring antibody repurposing for COVID-19: beyond presumed roles of therapeutic antibodies. *Sci. Rep.* 11, 10220 (2021).

71. Mapping Polyclonal Antibody Responses in Non-human Primates Vaccinated with HIV Env Trimer Subunit Vaccines. *Cell Rep.* 30, 3755–3765.e7 (2020).

72. 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. H. Haff, G. Klambauer, G. K. Sandve, V. Greiff, In silico proof of principle of machine learning-based antibody design at unconstrained scale. *bioRxiv* (2021), p. 2021.07.08.451480.

73. J. Glanville, S. Ives, J.-P. Bürckert, C. Pettus, K. R. Peñate, R. Hovde, N. Bayless, D. Shanghavi, K. Glanville, E. Calgua, S. Youssef, A general solution to broad-spectrum vaccine design for rapidly mutating viruses (2020), doi:10.21203/rs.3.rs-100459/v1.

74. P. A. Robert, A. L. Marschall, M. Meyer-Hermann, Induction of broadly neutralizing antibodies in Germinal Centre simulations. *Curr. Opin. Biotechnol.* 51 (2018), doi:10.1016/j.copbio.2018.01.006.

75. O. H. Schnaack, A. Nourmohammad, Optimal evolutionary decision-making to store immune memory. *eLife* (2021), doi:10.7554/eLife.61346.

76. A. Mayer, V. Balasubramanian, T. Mora, A. M. Walczak, How a well-adapted immune system is organized. *Proc. Natl. Acad. Sci. U. S. A.* 112, 5950–5955 (2015).

77. A. S. Perelson, G. F. Oster, Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination. *J. Theor. Biol.* 81 (1979), doi:10.1016/0022-5193(79)90275-3.

78. L. A. Mirny, V. I. Abkevich, E. I. Shakhnovich, How evolution makes proteins fold quickly. *Proc. Natl. Acad. Sci. U. S. A.* 95 (1998), doi:10.1073/pnas.95.9.4976.

79. Defining and Manipulating B Cell Immunodominance Hierarchies to Elicit Broadly Neutralizing Antibody Responses against Influenza Virus. *Cell Systems*. 11, 573–588.e9 (2020).

80. S. Miyazawa, R. L. Jernigan, Residue – Residue Potentials with a Favorable Contact Pair Term and an Unfavorable High Packing Density Term, for Simulation and Threading. *J. Mol. Biol.* 256, 623–644 (1996).

81. Y. Xiang, Z. Sang, L. Bitton, J. Xu, Y. Liu, D. Schneidman-Duhovny, Y. Shi, Integrative proteomics reveals exceptional diversity and versatility of mammalian humoral immunity (2020), doi:10.1101/2020.08.21.261917.

82. H. R. Hoogenboom, Selecting and screening recombinant antibody libraries. *Nat. Biotechnol.* 23, 1105–1116 (2005).

83. J. Jiménez-Luna, F. Grisoni, G. Schneider, Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*. 2, 573–584 (2020).

84. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 1–42 (2018).

85. M. L. Fernández-Quintero, N. D. Pomarici, B. A. Math, K. B. Kroell, F. Waibl, A. Bujotzek, G. Georges, K. R. Liedl, Antibodies exhibit multiple paratope states influencing V H-V L domain orientations. *Communications biology*. 3 (2020), doi:10.1038/s42003-020-01319-z.

86. M. L. Fernández-Quintero, K. B. Kroell, F. Hofer, J. R. Riccabona, K. R. Liedl, Mutation of Framework Residue H71 Results in Different Antibody Paratope States in Solution. *Front. Immunol.* 12, 630034 (2021).

87. F. Waibl, M. L. Fernández-Quintero, A. S. Kamenik, J. Kraml, F. Hofer, H. Kettenberger, G. Georges, K. R. Liedl, Conformational Ensembles of Antibodies Determine Their Hydrophobicity. *Biophys. J.* 120, 143–157 (2021).

88. J. Leem, J. Dunbar, G. Georges, J. Shi, C. M. Deane, ABodyBuilder: Automated antibody structure prediction with data-driven accuracy estimation. *MAbs*. 8 (2016), doi:10.1080/19420862.2016.1205773.

89. A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. Lawrence Zitnick, J. Ma, R. Fergus, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.* 118 (2021), doi:10.1073/pnas.2016239118.

90. N. J. Rollins, K. P. Brock, F. J. Poelwijk, M. A. Stiffler, N. P. Gauthier, C. Sander, D. S. Marks, Inferring protein 3D structure from deep mutation scans. *Nat. Genet.* 51, 1170–1176 (2019).

91. E. Engelhart, R. Lopez, R. Emerson, C. Lin, C. Shikany, D. Guion, M. Kelley, D. Younger, Massively Multiplexed Affinity Characterization of Therapeutic Antibodies Against SARS-CoV-2 Variants. *bioRxiv* (2021), p. 2021.04.27.440939.

92. 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. Lindestam 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).

93. P. Dash, A. J. Fiore-Gartland, T. Hertz, G. C. Wang, S. Sharma, A. Souquette, J. C. Crawford, E. Bridie 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).

94. J.-W. Sidhom, H. Benjamin Larman, D. M. Pardoll, A. S. Baras, DeepTCR is a deep learning framework for revealing sequence concepts within T-cell repertoires. *Nature Communications*. 12 (2021), , doi:10.1038/s41467-021-21879-w.

95. M. Widrich, B. Schäfl, H. Ramsauer, M. Pavlović, L. Gruber, M. Holzleitner, J. Brandstetter, G. K. Sandve, V. Greiff, S. Hochreiter, G. Klambauer, Modern Hopfield Networks and Attention for Immune Repertoire Classification. *arXiv [cs.LG]* (2020), (available at http://arxiv.org/abs/2007.13505).

96. A. Plante, D. M. Shore, G. Morra, G. Khelashvili, H. Weinstein, A Machine Learning Approach for the Discovery of Ligand-Specific Functional Mechanisms of GPCRs. *Molecules*. 24 (2019), doi:10.3390/molecules24112097.

97. J. Ingraham, A. Riesselman, C. Sander, D. Marks, Learning Protein Structure with a Differentiable Simulator. *ICLR* (2018) (available at https://openreview.net/pdf?id=Byg3y3C9Km).

98. F. Noé, S. Olsson, J. Köhler, H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*. 365 (2019), doi:10.1126/science.aaw1147.

99. D. Rosenbaum, M. Garnelo, M. Zielinski, C. Beattie, E. Clancy, A. Huber, P. Kohli, A. W. Senior, J. Jumper, C. Doersch, S. M. A. Eslami, O. Ronneberger, J. Adler, Inferring a Continuous Distribution of Atom Coordinates from Cryo-EM Images using VAEs. *arXiv* (2021), (available at http://arxiv.org/abs/2106.14108).

100. C. H. M. Rodrigues, Y. Myung, D. E. V. Pires, D. B. Ascher, mCSM-PPI2: predicting the effects of mutations on protein–protein interactions. *Nucleic Acids Res.* 47, W338–W344 (2019).

101. I. Sela-Culang, S. Alon, Y. Ofran, A Systematic Comparison of Free and Bound Antibodies Reveals Binding-Related Conformational Changes. *The Journal of Immunology*. 189, 4890–4899 (2012).

102. M. M. Al Qaraghuli, K. Kubiak-Ossowska, V. A. Ferro, P. A. Mulheran, Antibody-protein binding and conformational changes: identifying allosteric signalling pathways to engineer a better effector response. *Sci. Rep.* 10 (2020), doi:10.1038/s41598-020-70680-0.

103. M. L. Fernández-Quintero, J. R. Loeffler, J. Kraml, U. Kahler, A. S. Kamenik, K. R. Liedl, Characterizing the Diversity of the CDR-H3 Loop Conformational Ensembles in Relationship to Antibody Binding Properties. *Front. Immunol.* 9 (2019), doi:10.3389/fimmu.2018.03065.

104. A. Shaw, I. T. Hoffecker, I. Smyrlaki, J. Rosa, A. Grevys, D. Bratlie, I. Sandlie, T. E. Michaelsen, J. T. Andersen, B. Högberg, Binding to nanopatterned antigens is dominated by the spatial tolerance of antibodies. *Nat. Nanotechnol.* 14, 184–190 (2019).

105. (φ,ψ)2 Motifs: A Purely Conformation-Based Fine-Grained Enumeration of Protein Parts at the Two-Residue Level. *J. Mol. Biol*. 416, 78–93 (2012).

106. A. Cassioli, B. Bardiaux, G. Bouvier, A. Mucherino, R. Alves, L. Liberti, M. Nilges, C. Lavor, T. E. Malliavin, An algorithm to enumerate all possible protein conformations verifying a set of distance constraints. *BMC Bioinformatics*. 16 (2015), doi:10.1186/s12859-015-0451-1.

107. J.-W. Sidhom, H. B. Larman, D. M. Pardoll, A. S. Baras, Author Correction: DeepTCR is a deep learning framework for revealing sequence concepts within T-cell repertoires. *Nat. Commun.* 12, 2309 (2021).

108. 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*. 36, 3594–3596 (2020).

109. M. Pavlovic, L. Scheffer, K. Motwani, C. Kanduri, immuneML: an ecosystem for machine learning analysis of adaptive immune receptor repertoires. *bioRxiv* (2021) (available at https://www.biorxiv.org/content/10.1101/2021.03.08.433891v2.abstract).

110. M. I. J. Raybould, C. Marks, A. Kovaltsuk, A. P. Lewis, J. Shi, C. M. Deane, Public Baseline and shared response structures support the theory of antibody repertoire functional commonality. *PLoS Comput. Biol.* 17, e1008781 (2021).

111. 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.

112. V. Greiff, H. Redestig, J. Lück, N. Bruni, A. Valai, S. Hartmann, S. Rausch, J. Schuchhardt, M. Or-Guil, A minimal model of peptide binding predicts ensemble properties of serum antibodies. *BMC Genomics*. 13, 79 (2012).

113. B. Ogunjimi, K. Laukens, P. Meysman, Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. *in Bioinformatics* (2020) (available at https://academic.oup.com/bib/advance-article-abstract/doi/10.1093/bib/bbaa318/6042663).

114. X. Lin, J. T. George, N. P. Schafer, C. Clementi, J. N. Onuchic, Rapid Assessment of T-Cell Receptor Specificity of the Immune Repertoire. *bioRxiv* (2020) (available at https://www.biorxiv.org/content/10.1101/2020.04.06.028415v1.abstract).

115. W. Gao, S. P. Mahajan, J. Sulam, J. J. Gray, Deep Learning in Protein Structural Modeling and Design. *Patterns*. 1, 100142 (2020).

116. A. R. Jamasb, P. Lió, T. Blundell, Graphein-a Python Library for Geometric Deep Learning and Network Analysis on Protein Structures. *bioRxiv*

(2020) (available at https://www.biorxiv.org/content/10.1101/2020.07.15.204701v1.abstract).

117. P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M. M. Bronstein, B. E. Correia, Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods* (2019), , doi:10.1038/s41592-019-0666-6.

118. W. M. Kouw, M. Loog, An introduction to domain adaptation and transfer learning (2018) (available at http://dx.doi.org/).

119. S. Boyoglu-Barnum, G. B. Hutchinson, J. C. Boyington, S. M. Moin, R. A. Gillespie, Y. Tsybovsky, T. Stephens, J. R. Vaile, J. Lederhofer, K. S. Corbett, B. E. Fisher, H. M. Yassine, S. F. Andrews, M. C. Crank, A. B. McDermott, J. R. Mascola, B. S. Graham, M. Kanekiyo, Glycan repositioning of influenza hemagglutinin stem facilitates the elicitation of protective cross-group antibody responses. *Nat. Commun.* 11, 1–12 (2020).

120. A. B. Ward, I. A. Wilson, The HIV-1 envelope glycoprotein structure: nailing down a moving target. *Immunol. Rev.* 275, 21–32 (2017).

121. R. Andrabi, C.-Y. Su, C.-H. Liang, S. S. Shivatare, B. Briney, J. E. Voss, S. K. Nawazi, C.-Y. Wu, C.-H. Wong, D. R. Burton, Glycans function as anchors for antibodies and help drive HIV broadly neutralizing antibody development. *Immunity*. 47, 524 (2017).

122. R. M. Karp, Reducibility among Combinatorial Problems. *Complexity of Computer Computations*, 85–103 (1972).

123. The PyMOL Molecular Graphics System, Version 1.8 Schrödinger, LLC, (available at http://www.sciepub.com/reference/159710).

124. D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. *arXiv* (2014), (available at http://arxiv.org/abs/1412.6980).

125. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A system for large-scale machine learning (2016) (available at https://research.google/pubs/pub45381/).

126. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv* (2019), (available at http://arxiv.org/abs/1912.01703).

127. M. W. D. P. Kingma, An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning* (2019) (available at https://ieeexplore.ieee.org/document/9051780).

128. I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework (2016) (available at https://openreview.net/pdf?id=Sy2fzU9gl).

129. E. Dupont, Learning Disentangled Joint Continuous and Discrete Representations. *Adv. Neural Inf. Process. Syst.* 31 (2018) (available at https://proceedings.neurips.cc/paper/2018/file/b9228e0962a78b84f3d5d92f4faa000b-Paper.pdf).

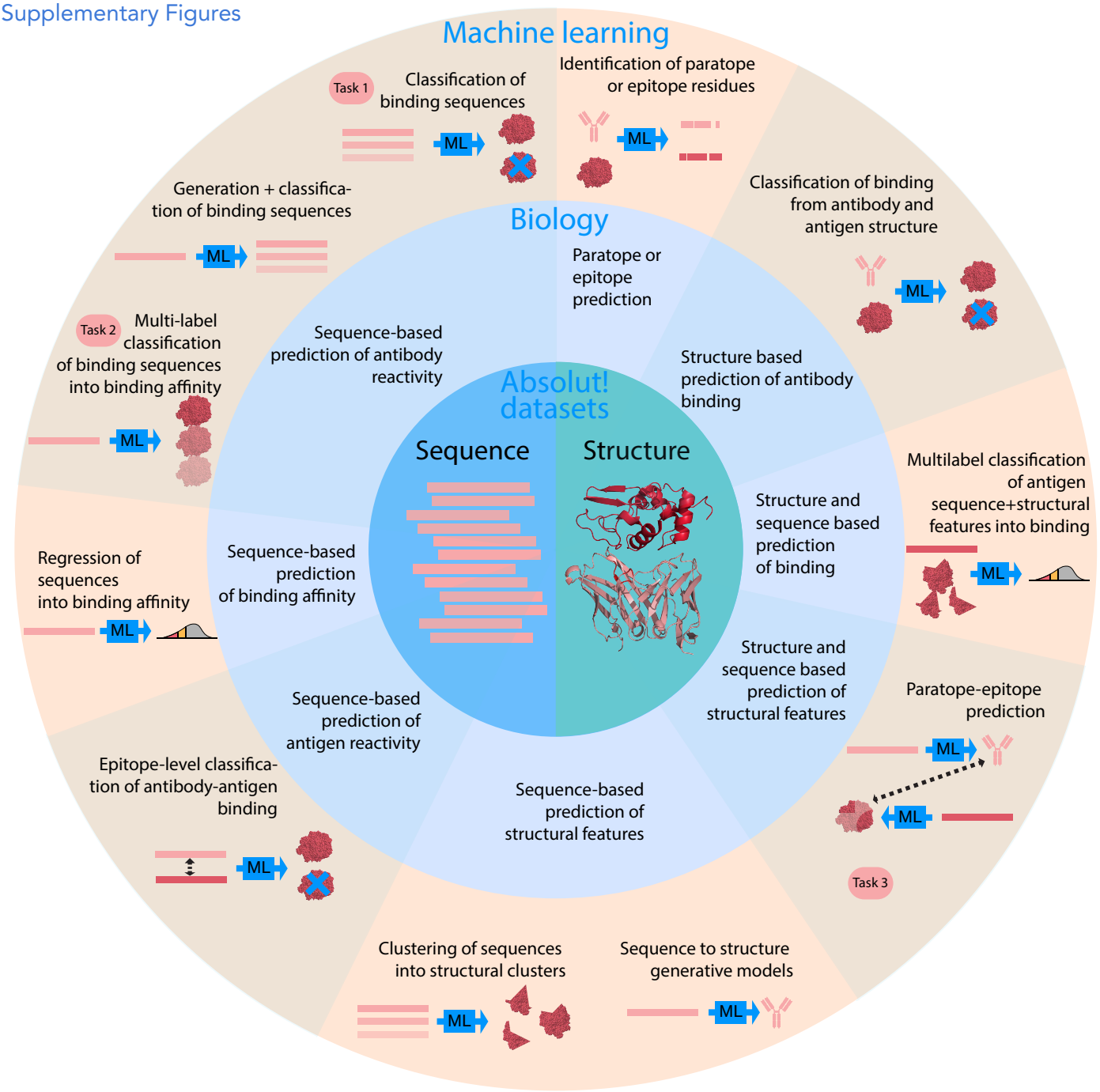
130. H. Wickham, *ggplot2: Elegant Graphics for Data Analysis* (Springer-Verlag New York, 2009).

131. M. Waskom, seaborn: statistical data visualization. *J. Open Source Softw.* 6, 3021 (2021).

132. J. D. Hunter, Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*. 9, 90–95 (2007).

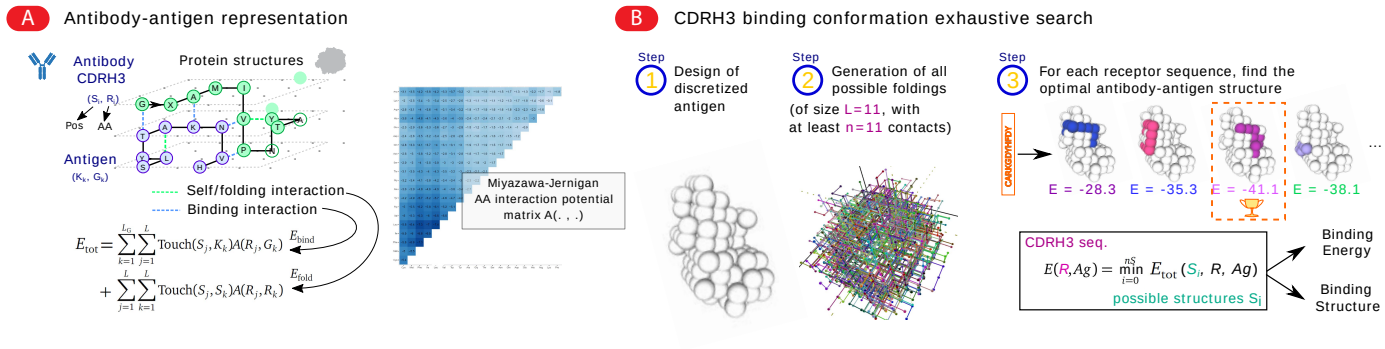
133. O. Wagih, ggseqlogo: a versatile R package for drawing sequence logos. *Bioinformatics*. 33 (2017), doi:10.1093/bioinformatics/btx469.

Supplementary Figures



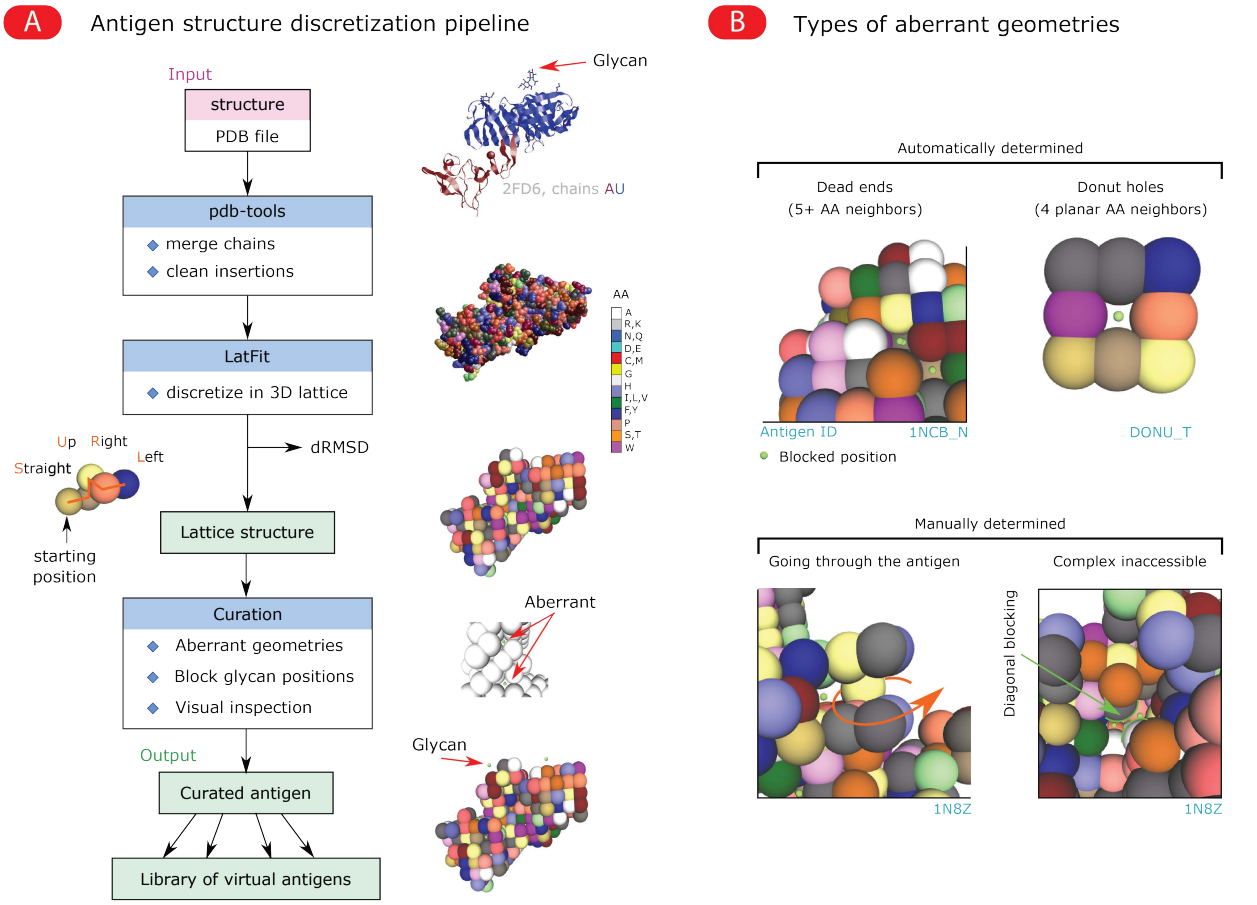
Supplementary Figure 1 (refers to Figure 1A) | Immunological antibody specificity prediction problems formalized as machine-learning tasks for which Absolut! can generate synthetic benchmark datasets.

Biological problems of antibody-antigen binding are very diverse and need translation into ML tasks performed on a specific type of experimental dataset. Datasets branch into two main categories: sequence- and structure-based (central ring). A large proportion of biological questions (middle ring) such as the prediction of antibody reactivity (to a defined antigen), binding affinity, antigen reactivity (to a defined antibody), and prediction of antibody-antigen structural features may be addressed (partially) by relying on sequence data alone, however, structural data contains the highest amount of information in terms of resolution and fidelity for the prediction of antibody-antigen binding (antibody and antigen structures are depicted in salmon and red respectively). Each biological problem can be converted into ML problems (outer ring) with a diverse panel of formulations, such as classification/generation of binding sequences, clustering of sequences into binding groups, multi-label classification of sequences into binding affinity class, prediction of structural features, and classification of binding from antibody and antigen structure. Most formulations typically fall into classification, regression, or paratope-epitope prediction (Figure 1A), and Absolut! is suited to generate *in silico* datasets for all of them.

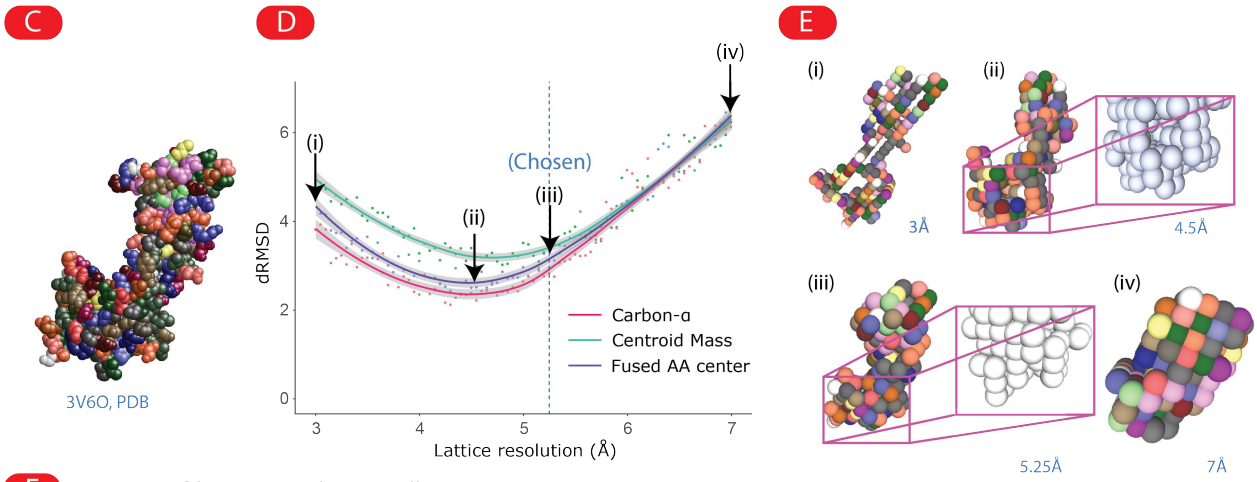


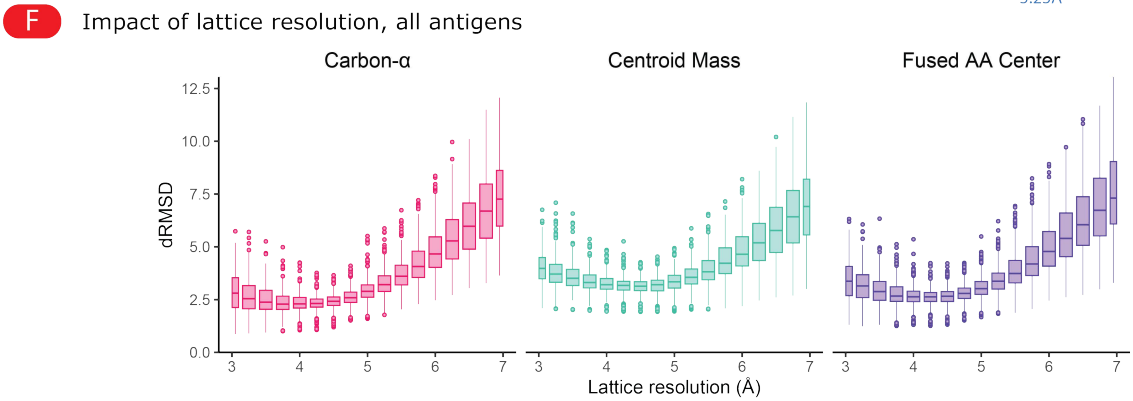
Supplementary Figure 2 (refers to Figure 1B) | Representation of antibody-antigen binding and deterministic calculation of the binding structure.

(A) Synthetic antibody-antigen binding structures are represented as lattice peptides whose neighboring AAs interact (non-covalently) via the Miyazawa-Jernigan interaction potential (*50*). The stability of an antibody-antigen complex is assessed by its total energy (E_{tot}) that sums two types of energies are defined from the interaction potential: the binding energy (E_{bind}), summing the energy of bonds between the two proteins, and the folding energies (E_{fold}), summing the energy of bonds within the antibody. The folding energy of the antigen is neglected since its structure is fixed and this energy is therefore constant. (B) Exhaustive docking: The energetically optimal binding conformation of an antibody chain sequence to an antigen is found by exhaustive search to minimize the energy of the binding complex. More details can be found in the Methods section.

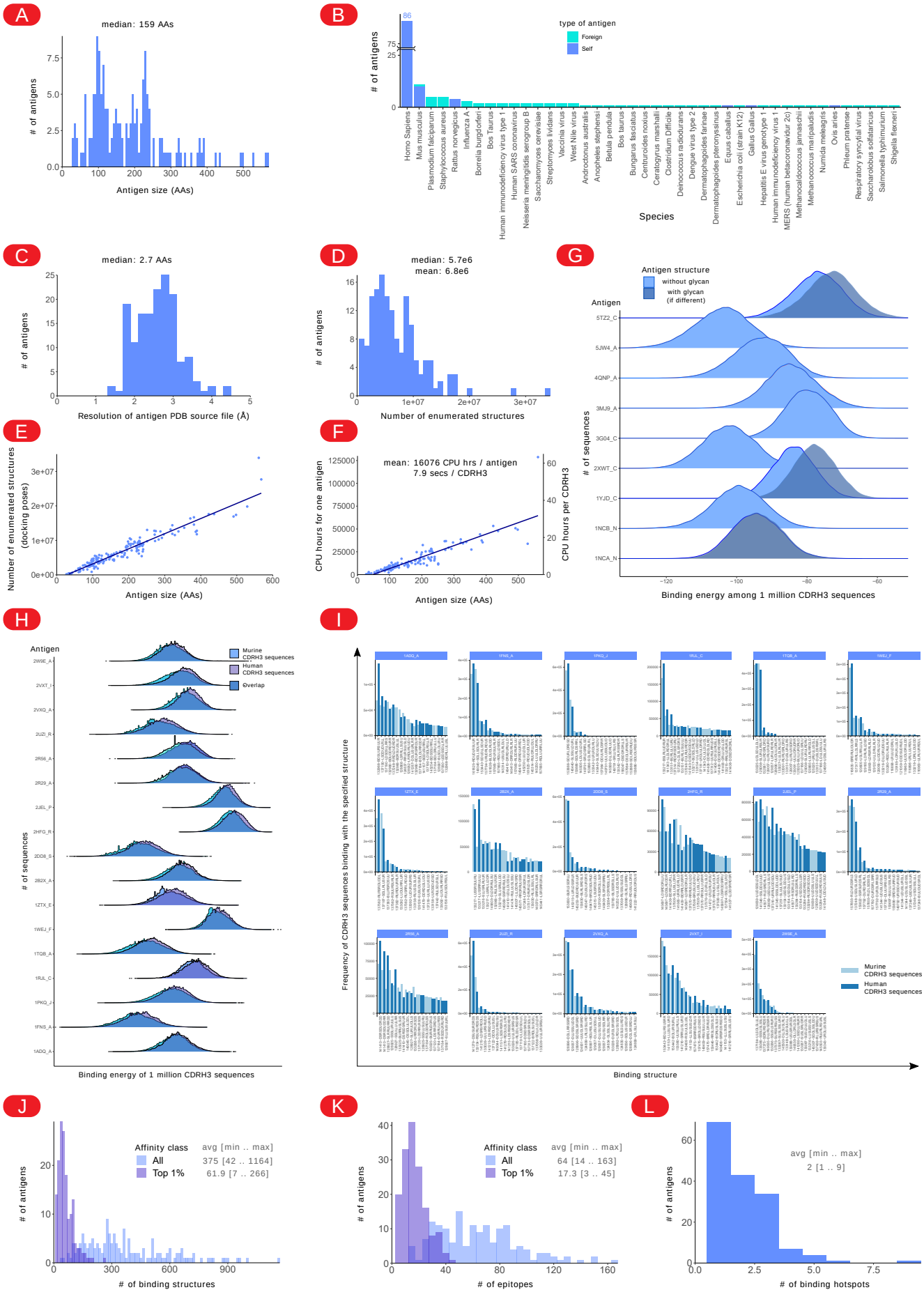


Impact of lattice resolution, one antigen



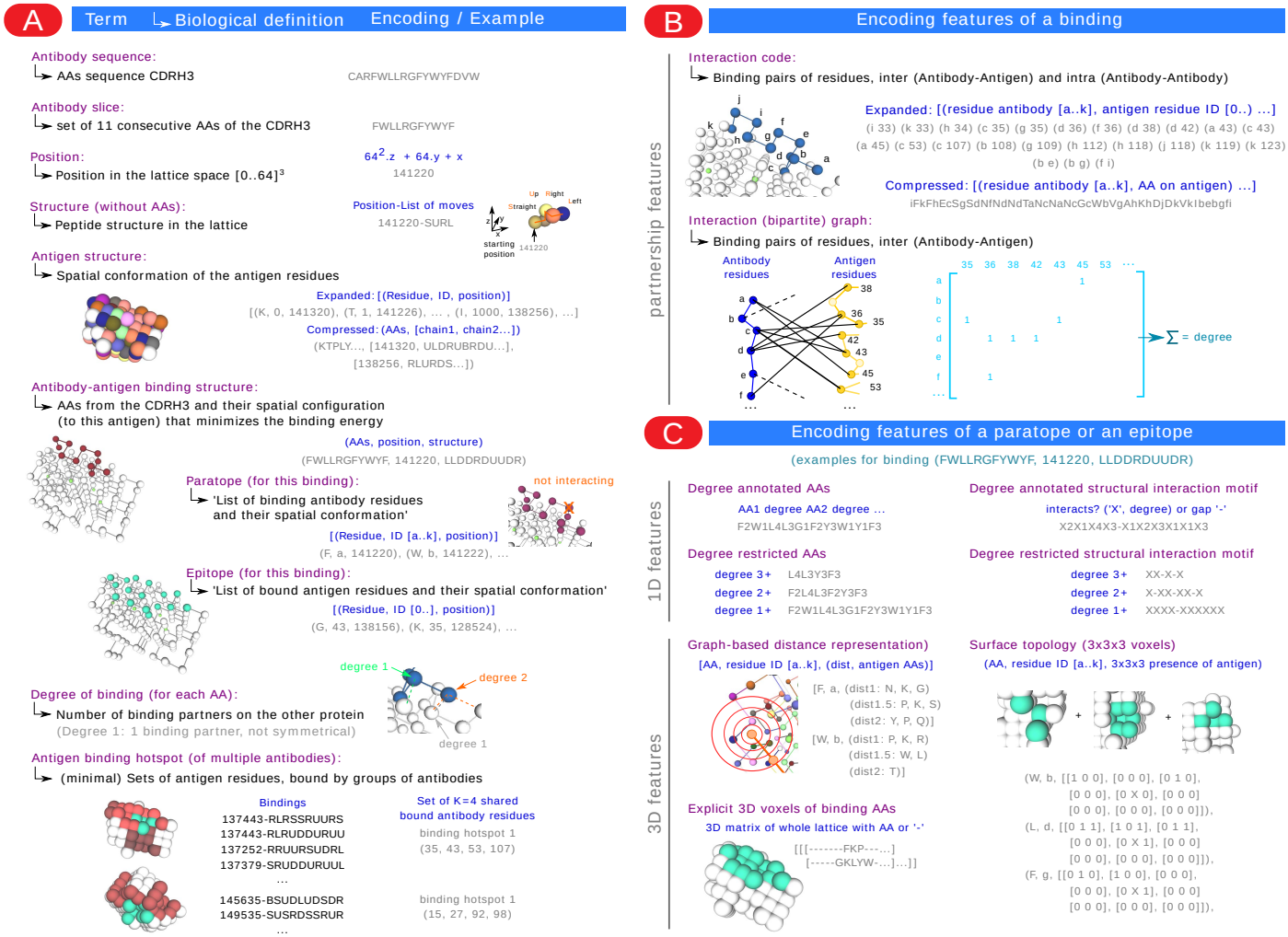


Supplementary Figure 3 (refers to Figure 1B) | Steps of the antigen discretization pipeline and optimization of the lattice resolution. (A) Antigen discretization pipeline. Antigens are filtered for the chains of interest and removal of insertions by pdb-tools (*47*) and are further discretized into a 3D lattice using LatFit (*48*), which outputs a new PDB file with both lattice and real-world AA positions side-by-side. These positions are transformed into a lattice structure (see Methods), where each chain is defined by a starting position and a string of letters representing moves in the lattice (Straight, Up, Down, Left, Right). The presence of glycans on the protein is parsed from the original PDB and added at the vicinity of the designated residues that are closest to the original position (see Methods). Finally, aberrant geometries are removed (curated antigen, see Methods). (B) Definition of "aberrant geometries" that are blocked from antibody access during discretization. Dead-ends and donut-shaped holes are automatically detected and blocked recursively. Subsequently, more complex holes are manually determined by visual inspection, such as larger-than-donut holes that would allow the antibody to go through the antigen, or more complex pockets where the antibody loop would be "diagonally" blocked to enter. (C–F) Choice of the best lattice resolution for discretization. (C–E) Discretization of the antigen of PDB entry 3V6O with different lattice resolutions (distance between two lattice points), from 3Å to 7Å. (C) Original PDB structure of antigen 3V6O with AA coloring (shapely palette). (D) Quality of the discretization of 3V6O depending on the lattice resolution, measured as the dRMSD, quantifying the average distance between pairs of AAs from the original PDB and their lattice position. The color represents the method of discretization: using the carbon alpha, the mass center of the centroid, or the fused center of the whole AA. (E) Result of discretization using the four lattice resolutions depicted in panel (D) using the fused center method, and showing the zoom on a concave area. (F) Discretization efficacy (dRMSD) of 130 antigens of the database, as a function of the lattice resolution and discretization method. In the manuscript, we opted for 5.25Å resolution (arrow C) in panel D, using a fused AA center' method.

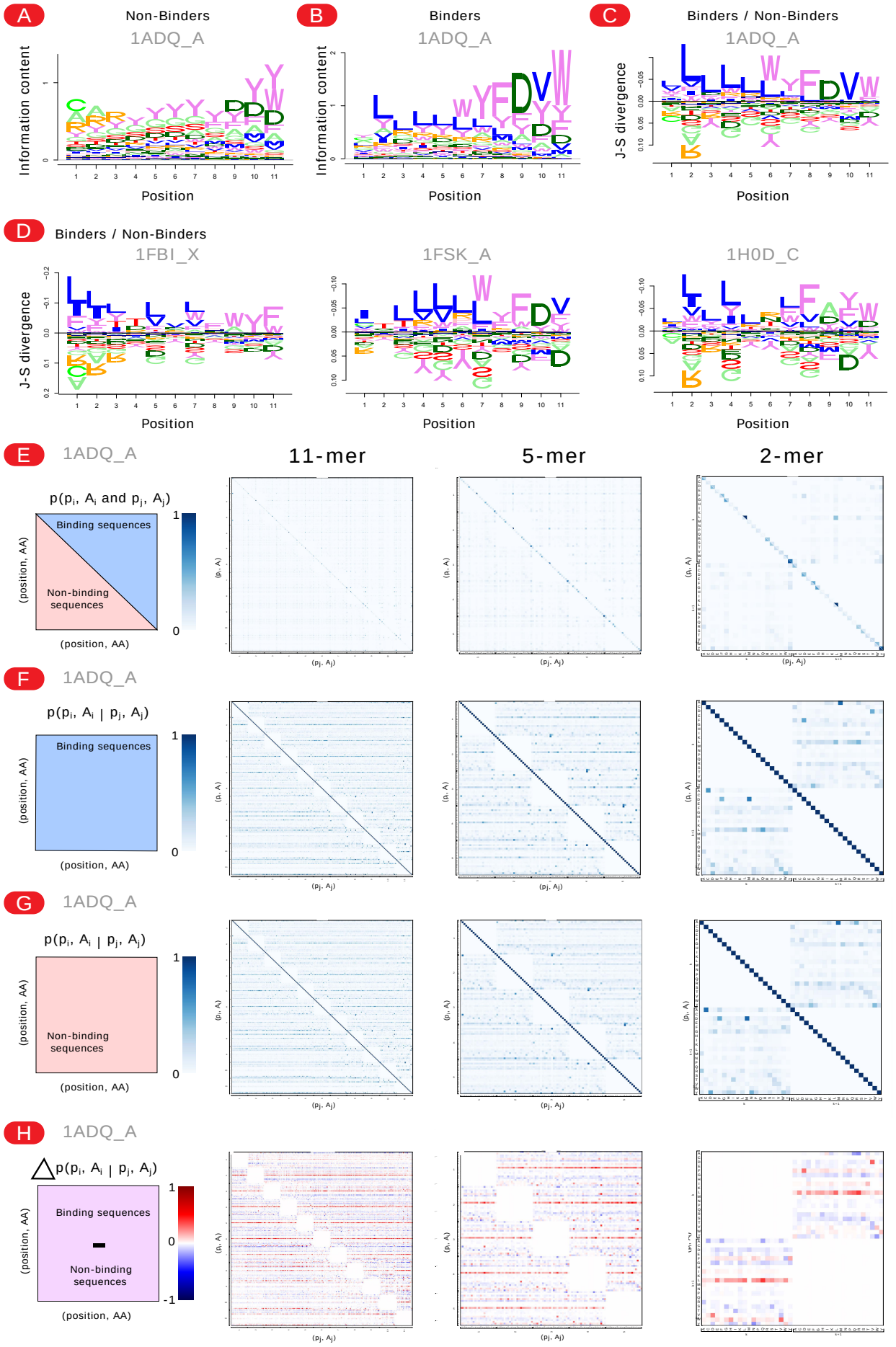

A
median: 159 AAs
# of antigens
Antigen size (AAs)
B
type of antigen
Foreign
Self
# of antigens
Species
Homo Sapiens
Mus musculus
Plasmodium falciparum
Staphylococcus aureus
Rattus norvegicus
Influenza A
Borrelia burgdorferi
Bos Taurus
Human immunodeficiency virus type 1
Human SARS coronavirus
Neisseria meningitidis serogroup B
Saccharomyces cerevisiae
Streptomyces lividans
Vaccinia virus
West Nile virus
Androctonus australis
Anopheles stephensi
Betula pendula
Bos taurus
Bungarus fasciatus
Centruroides noxius
Ceratogyrus marshalli
Clostridium Difficile
Deinococcus radiodurans
Dengue virus type 2
Dermatophagoides farinae
Dermatophagoides pteronyssinus
Equus caballus
Escherichia coli (strain K12)
Gallus Gallus
Hepatitis E virus genotype 1
Human immunodeficiency virus 1
MERS (human betacoronavirus 2c)
Methanocaldococcus jannaschii
Methanococcus maripaludis
Numida meleagris
Ovis aries
Phleum pratense
Respiratory syncytial virus
Saccharolobus solfataricus
Salmonella typhimurium
Shigella flexneri
C
median: 2.7 AAs
# of antigens
Resolution of antigen PDB source file (Å)
D
median: 5.7e6
mean: 6.8e6
# of antigens
Number of enumerated structures
G
Antigen structure
without glycan
with glycan (if different)
Antigen
5TZ2_C
5JW4_A
4QNP_A
3MJ9_A
3G04_C
2XWT_C
1YJD_C
1NCB_N
1NCA_N
# of sequences
Binding energy among 1 million CDRH3 sequences
E
Number of enumerated structures (docking poses)
Antigen size (AAs)
F
mean: 16076 CPU hrs / antigen
7.9 secs / CDRH3
CPU hours for one antigen
CPU hours per CDRH3
Antigen size (AAs)
H
Antigen
Murine CDRH3 sequences
Human CDRH3 sequences
Overlap
# of sequences
Binding energy of 1 million CDRH3 sequences
I
Frequency of CDRH3 sequences binding with the specified structure
Murine CDRH3 sequences
Human CDRH3 sequences
Binding structure
J
Affinity class
avg [min .. max]
All
375 [42 .. 1164]
Top 1%
61.9 [7 .. 266]
# of antigens
# of binding structures
K
Affinity class
avg [min .. max]
All
64 [14 .. 163]
Top 1%
17.3 [3 .. 45]
# of antigens
# of epitopes
L
avg [min .. max]
2 [1 .. 9]
# of antigens
# of binding hotspots

Supplementary Figure 4 (refers to Figure 1C and Figure 2) | Computational complexity and binding diversity of the Absolut! database.

(A) AA size distribution of the 159 antigens (AA). (B). Species of origin of the antigens (blue: foreign and turquoise: self). (C) Resolution of the PDBs of the 159 antigens. (D) Distribution of the number of enumerated structures per antigen (tested docking poses for each CDRH3 affinity calculation). (E) Number of enumerated structures as a function of the antigen size. (F) Computational resources needed to generate the Absolut! database. Left axis: CPU hours to calculate the 6.9 million CDRH3 sequences for each antigen (as function of the antigen size). Right axis: average time to compute the affinity of one CDRH3 sequence to an antigen depending on its size. (G) Effect of glycans on binding profiles. Distribution of affinities of 1 million murine CDRH3 to antigens that contain glycans (See Table S1), when calculated for binding with (grey) or without (blue) these glycans. The grey curve is only shown when it is different from the blue one. (H,I) Affinities and binding conformations between 1 million murine CDRH3 and 1 million human CDRH3 sequences share the same ranges of affinities. (H) Distribution of binding energies (affinity). Human and murine sequences follow similar distributions of affinities for each antigen. (I) Distribution of the top 15 most used binding structures for each antigen. Human and murine CDRH3 sequences share the same usage of binding structures. Structures with only one color denote those that were not in the top 15 most used structures in the respective other category (mouse or human). (J–L) Structural diversity of binding for the CDRH3 sequences (either high affinity purple, top 1% or all sequences, blue) to each antigen. (J) Number of distinct antibody-antigen binding structures. (K) Number of epitopes. (L) Number of binding hotspots.

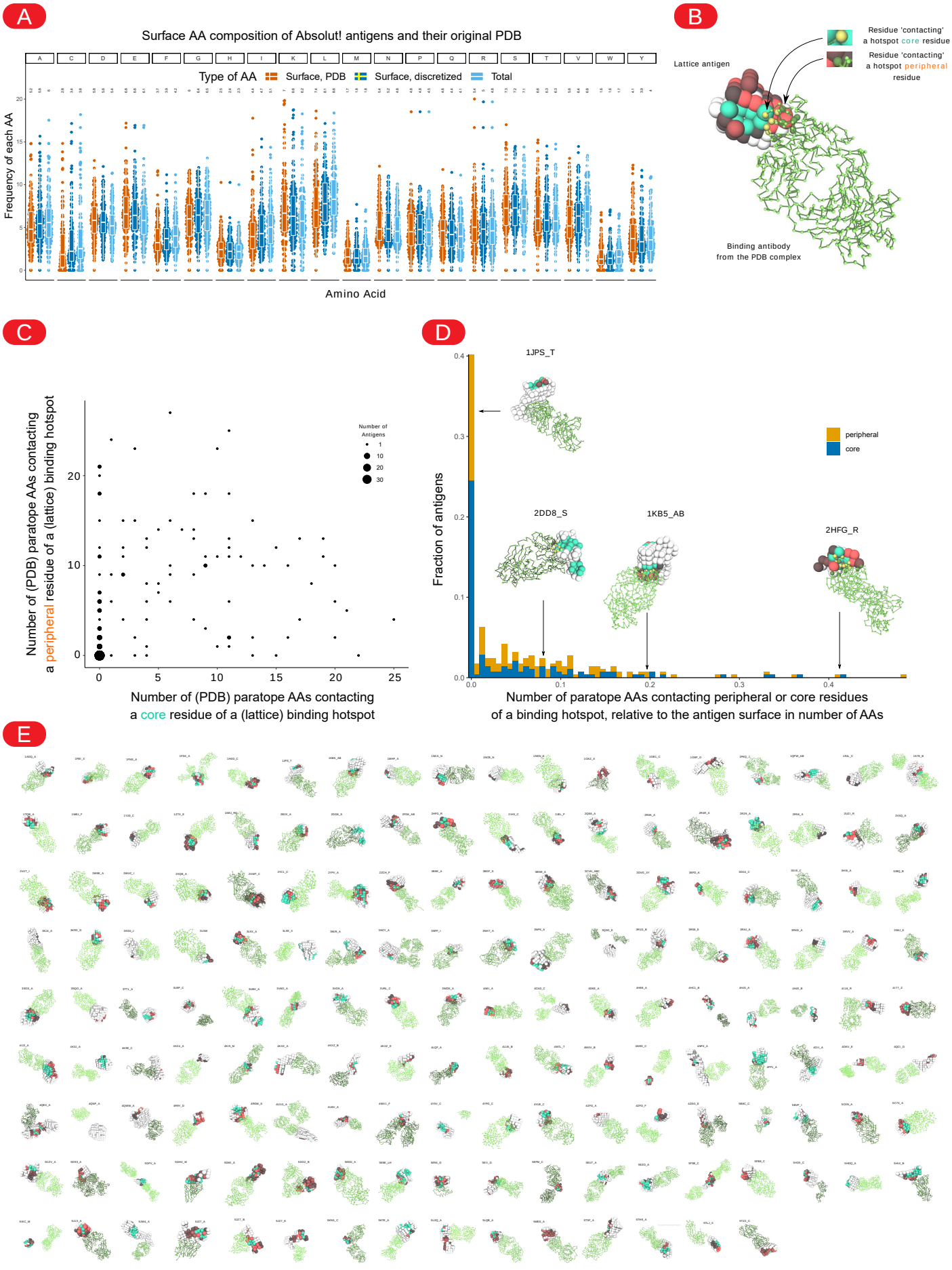


Supplementary Figure 5 (refers to Figure 1) | *In silico* formalizing of antibody-antigen binding features and their encodings for ML. (A) Definitions. An antibody slice refers to 11 consecutive AAs of the CDRH3, which is the size used to compute binding (see Methods) (*51*). AAs can only occupy integer 3D lattice positions encoded as a single integer number. A peptide structure is defined as a starting position and a list of moves in space, without knowing which AAs are in this peptide. An antibody binding is then represented as the best spatial conformation of a slice that minimizes its energy (according to the interaction of neighboring lattice residues from the Miyazawa-Jernigan potential, see Methods), and is stored as the AAs with their structure in 3D. The paratope refers to the spatial configuration of only the antibody AAs participating in the interaction, while the epitope is the spatial configuration of the antigen AAs that are involved in the binding. Therefore, we define "paratope" and "epitope" as referring to the full structural information of the binding from either side. In a binding structure, an AA's degree of binding refers to its number of binding partners on the opposite protein. Note that AAs of degree 2 on the antibody can bind to residues of degree 1 on the antigen (residue degree asymmetry). (B) Representations of the binding interface. The list of binding partners or the bipartite graph of interaction can conveniently describe some properties of the binding interface, while losing the spatial configuration of the epitope or paratope (surjective mapping). (C) Representations of the paratope or epitope. Paratope and epitope may be represented with different resolutions, so as to represent the type of available experimental information, or as encoding features for ML pipelines. The binding AAs can be annotated with their binding degree, or as "structural interaction motifs" (*3*) that only encode binding positions and gaps. The 3D environment of each AA may be encoded as a graph of neighbors for each range of distances (*55*), or as explicit presence of AAs around (local surface topology). Finally, the full conformation of a paratope or epitope can be given as a 3D matrix of voxels (3D cubic units) describing the type of AA at each position in the lattice, as used for protein docking strategies (*38*).

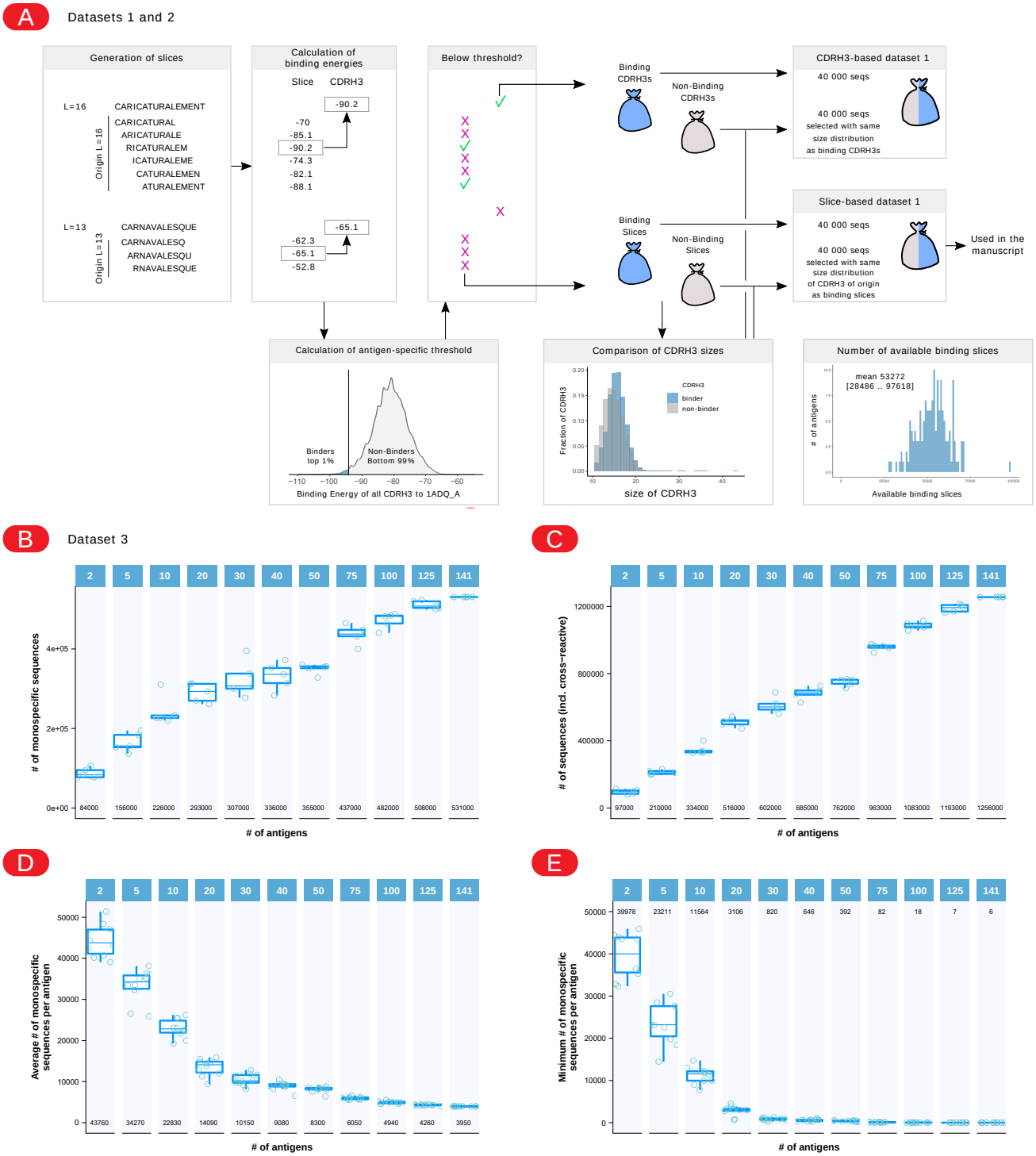

A
Non-Binders
1ADQ_A
Information content
Position
B
Binders
1ADQ_A
Information content
Position
C
Binders / Non-Binders
1ADQ_A
J-S divergence
Position
D
Binders / Non-Binders
1FBI_X
1FSK_A
1H0D_C
J-S divergence
Position
E
1ADQ_A
11-mer
5-mer
2-mer
p(pi, Ai and pj, Aj)
Binding sequences
Non-binding sequences
(position, AA)
F
1ADQ_A
p(pi, Ai | pj, Aj)
Binding sequences
G
1ADQ_A
p(pi, Ai | pj, Aj)
Non-binding sequences
H
1ADQ_A
Δp(pi, Ai | pj, Aj)
Binding sequences
Non-binding sequences

Supplementary Figure 6 (refers to Figure 2C) | Positional AA frequencies and positional dependencies of binder and non-binder CDRH3 sequences to antigen 1ADQ_A

The binding CDRH3 sequences to antigen 1ADQ (as an example) were assessed for positional dependencies. (A–C) AA positional bias between binder (from the top 1%) and non-binder (from the bottom 99%) sequences (11-mers) to 1ADQ_A (see Figure 2B for details). For each CDRH3 (binder or non-binder), its 11-mer with highest affinity was taken to build the aligned logo plot. (A) Positional AA usage is shown separately for non-binders, representing the sequence bias of CDRH3 sequences generated by VDJ recombination, as seen by the "CAR" sequence when the 11-mer is close to the beginning of the CDRH3 or the DYW sequence when the 11-mer is close to its end. (B) Positional AA usage of non-binder sequences, showing remaining biases from CDRH3 sequences but with a preferential usage of the end of the CDRH3 as seen by the DYW sequence. (C) Comparison of binder and non-binder positional AA usage shown as the Jensen-Shannon divergence between the distribution of AAs at each position between binders and non-binders. (D) Comparison of binder and non-binder positional AA usage analogous to panel C for three different antigens, 1FBI_X, 1FSK_A and 1H0D_C. (E–H) The positional dependencies are further illustrated by the conditional probabilities of having an AA at a position knowing the AA at another position, assessed on all subsequences of size 2 (right), 5 (middle) or 11 (left) AAs of the binders and non-binders. (E) The simultaneous appearance $p(p_i, A_i$ and $p_j, A_j)$ of two amino acids A_i and A_j at two different positions p_i and p_j are quantified in binders and non-binders separately. (F–G) The conditional probability $p(p_i, A_i \mid p_j, A_j)$ to observe a certain AA A_i at a position p_i knowing the presence of another AA A_j at a position p_j is shown, which normalizes the correlations (simultaneous appearance) by the AA composition difference between binders and non-binders shown in panel (C). Residues A_i (y axis) that show consistently higher conditional probability across all positions (blue horizontal lines in F,G and blue or red horizontal lines in H) reveal only a higher AA usage of A_i while local colored points show complex positional dependencies. (H) The difference $\Delta p(p_i, A_i \mid p_j, A_j) = p(p_i, A_i \mid p_j, A_j)$ (binder) - $p(p_i, A_i \mid p_j, A_j)$ (non-binder) between the conditional probabilities in binders and non-binders reveals binder-specific positional dependencies. The right panel (2-mer) is the one shown in Figure 2C.

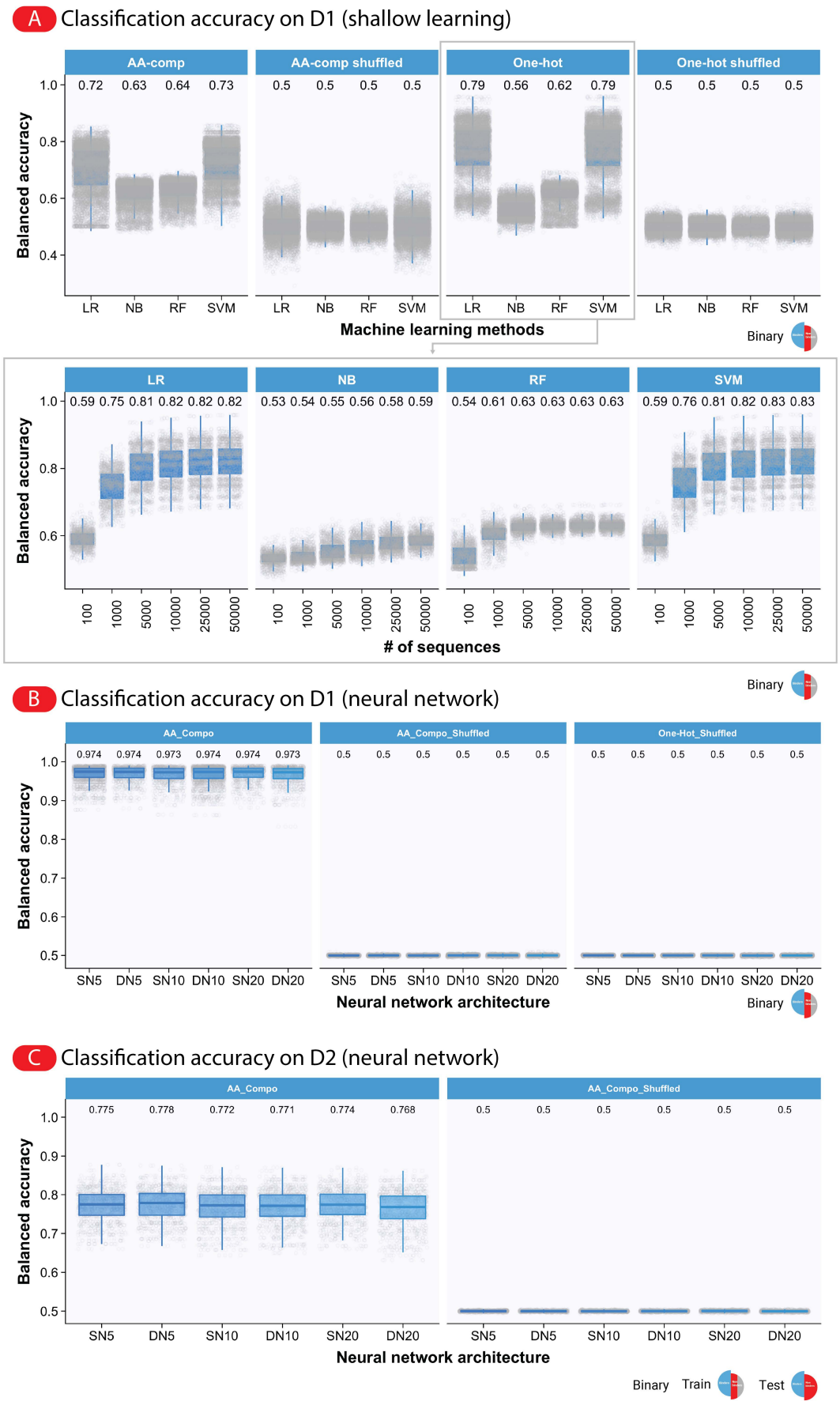

A
Surface AA composition of Absolut! antigens and their original PDB
Type of AA
Surface, PDB
Surface, discretized
Total
Frequency of each AA
Amino Acid
B
Lattice antigen
Residue 'contacting' a hotspot core residue
Residue 'contacting' a hotspot peripheral residue
Binding antibody from the PDB complex
C
Number of (PDB) paratope AAs contacting a peripheral residue of a (lattice) binding hotspot
Number of (PDB) paratope AAs contacting a core residue of a (lattice) binding hotspot
Number of Antigens
D
1JPS_T
2DD8_S
1KB5_AB
2HFG_R
peripheral
core
Fraction of antigens
Number of paratope AAs contacting peripheral or core residues of a binding hotspot, relative to the antigen surface in number of AAs
E

Supplementary Figure 7 | Concordance of Absolut! antibody-antigen binding structures with experimental crystal structures. (A) Surface AA composition of discretized antigens, compared to the experimental AA distribution in their original PDB file, or to the entire AA composition (not only surface, which is therefore identical between the PDB and lattice antigen). (B) Each antigen was built from PDB structures containing both the antigen and at least one binding antibody. Description of the overlap between the experimental antibody binding structure in the original PDB, and the predicted binding hotspots (see C). For a PDB antibody-antigen binding structure, paratope residues in the PDB, contacting a shared binding hotspot position (core, green) or a peripheral residue of a binding hotspot (dark red) in the lattice can be quantified. A PDB paratope residue is considered as contacting the antigen if it lies within a radius of 8Å to a lattice antigen residue, to mimic the lattice resolution of 5.25Å and the antigen discretization error RMSD around 3.5Å (Figure S3). The different possible hotspots are all "merged", meaning that in the lattice, by definition of the hotspots, each structure of a binder sequence in the Absolut! dataset was binding at least 4 green positions in one of the hotspots. (C) The number of PDB paratope residues contacting a core or peripheral residue of a binding hotspot were quantified. (D) Distribution of the number of paratope residues contacting a core and peripheral (side) epitope residues normalized by the number of AAs at the surface of the lattice antigen. 75% of PDB antigen-antibody complexes were in contact with the lattice hotspot core residues identified from the 6.9 million murine CDRH3 sequences, while the remaining structures show a high variation of overlap, from few paratope residues (relative to the total antigen surface) such as antigen 2DD8, up to a strong overlap in the case of antigen 2HFG. When inspecting peripheral residues of binding hotspots, 85% of antibody-antigen complexes overlapped with peripheral hotspot residues. (E) Experimental PDB antibody binding versus predicted hotspots for all the 159 Absolut! antigen-antibody complexes. In a few cases, the PDB antibody was binding to another protein chain than the one of the discretized antigen, in which case the PDB antibody and discretized antibody chain do not interact in the figure.



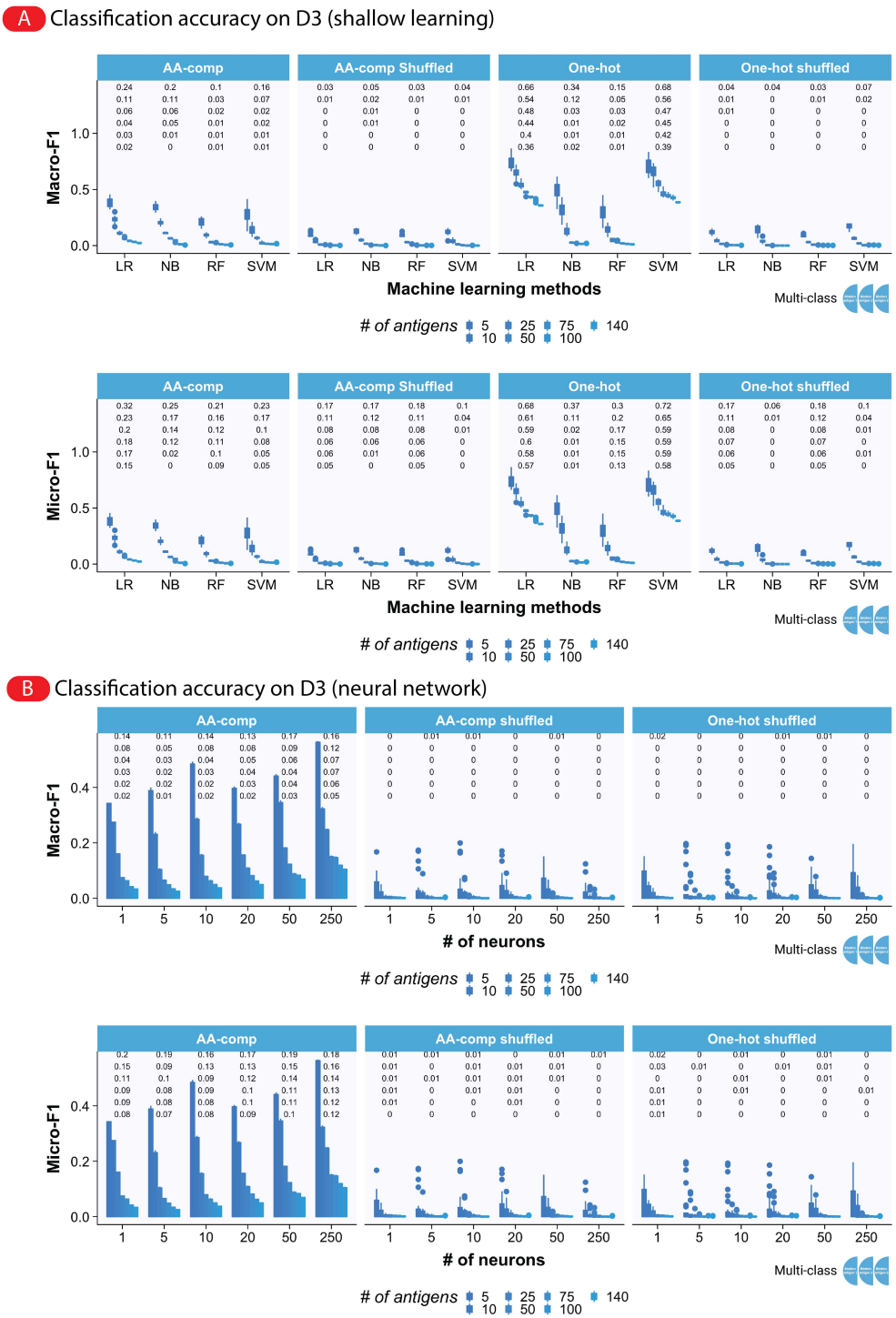
Supplementary Figure 8 (refers to Figure 3) | Generation and properties of classification datasets 1, 2 and 3

(A) Pipeline for the generation of dataset 1 for a chosen antigen (see Methods). 11-mer slices are generated from the CDRH3 sequences and assessed for binding affinity to the antigen. The lowest energy among slices defines the CDRH3 energy. A (top 1%) binding threshold is calculated and each CDRH3 (or each slice) is labeled with a 'binder' or 'non-binder' tag according to the CDRH3-defined top 1% threshold. The binding CDRH3s have a bias to be longer (and the binding slices tend to stem from longer CDRH3s), which creates an AA composition bias between binder and non-binder CDRH3s (or slices), due to the frequent "CAR" sequence in the beginning of CDRH3s. Therefore, for building a CDRH3-based or slices-based-dataset 1, 40 000 binders are taken and 40 000 non-binders are sampled that follow the same CDRH3 size distributions as the selected binders. (B–E) Number of available sequences for Dataset 3 as a function of the number N of considered antigens. (B) number of total monospecific sequences to one of the N antigens (but that can be cross-reactive to antigens others than the N selected ones); (C) number of total sequences (including cross-reactive sequences within the N selected antigens) to one or more of the N antigens; (D) average and (E) minimum number of monospecific sequences recognizing each antigen (i.e., belonging to each class).

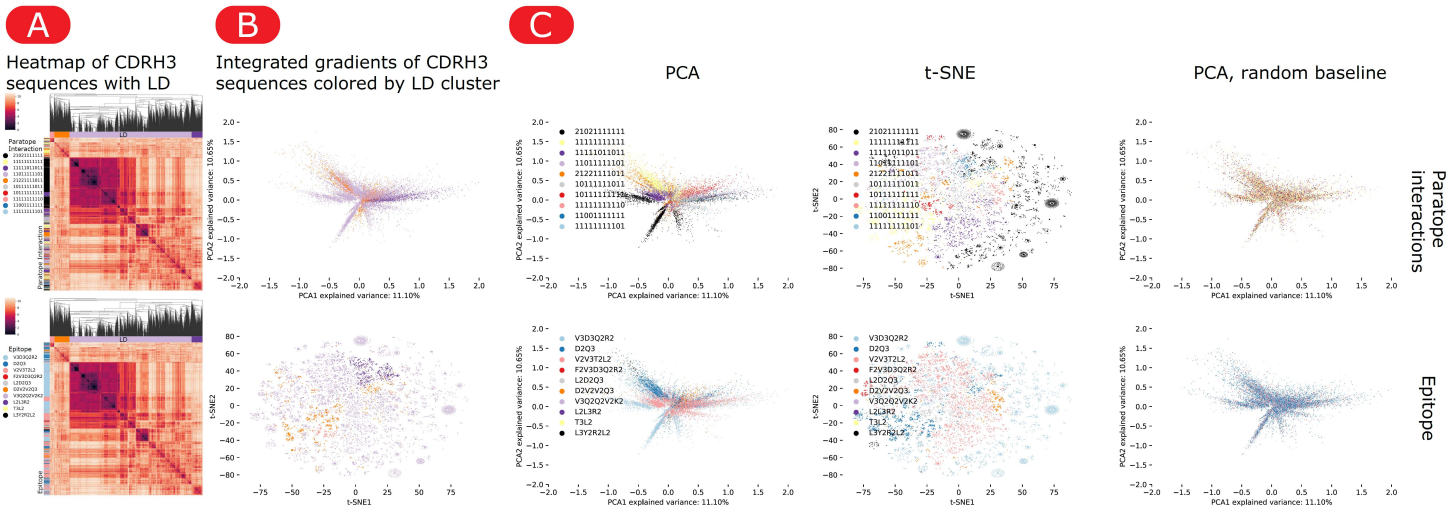

A Classification accuracy on D1 (shallow learning)
AA-comp
AA-comp shuffled
One-hot
One-hot shuffled
0.72 0.63 0.64 0.73
0.5 0.5 0.5 0.5
0.79 0.56 0.62 0.79
0.5 0.5 0.5 0.5
Balanced accuracy
LR NB RF SVM
Machine learning methods
Binary
LR
NB
RF
SVM
0.59 0.75 0.81 0.82 0.82 0.82
0.53 0.54 0.55 0.56 0.58 0.59
0.54 0.61 0.63 0.63 0.63 0.63
0.59 0.76 0.81 0.82 0.83 0.83
100 1000 5000 10000 25000 50000
# of sequences
B Classification accuracy on D1 (neural network)
AA_Compo
AA_Compo_Shuffled
One-Hot_Shuffled
0.974 0.974 0.973 0.974 0.974 0.973
0.5 0.5 0.5 0.5 0.5 0.5
SN5 DN5 SN10 DN10 SN20 DN20
Neural network architecture
C Classification accuracy on D2 (neural network)
0.775 0.778 0.772 0.771 0.774 0.768
Train
Test

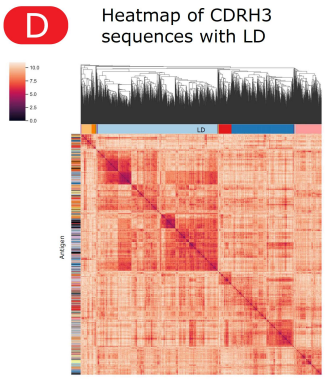
Supplementary Figure 9 (refers to Figure 3) | Classification accuracy on D1 and D2 for amino acid composition (AA-comp) and one-hot encodings.

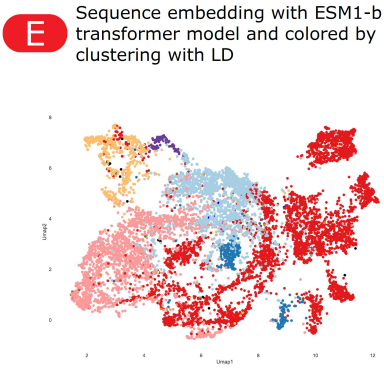
(A) Shallow learning. In addition to one-hot encoding, we used the frequency of the AAs (AA-comp) to encode CDRH3 sequences and as a control we trained the models on label-shuffled datasets (shuffled). Binary classification median balanced accuracy (shuffled accuracy) ranged between 0.64–0.73 (0.5) and 0.56–0.79 (0.5) for AA-comp and one-hot encoding respectively. Maximum accuracy (0.83) was obtained by SVM trained on the largest training dataset (50 000 sequences). (B) AA-comp encoded neural network binary classification median balanced accuracy ranged between 0.816–0.82 and shuffled median balanced accuracy for one-hot encoding 0.5 (see Figure 3 for accuracy values of models with one-hot encoding). (C) Median classification accuracy of neural network trained on D1 and tested on D2 for AA-comp ranged between 0.768–0.778 (shuffled accuracy at 0.5). For each antigen, model, encoding and $n_{CDRH3_training}$, training and test were replicated 10 times.

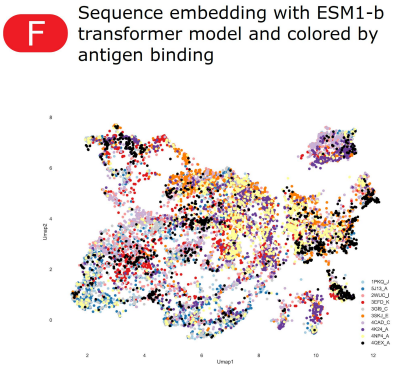


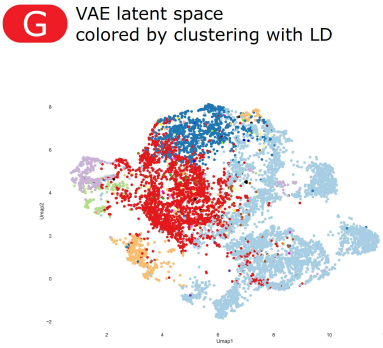
Supplementary Figure 10 (refers to Figure 3) | Classification accuracy on D3 for AA composition (AA-comp) and one-hot encodings. Accuracy of shallow architectures, shuffled control and AA composition encoding are assessed on D3 generated for different antigen numbers as described in Figure 3. (A) Shallow learning. Macro-F1 multi-class classification median accuracy (shuffled accuracy) ranged between 0–0.24 (0–0.05) and 0.01–0.68 (0–0.07) for AA-comp and one-hot encoding respectively. Micro-F1 multi-class classification median accuracy (shuffled accuracy) ranged between 0–0.32 (0–0.18) and 0.01–0.72 (0–0.18) for AA-comp and one-hot encoding respectively. (B) Neural network. Shuffled Macro-F1 multi-class classification median accuracy ranged between 0–0.01 and 0.02 for AA-comp and one-hot encoding respectively. Shuffled Micro-F1 multi-class classification median accuracy ranged between 0–0.01 and 0.03 for AA-comp and one-hot encoding respectively. See Figure 3 for unshuffled accuracy. For each model, encoding and # of antigens, training, and test were replicated 10 times.

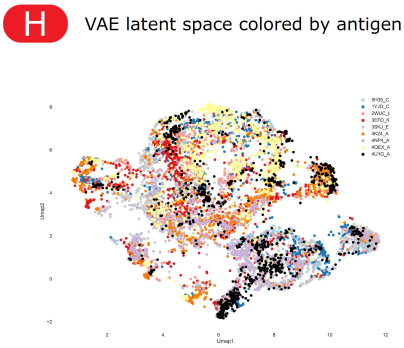

A
Heatmap of CDRH3 sequences with LD
B
Integrated gradients of CDRH3 sequences colored by LD cluster
C
PCA
t-SNE
PCA, random baseline
Paratope interactions
Epitope
PCA1 explained variance: 11.10%
PCA2 explained variance: 10.65%
t-SNE1
t-SNE2


D
Heatmap of CDRH3 sequences with LD
LD
Antigen


E
Sequence embedding with ESM1-b transformer model and colored by clustering with LD
Umap1
Umap2

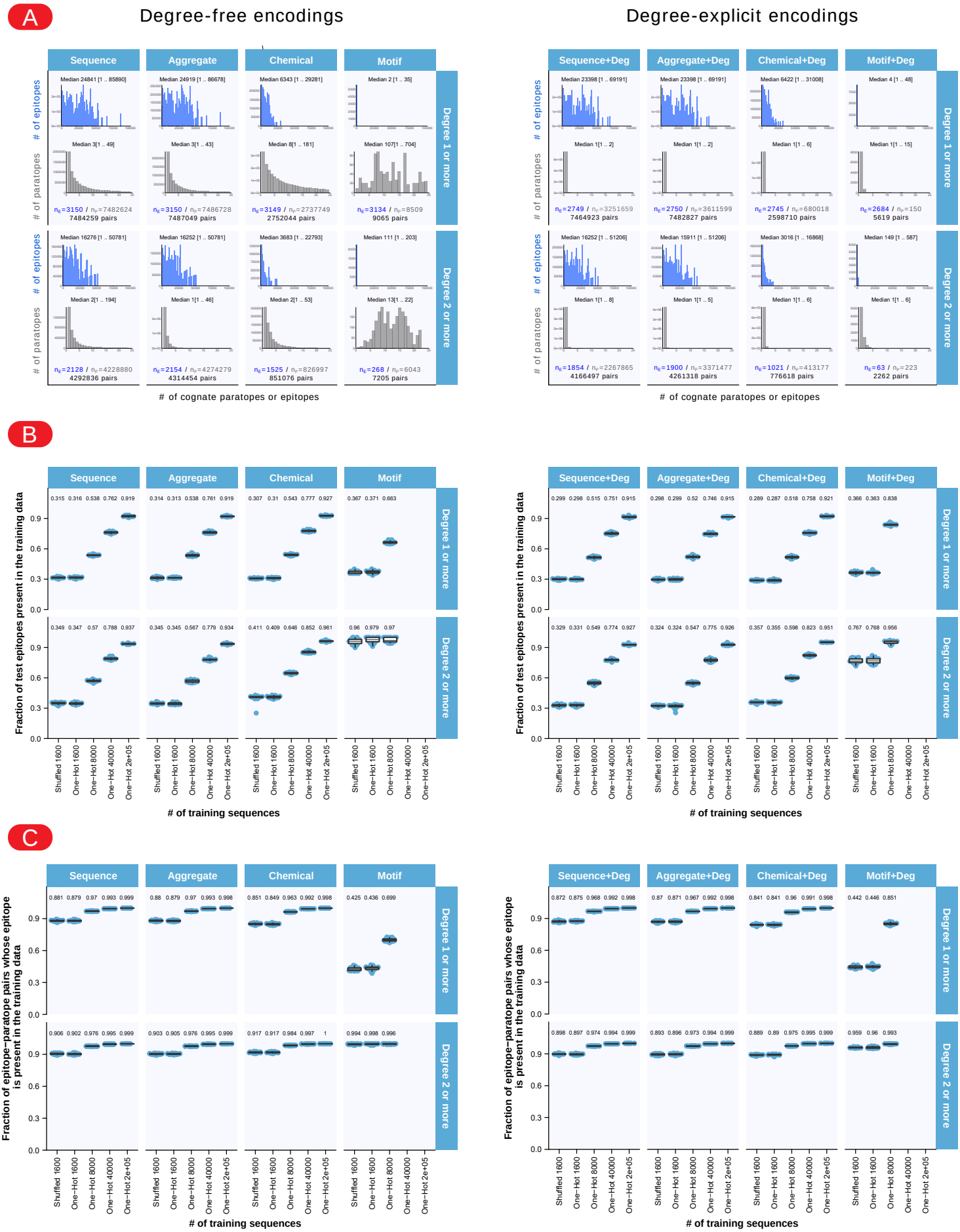

F
Sequence embedding with ESM1-b transformer model and colored by antigen binding
Umap1
Umap2


G
VAE latent space colored by clustering with LD
Umap1
Umap2

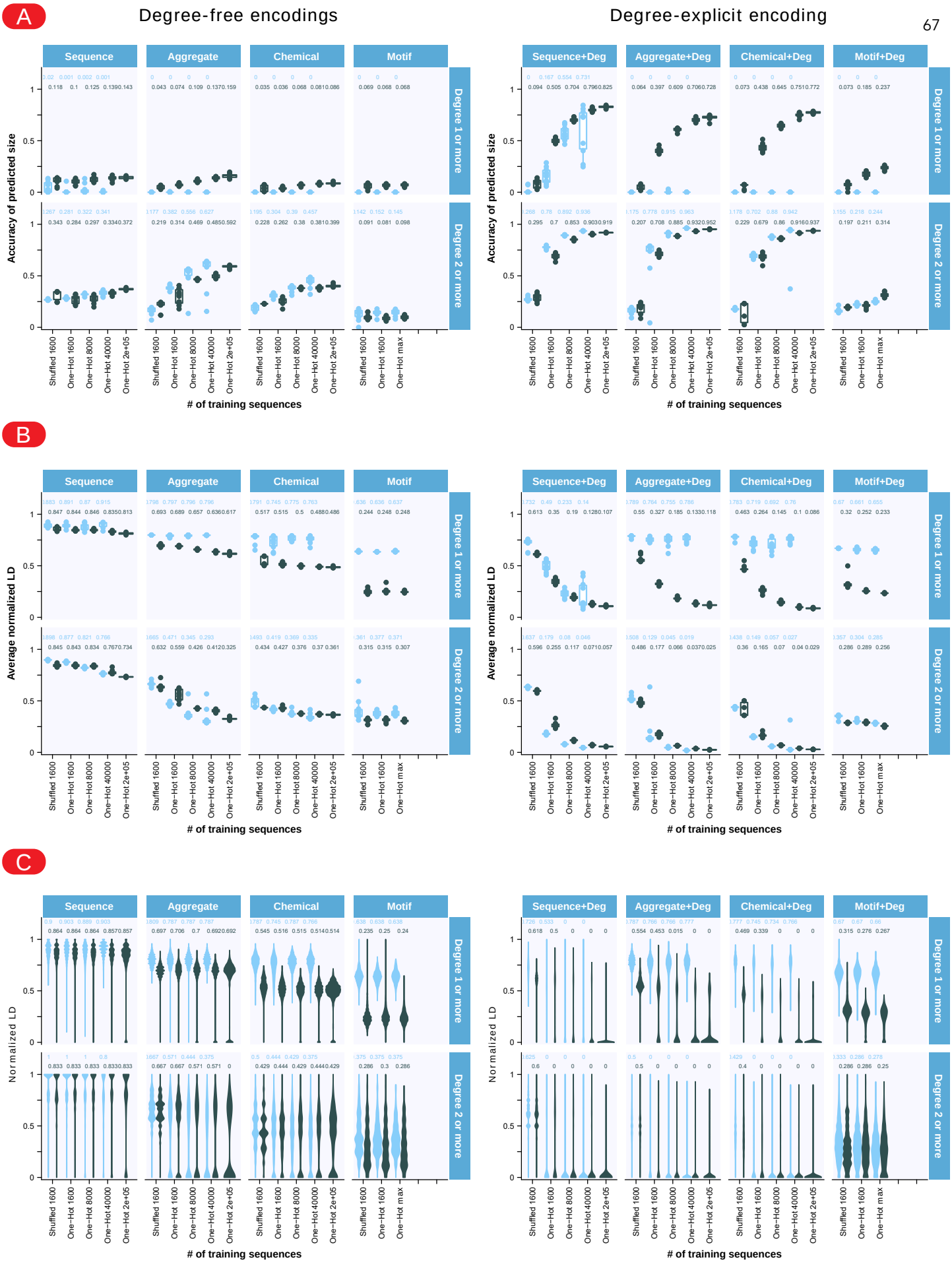

H
VAE latent space colored by antigen
Umap1
Umap2

Supplementary Figure 11 (refers to Figure 3) | Structural properties of antibody binding can be discriminated using sequence-trained model integrated gradients, but not using sequence similarity, sequence clustering, nor VAE-based latent space representations.

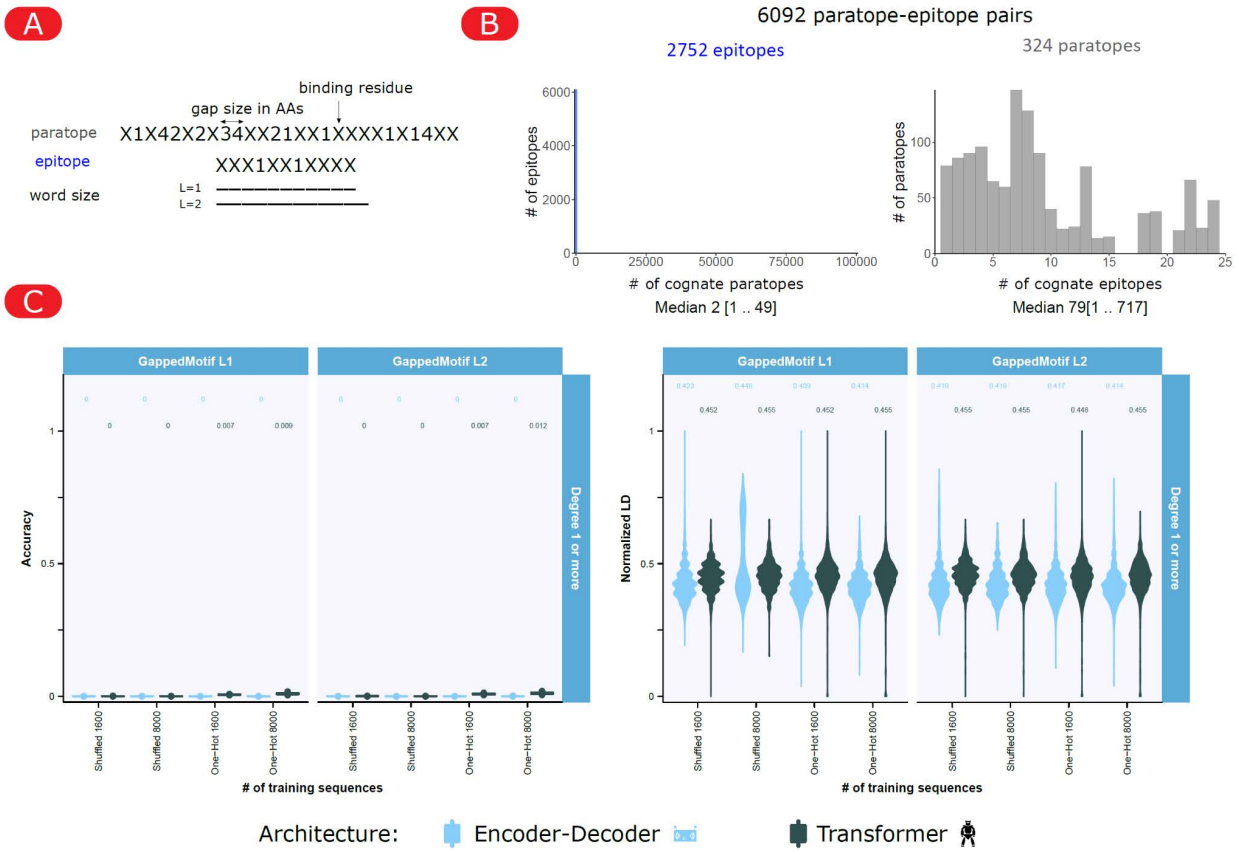
(A) Similar sequences do not cluster according to paratope (degree-explicit motif encoding, top) nor epitope encoding (degree-explicit sequence encoding, bottom). 25 000 binder 11-mer sequences to 1ADQ_A from D1 (Figure S8) were clustered according to their LD, identifying 10 clusters (horizontal color bar) in the heatmaps. The two heatmaps are identical but showing as vertical color bars the annotation of sequences with either their paratope (top, 10 colors) or epitope (bottom, 10 colors) encoding. (B–C) Integrated gradients can inform whether CDRH3 sequences share a paratope or epitope encoding PCA and t-SNE of integrated gradients of each binder sequences, computed on the SN10 architecture trained with 40 000 training sequences (dataset D1, antigen 1ADQ_A, containing both binders and non-binders Figure S8A), colored according to different properties of the sequences: (B) colored according to the 10 clusters generated by agglomerative clustering on the LD between AA sequences (see horizontal color bar in panel A); (C) colored with associated paratope encoding (top) or epitope encoding (bottom); or colored with shuffled paratope/epitope labels as a comparison (right) for detecting confirmation bias. Sequence coloring by paratope (top) is well separated in the integrated gradients-generated PCA and t-SNE, while agglomerative clustering based on LD sequence similarity could not recover these binding-specific informations. This suggests that taking a new sequence, calculating the integrated gradients of this sequence from the trained model, and integrating the gradients vector into the t-SNE or PCA could be predictive of its binding paratope encoding. In comparison, sequences colored by epitope encoding show more than one cluster for the same color, reminiscent of different modes of binding (paratopes) to the same epitope. (D) Similar CDRH3 sequences bind to different antigens. Sequences from D3 with mono-specific binding sequences to 20 different antigens were clustered according to their similarity (LD), leading to the delineation of 23 clusters (horizontal color bar), and colored according to which antigen they bind (vertical color bar, 20 colors). Clusters of sequence similarity were not predictive for the sequences' target antigen. (E–H): Neither VAE architecture nor ESM1-b transformer embeddings can isolate CDRH3 sequences by antigen specificity. To save computational time, 233 605 monospecific sequences binding monospecifically 10 different antigens from panel (D) were embedded in the latent space of a VAE (lower row) or through the ESM1-b transformer model (*89*) (upper row) and reduced with UMAP (see Methods). Sequence embeddings are colored according to their LD cluster identified from (D) (left) and by antigen recognition (right). 10 000 embedded sequences are shown for the VAE and 9 000 for the ESM1-b.


A
Degree-free encodings
Degree-explicit encodings
Sequence
Aggregate
Chemical
Motif
Sequence+Deg
Aggregate+Deg
Chemical+Deg
Motif+Deg
Degree 1 or more
Degree 2 or more
# of epitopes
# of paratopes
# of cognate paratopes or epitopes
B
Fraction of test epitopes present in the training data
# of training sequences
Shuffled 1600
One−Hot 1600
One−Hot 8000
One−Hot 40000
One−Hot 2e+05
C
Fraction of epitope−paratope pairs whose epitope is present in the training data

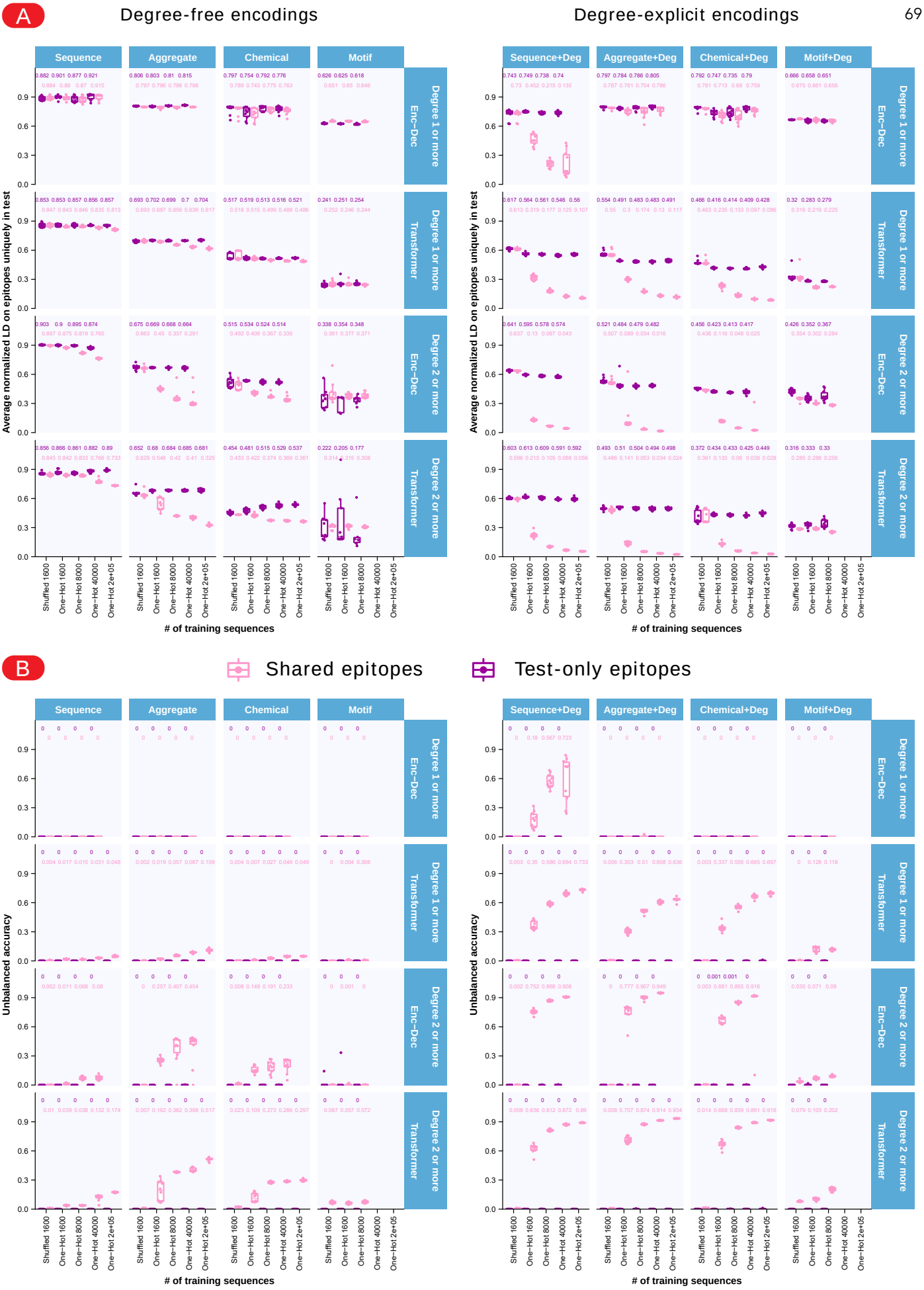
Supplementary Figure 12 (refers to Figure 4) | Attributes of the paratope-epitope dataset depending on the encoding.
(A) Numbers of cognate paratopes and epitopes with respect to encodings and degree filtering for degree-free (left) and degree-explicit (right). The largest number of pairs was observed for the sequence encoding (4 292 836 and 4 166 497 for degree-free and degree explicit) and the smallest number of pairs was observed for the motif encoding (7 205 and 2 262 for degree-free and degree explicit respectively). (B) Fraction (relative overlap) of test epitopes that were also present in the training data. Broadly the overlap between test and training epitopes increased as a function of the number of training sequences. (C) Fraction of paratope-epitope pairs whose epitopes were present in the training data. As in (B), the overlap increased as a function of the number of training sequences. 10 replicates are shown in each panel/condition.


A
Degree-free encodings
Degree-explicit encoding
Sequence
Aggregate
Chemical
Motif
Sequence+Deg
Aggregate+Deg
Chemical+Deg
Motif+Deg
Degree 1 or more
Degree 2 or more
Accuracy of predicted size
B
Average normalized LD
C
Normalized LD
Shuffled 1600
One-Hot 1600
One-Hot 8000
One-Hot 40000
One-Hot 2e+05
One-Hot max
# of training sequences
Architecture:
Encoder-Decoder
Transformer

Supplementary Figure 13 (refers to Figure 4) | Performance of the transformer and encoder-decoder architectures on the paratope-epitope prediction problem with different encodings.
(A–C) Additional accuracy measures for both architectures depending on a large panel of encodings, for degree-free (left) or degree-explicit (right) encodings, with (upper facet) or without (lower facet) residues binding with degree 1. Each point is an independent training, 10 different replicates per condition were performed. Learning on a shuffled dataset is compared with learning on 1 600–40 000 (and 200 000 only for the Transformer architecture) unique paratope-epitope pairs, while all conditions are tested on 100 000 unique pairs when available, or downscaled to reach 80% training and 20% testing (see Methods). The "motifs" encodings only allowed for 2 262 to 9 065 unique paratope-epitopes pairs (see Figure S12A), which are referred to as One-Hot (Max) conditions, and higher dataset size is therefore not available). (A) Accuracy of predicting the proper epitope size: fraction of predicted epitopes with the same length as the target epitope. (B) Average Levenshtein distance between predicted and target epitope, normalized to the length of the largest of the two. (C) Distributions of normalized Levenshtein distance between predicted and target (test set) epitopes.

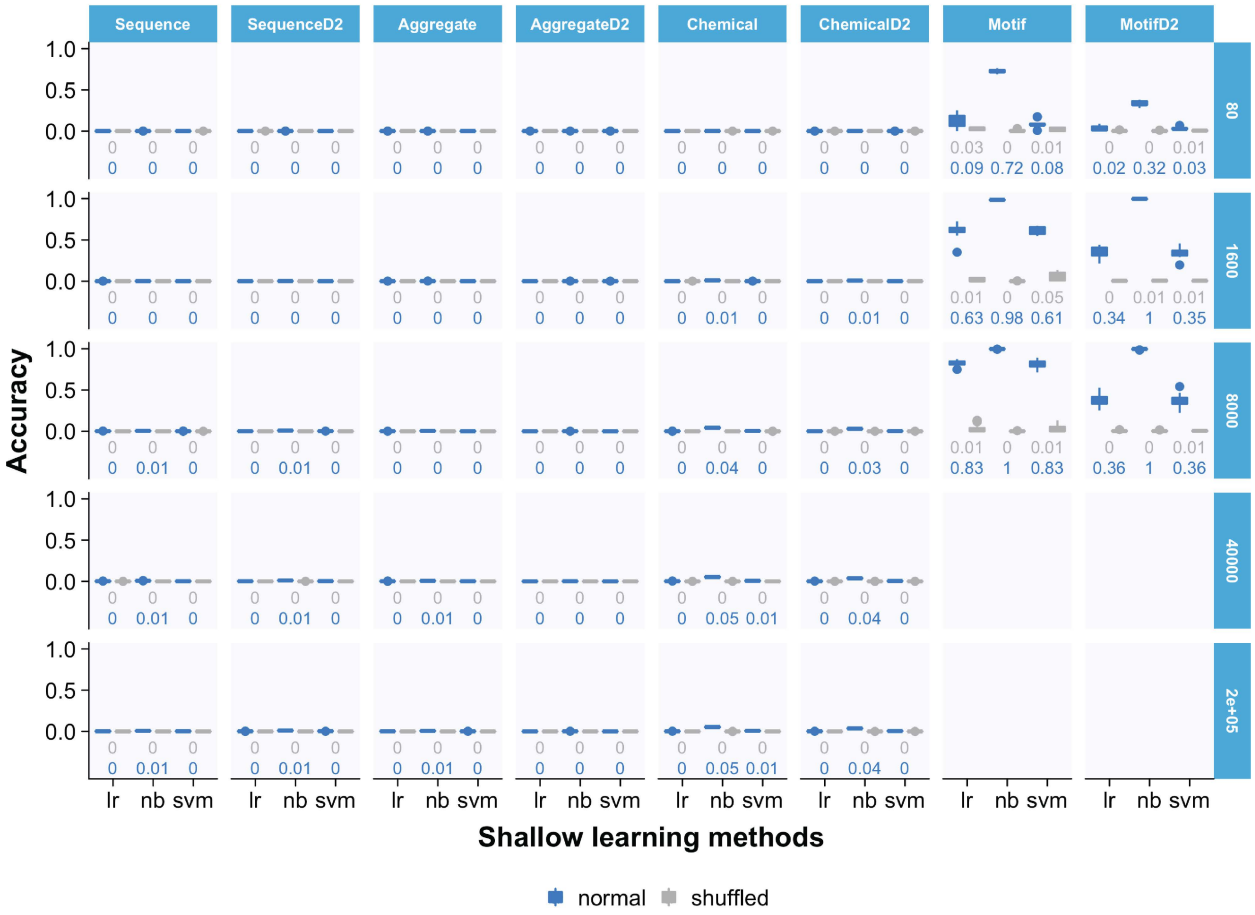


Supplementary Figure 14 (refers to Figure 4) | Prediction accuracy of the paratope-epitope prediction problem with the transformer and encoder-decoder architecture, using the "Gapped Motif" encoding.
(A) Illustration of the gapped motif encoding. Here, interacting AA residues were encoded as the string X and the size (number) of non-interacting residues as integers. (B) The distribution of the number of cognate paratopes (left) and cognate epitope (right). A total of 6092 paratope epitope pairs were observed with 2752 and 324 unique paratope and epitope respectively. (C) Prediction accuracy (left) and LD (right) for encoder-decoder (blue) and transformer (black). Broadly, both architectures yielded poor prediction accuracy (~0) and no increased LD as compared to the shuffled controls (LD ~0.4).


A
Degree-free encodings
Degree-explicit encodings
Sequence
Aggregate
Chemical
Motif
Sequence+Deg
Aggregate+Deg
Chemical+Deg
Motif+Deg
Degree 1 or more Enc-Dec
Degree 1 or more Transformer
Degree 2 or more Enc-Dec
Degree 2 or more Transformer
Average normalized LD on epitopes uniquely in test
Shuffled 1600
One-Hot 1600
One-Hot 8000
One-Hot 40000
One-Hot 2e+05
# of training sequences
B
Shared epitopes
Test-only epitopes
Unbalanced accuracy

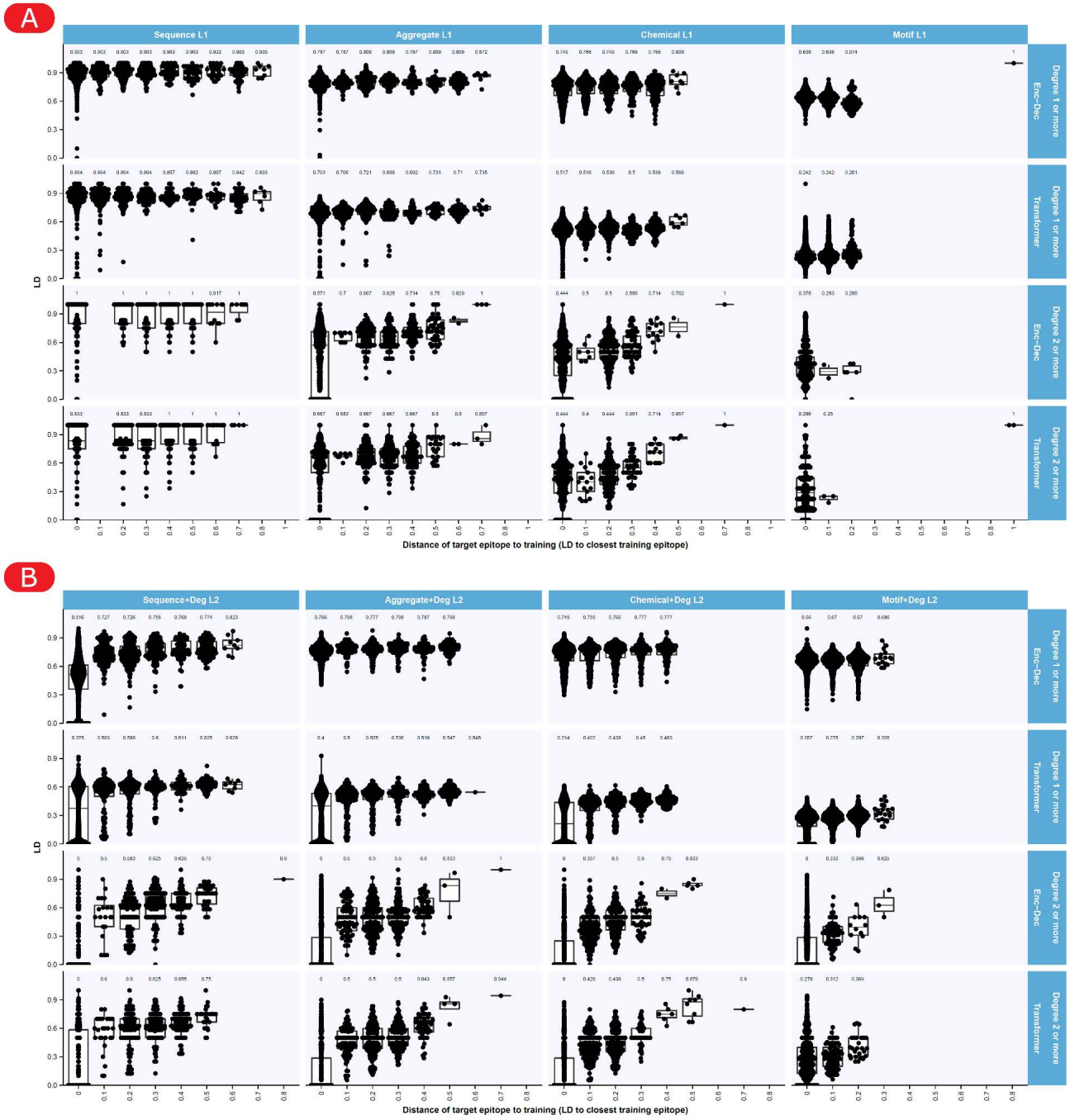
Supplementary Figure 15 (refers to Figure 4) | Paratope-epitope prediction accuracy on paratope-epitope pairs whose epitope was present in the test dataset or not.

To examine the prediction accuracy of epitopes that were not present in the training set (test-only epitopes, called "neo epitopes" here), we calculated separately the prediction accuracy (B) and LD (A) of shared (present in both training and test datasets) and test-only epitopes. Broadly the test-only epitopes yielded markedly lower accuracy (B) and larger LD (A) in comparison to shared epitopes across all encodings and degree filtering.



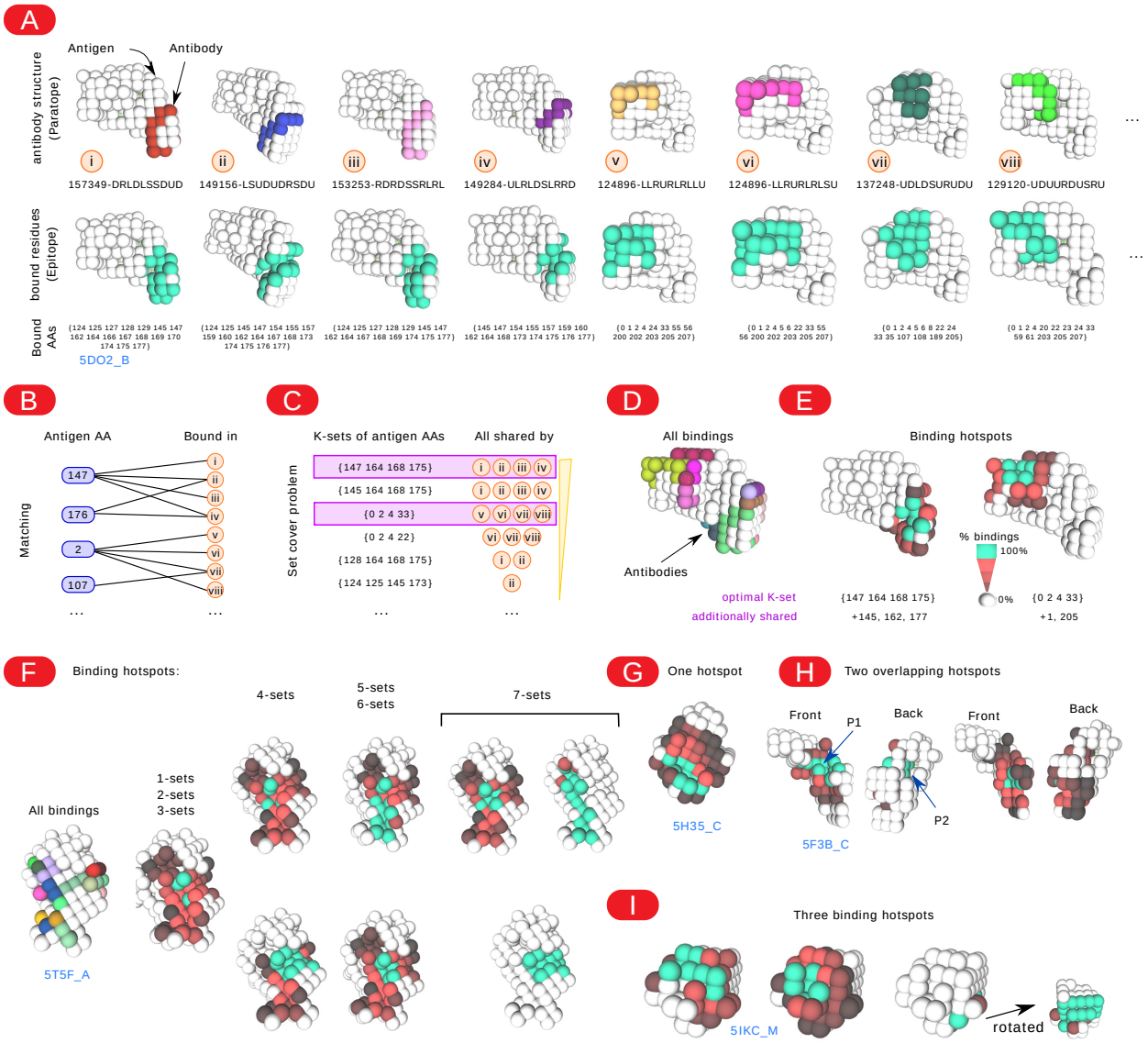
Supplementary Figure 16 (refers to Figure 4) | Paratope-epitope prediction with shallow learning methods.

As in Figure 4, we examined the performance of shallow learning methods in predicting paratope-epitopes pairs (degree-free encoding). The highest accuracy (shuffled accuracy) values ranged between 0.36–1 (0–0.01) for the model trained on the largest training dataset ($n_{CDRH3_training}$=8 000) and motif encoding (MotifD2). We note that for the motif encoding the maximum $n_{CDRH3_training}$ are 4 495 and 1 809 but we show them in the 8 000 row. In contrast to Figure 4, aggregate and chemical encodings did not improve the prediction accuracy even with increased size of training datasets indicating that shallow learning methods were unable to capture long-range sequence dependencies in the data. Due to the large memory footprint required to train shallow learning models with 200 000 training sequences, the RF classifier was excluded in this analysis as it does not support incremental training (minibatch). For each model, encoding and $n_{CDRH3_training}$, training and test were replicated 10 times.

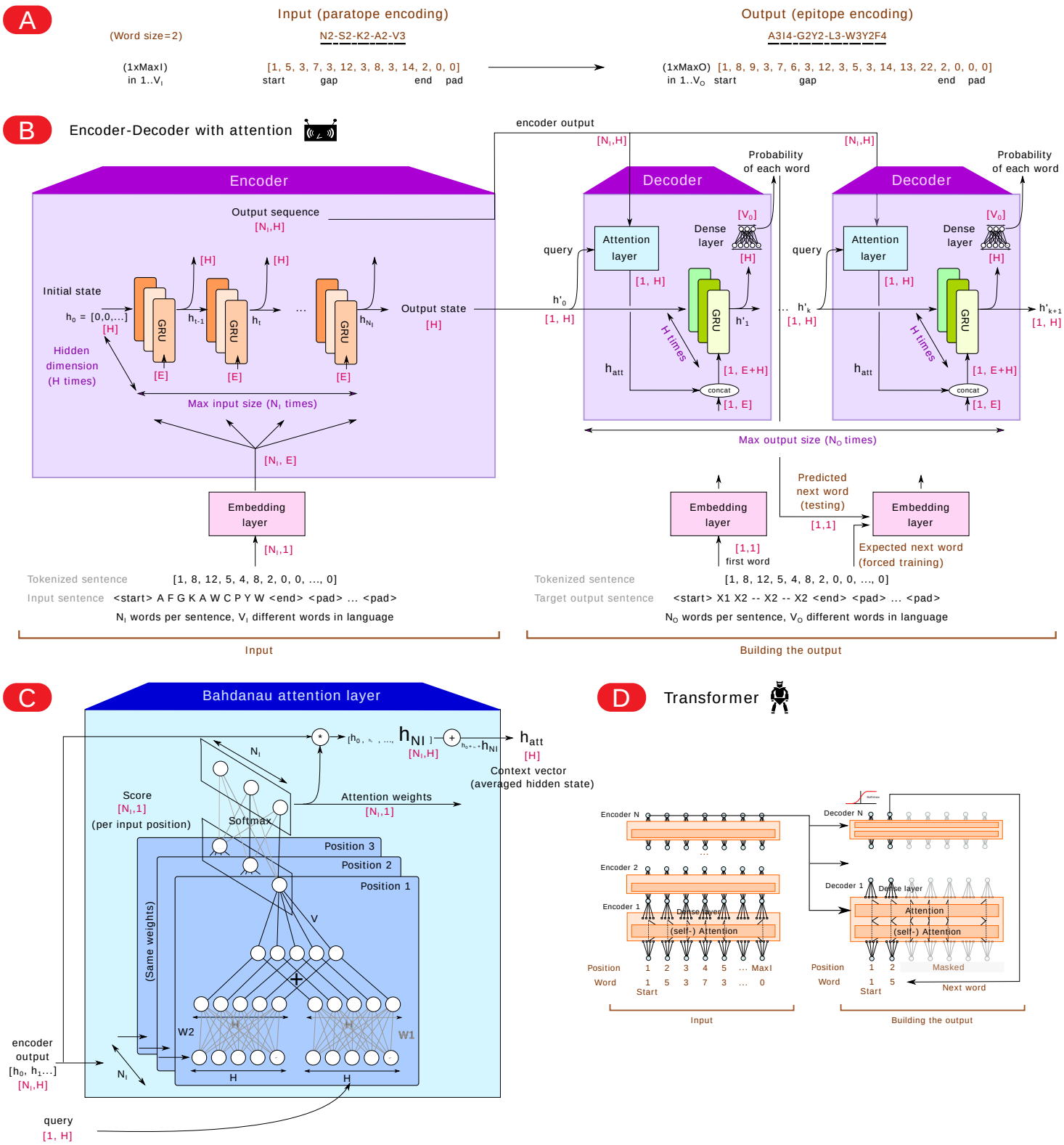


Supplementary Figure 17 (refers to Figure 4) | Levenshtein distance of predicted epitopes depending on their similarity to epitopes in the training dataset (x-axis).

For each test paratope-epitope pair, the closest epitope in the training dataset (lowest LD) was identified, defining a distance between the epitope and the training dataset (x axis). The LD of the predicted epitope to the target epitope (y axis) is shown for 5000 pairs in each condition (or all the test pairs for the motifs encoding). 1600 sequences were used for training (enabling to have a high fraction of test-specific epitopes compared to larger training datasets, see Figure S12B,C). The conditions for which close epitopes led to a better accuracy suggest that the trained model can learn patterns of sequence-wise close epitopes (degree-explicit encodings, with degree at least 2 and chemical degree-implicit also with degree at least 2), and that trained models may be applied to unseen epitopes (neoepitopes) provided they are not too different from the training epitopes. (A) Degree-free encodings. (B) Degree-explicit encodings.



Supplementary Figure 18 (refers to Methods) | Generation of binding hotspots from a list of antibody binding structures.
(A) The epitope residues of each binding (i to viii, upper panel) are extracted (turquoise residues, lower panel), provided they bind to an antibody residue of degree at least D (D=1 in this example). The IDs (numbering) of the epitope residues are shown below the images. (B) Each antigen residue is matched with all corresponding binding structures where it belongs to the epitope. (C) All possible K-sets (example with K=4 here) of residues are associated with the binding structures that share these K residues. The K-set covering the most structures is taken as the first binding hotspot; these structures are removed, and the next K-set that covers the most remaining structures is taken as the second binding hotspot, etc. The two hotspots for 5DO2_B are highlighted in purple. (D) Example of all binding structures of the 6.9 million murine CDRH3 sequences around the antigen with PDB ID 5DO2, chain B. Each antibody (11-mer) structure has a different color and potentially overlaps with other antibodies. (E) The two binding hotspots associated with all binding structures to antigen 5DO2_B. Each hotspot represents a group of epitopes (/binding structures). Among a group, the turquoise residues are 100% shared, and are K AA or more by definition. The level of red is a 3D heatmap on the percent of hotspot associated antibodies that cover this residue (with the degree constraint), and therefore shows the diversity of epitope residues from different binding structures within a binding hotspot. (F) Impact of the size of K-sets on the number of hotspots and their shape for antigen 5T5F_A. Smaller values of K cluster binding structures with a too small overlap in their epitopes. Bigger values of K lead to duplicate overlapping hotspots, and K=4 was chosen as a good trade-off among 50 inspected antigens (not shown) (G–I) Example of the diversity of hotspots for the 6.9 million sequences, with two non-overlapping ones in E, a unique one in G, two non-overlapping hotspots in H, or three overlapping hotspots in I. Although the hotspots tend to form concave topologies, the antigen 5F3B_C in panel H shows a more "immunogenic" pocket (P1) and a "non-immunogenic" pocket (P2), demonstrating that antigen pockets are not necessarily binding hotspots in the Absolut! framework.


A
Input (paratope encoding)
(Word size=2)
N2-S2-K2-A2-V3
(1xMaxI)
in 1..V_I
[1, 5, 3, 7, 3, 12, 3, 8, 3, 14, 2, 0, 0]
start gap end pad
Output (epitope encoding)
A3I4-G2Y2-L3-W3Y2F4
(1xMaxO) [1, 8, 9, 3, 7, 6, 3, 12, 3, 5, 3, 14, 13, 22, 2, 0, 0, 0]
in 1..V_O start gap end pad
B
Encoder-Decoder with attention
encoder output
Encoder
Output sequence
[N_I,H]
Initial state
h_0 = [0,0,...]
[H]
Hidden dimension (H times)
GRU
[E]
Output state
[H]
Max input size (N_I times)
[N_I, E]
Embedding layer
[N_I,1]
Decoder
Probability of each word
Attention layer
query
Dense layer
[V_O]
[1, H]
h'_0
h'_1
h'_k
h'_{k+1}
H times
h_att
[1, E+H]
concat
[1, E]
Max output size (N_O times)
Predicted next word (testing)
[1,1]
first word
Expected next word (forced training)
Tokenized sentence [1, 8, 12, 5, 4, 8, 2, 0, 0, ..., 0]
Input sentence <start> A F G K A W C P Y W <end> <pad> ... <pad>
N_I words per sentence, V_I different words in language
Target output sentence <start> X1 X2 -- X2 -- X2 <end> <pad> ... <pad>
N_O words per sentence, V_O different words in language
Input
Building the output
C
Bahdanau attention layer
N_I
Score
[N_I,1]
(per input position)
Softmax
Attention weights
[N_I,1]
h_NI
[N_I,H]
h_att
[H]
Context vector (averaged hidden state)
Position 3
Position 2
Position 1
(Same weights)
V
W2
W1
H
encoder output
[h_0, h_1...]
[N_I,H]
query
[1, H]
D
Transformer
Encoder N
Encoder 2
Encoder 1
Decoder N
Decoder 1
Dense layer
(self-) Attention
Attention
Position
Word
Start
MaxI
Masked
Next word
Input
Building the output

Supplementary Figure 19 (refers to Methods and Figure 4) | Architecture of the encoder-decoder (NMT) and transformer used for the paratope-epitope prediction task.

(A) Input paratope-epitope pairs were encoded according to a range of sequence-structures encodings as described in Figure 4A, start and end positions were encoded by the start and end tokens, and sequences were padded with zero-padding to match the length of the longest sequence in the dataset. Degree-explicit encodings are tokenized by words of size two (with gaps as one word, as shown here), while degree-free encodings are tokenized by words of size one. (B) Illustration of the encoder-decoder architecture (neural machine translation, NMT (*65*)). Numerical representation of the input (paratope) sequence was learned by the embedding layer, the output of the embedding layer was passed to the encoder layer (gated recurrent unit, GRU of hidden dimension 512; see Methods) and finally a decoder layer with attention was used to translate the encoded input into an epitope. A Bahdanau attention layer (see panel C) weighs the encoder outputs according to the previous hidden state of the decoder. Numbers in brackets denote tensor dimensions. N_I: maximum input size; N_O: maximum output size; E: Embedding dimension (6); H: Hidden layer dimension (512); V_I/V_O: number of possible tokens in input/output tokenized language. GRUs with the same color share the same parameters. Word by word, the output of the decoder is transformed into the next token of highest weight (softmax). Forced training denotes when the next target word (not the predicted one) is used to generate the next token. (C) Illustration of the Bahdanau attention layer. (D) Illustration of the transformer architecture (*66*). The encoder layer comprises N stacked-encoders with self attention and the decoder layer comprises N stacked-decoders with attention and self-attention layers (see Methods).

Supplementary Table 1 (refers to Figure 2A) | List of the 159 antigens provided in the Absolut! database.

The PDB ID, the discretized chain, and information on the antibody-antigen crystal structure are provided. (*) these antigens with more than one discretization were not considered for ML tasks involving multiple antigens, to avoid considering the same antigen twice, leading to 142 remaining antigens for multi-class antigen ML tasks (see Figure 3, Figure S8). (⌘) antigens containing glycans.

PDB	Chain	Antigen	Antibody Name	Species of Antigen	Linked Disease / Relevance	Resolution (Å)
4AEI	A	Aah2 scorpion α-mammal toxin scorpion	Fab4C1	Androctonus australis	Venom	2,3
4OKV	E	Anopheles anti-platelet protein	8H7	Anopheles stephensi	Anticoagulant / Mosquito bite	1,8
1FSK	A	major birch pollen allergen Bet v 1	BV16	Betula pendula	Allergy	2,9
1RJL	C	OspB-CT: Outer surface protein B	H6831	Borrelia burgdorferi	Infection - Lyme Disease	2,6
1OSP	O	OSPA: Outer Surface Protein A	184.1			2
5L0Q	A	ADAM10: Disintegrin and metalloproteinase domain-containing protein 10	8C7	Bos Taurus	-	2,8
2R56	A	BLG: β-lactoglobulin (allergen of bovine milk)	-	Bos taurus	Allergy	2,8
3JBQ	B	Phosphodiesterase 5/6 chimera catalytic domain	ROS-1	Bos Taurus		11
4QWW	A	ACHE: Acetylcholinesterase (snake venom)	Fab410	Bungarus fasciatus	Poison	2,7
2YC1	C	β-mammal toxin Cn2 from Scorpion	9004G	Centruroides noxius	Poison	1,9
5EPM	C	Cm1a: Beta-theraphotoxin from Tarantula	-	Ceratogyrus marshalli	Poison	1,8
4NP4	A	TOXB: Toxin B	bezlotoxumab	Clostridium Difficile	Infection	2,9
5KTE	A	MntH Divalent metal cation transporter	-	Deinococcus radiodurans	-	3,9
2R29⌘	A	Envelope Protein E	-	Dengue virus type 2	Infection	3
3RVV	A	DERF1: Peptidase 1 (group 1 house dust mite allergen)	4C1	Dermatophagoides farinae	Allergy	1,9
4PP1	A	DERP1: Peptidase 1 (Der P 1 house dust mite allergen)	5H8	Dermatophagoides pteronyssinus	Allergy	3
2JEL	P	ptsH: Phosphocarrier protein HPr, histidine containing protein	JEL42	Escherichia coli (strain K12)	Infection	2,5
3RKD ⌘	A	ORF2: Capsid Protein	8C11	Hepatitis E virus genotype 1	Infection	1,9
3VRL	C	HIV Gag protein	A10F9	Human immunodeficiency virus 1	Infection - HIV	3,2
5DHV	M	HIV-1 Rev NTD dimers	-			2,3
5EU7	A	IN: HIV-1 integrase catalytic core	-			2,6
2DD8 *	S	SARS-CoV Virus Spike glycoprotein	m396	Human SARS coronavirus	Infection - SARS	2,3

3BGF ⌘	S		F26G19 Fab			3
4QNP ⌘	A	N1 neuraminidase	CD6	Influenza A (H1N1)	Infection - Flu	2,8
1NCA *⌘	N	N9 Neuraminidase	NC41	Influenza A virus (H11N9)		2,5
1NCB ⌘	N		NC41			2,5
5DO2 ⌘	B	S protein	4C2	MERS (human betacoronavidur 2c EMC/2012)	Infection - MERS	2,4
3GI9	C	ApcT: NA+ dependent AA Transporter	7F11	Methanocaldococcus jannaschii	-	2,5
4CAD	C	RCE1: RAS and A-factor converting enzyme	-	Methanococcus maripaludis	Cancer	2,5
4H88	A	Major prion protein mouse	POM1	Mus musculus	Prion disease	1,9
2YPV	A	fHbp: factor H binding protein (Lipoprotein)	12C1	Neisseria meningitidis serogroup B	Infection - Virulence factor	1,8
5T5F	A	fHbp: factor H binding protein variant B24	Jar5		Infection - Meningitis	3
1FBI	X	Guinea Fowl Lysozyme	F9.13.7	Numida meleagris	-	3
2VXQ	A	PHLPII: Pollen allergen Phl p 2	-	Phleum pratense	Allergy	1,9
4QEX	A	EBA-175: Erythrocyte-binding antigen-175	-	Plasmodium falciparum	Infection - Malaria	4,5
1OB1	C	MSP1: Merozoite Surface Protein 1 (C-term fragment)	-			2,9
4U1G	A	RH5: Reticulocyte binding protein 5	QA1			3,1
2Q8A	A	AMA1: Malaria Apical membrane antigen 1	1F9			2,4
5EZO	A	CyRPA: Cysteine-Rich Protective Antigen	-			3,6
4N9G	C	Computationally Designed RSV-Presenting Epitope Scaffold	17HD9	Respiratory syncytial virus	Infection	2,5
5H35	C	Membrane protein TRIC: trimeric intracellular cation channel orthologue	-	Saccharolobus solfataricus	-	2,6
3KS0	J	heme domain of flavocytochrome b2	B2B4	Saccharomyces cerevisiae	-	2,7
5EII	G	Histone chaperone ASF1	-		-	2,4
3NCY	A	AdiC: arginine agmatine antiporter	-	Salmonella typhimurium	Infection	3,2
1OAZ	A	TRXA: Thioredoxin 1	SPE7	Shigella flexneri	Infection - Shigellosis	2,8
4RGM	S	Enterotoxin type B	20B1	Staphylococcus aureus	Infection	2,7
5HDQ	A	MNTC:ABC transporter substrate-binding protein	305-78-7			1,8
1NSN	S	SNASE: Staphylococcal nuclease complex	N10			2,8
4U6V	A	α-HL: Alpha-hemolysin	MEDI4893			2,6
5JW4 ⌘	A	Clumping factor A	Tefibazumab			3,7
3EFD	K	KcsA (cytoplasmic domain of KcsA) K(+) voltage-dependent channel	-	Streptomyces lividans	-	2,6
2IH3	C	KCSA: Voltage-gated potassium channel	-	Streptomyces lividans	-	1,7
4LQF ⌘*	A	A33R	A2C7	Vaccinia virus	Infection - Smallpox	2,3
4LU5	B		A20G2			2,9
1ZTX⌘	E	West Nile Virus Envelope Protein DIII	E16	West Nile virus	Infection	2,5
4OII	A	NS1: Non-structural protein NS1	22NS1			3
1TQB	A	PRNP: Prion Protein	VRQ14	Ovis aries	Prion Diseases	2,6
2W9E	A	PRNP: Major Prion Protein	ICSM 18	Homo Sapiens	Prion Disease	2,9
3VG9	A	ADORA2A: A2A Adenosine receptor	-		-	2,7
1H0D	C	Angiogenin	26-2F		Cancer	2
4ZFG	A	ANGPT2: Angiopoietin-2	5A12		neovascular age-related macular degeneration	2,3
4ZSO	E	B7-H6 (tumor cell ligand for NKp30)	-		Cancer	2,5

3R1G	B	BACE1: Beta-secretase 1	YW412.8.31		-	2,8
2HFG	R	BAFFR (Tumor necrosis factor receptor superfamily member 13C)	CB3s		-	2,6
5JZ7	A	Beta-nerve growth factor	MEDI578		-	3,4
5JZ7 *	B				-	3,4
5JZ7 *	E				-	3,4
3NH7	A	BMPR1A: Bone morphogenetic protein receptor type-1A Human	AbD1556		-	2,7
3Q3G	E	CD11b: Integrin alpha-M	107		-	2,7
3NFP	I	CD25: IL-2Rα	daclizumab		Organ Transplantation	2,9
4ZFO	F	CD269/BCMA: Tumor necrosis factor receptor superfamily member 17	J22.9-xi		Cancer - Multiple myeloma	1,9
5TLJ_3	X	CD27	M2177, M2191		-	3,5
1YJD⌘	C	CD28: T-cell-specific surface glycoprotein CD28	5.11A1		-	2,7
3RAJ	A	CD38: ADP-Ribosyl cyclase 1	HB7		-	3
5DMI	A	CD40	CHI220		Autoimmunity	3,7
5TZ2⌘	C	CD47	C47B222		Cancer	2,3
5DFV	A	CD81: Tetraspanin-28	-		-	2,8
1QFW	AB	CGB: chorionic gonadotropin, alpha subunit	-		Pregnancy Diagnosis	3,5
5IKC	M	DBCN: Neuronal migration protein doublecortin	6H10		-	2,1
3SKJ	E or F	EphA2: Ephrin type-A receptor 2	1C1		-	2,5
4Y5V	C	EpoR: Erythropoietin receptor human	Diabody 305		Cancer	2,6
5E8D	A	EREG: Proepiregulin	9E5		Cancer - Colon Cancer	2,5
5E8E	LH	F2: Thrombin	-		Platelets aggregation / Thrombic disorders	1,9
4KI5	M	F8: Coagulation factor VIII human	3E6, G99		-	2,5
4WV1	F	FGFR2: Fibroblast growth factor receptor 2	-		-	2,4
4HJ0 *	A	GIPR: Gastric inhibitory polypeptide receptor	Gipg013		-	3
4HJ0	B	GIPR: Gastric inhibitory polypeptide receptor	Gipg013		-	3
5E94	G	GLP-1R: Glucagon-like peptide 1 receptor	Fab 3F52		Diabetes	2
5C7X	A	GM-CSF Granulocyte-macrophage colony-stimulating factor Human	MOR04357		-	3
1S78	B	HER2: Receptor protein-tyrosine kinase erbB-2	pertuzumab		Cancer	3,3
4K3J	A	HGF: Hepatocyte growth factor	Onartuzumab		-	2,8
2R0K *	A	HGFA: Hepatocyte growth factor activator	Fab58		-	3,5
2WUC	A	HGFA: Hepatocyte growth factor activator long chain	FAB40.DELTA TRP		-	2,7
2UZI	R	hras: GTPase Hras	-		Cancer	2
4YPG	C	IFNα2: Interferon α-2	Sifalimumab		-	3
3KR3	D	IGFII: Insulin-like growth factor II human	DX-2647		Cancer	2,2
1ADQ	A	IGG4 Fc region	Rheumatoid factor autoantibody		Autoimmunity - Rheumatoid Arthritis	3,2
4I77	Z	IL-13	lebrikuzumab		-	1,9
2XQB	A	IL-15	DISC0280		-	2,6
5HI4	B	IL17-A	CAT-2000		Autoimmunity	1,8
2VXT	I	IL-18	125-2H		-	1,5

5BVP	I	IL-1β	canakinumab		juvenile idiopathic arthritis (sJIA) or cryopyrin-associated periodic syndrome (CAPS)	2,2
5LQB	A	IL-2	NARA1 - Mimobody		-	2
4DKE	A	IL-34	Fab1.1		-	3
3L5W *	I	Interleukin-13	C836		-	2
3L5X	A	Interleukin-13	H2L6		-	1,9
3DVG	XY	K63-linked ubiquitin	Apu.3A8		-	2,6
3V6O	A	LEPR: Leptin receptor	9F8		Obesity	2
3TT1	A	LeuT: Leucine transporter, Conformation 1	-		-	3,1
3HI6	A	LFA1: Integrin alpha-L	AL-57		-	2,3
4MXV	B	LTA Lymphotoxin-α	-		-	3,2
4HC1	B	MADCAM1: Mucosal addressin cell adhesion molecule 1	10G3		-	2,9
4K9E	C	Mast/stem cell growth factor receptor Kit D4D fragment	Fab19		-	2,7
5TH9	A	MMP9: Matrix metalloproteinase-9	GS-5745		Cancer	3
3BN9	A	MT-SP1: Membrane-type serine protease 1 / ST14: Suppressor of tumorigenicity 14 protein	E2		Cancer	2,2
3NPS *	A	MT-SP1: Membrane-type serine protease 1 / ST14: Suppressor of tumorigenicity 14 protein	S4		-	1,5
3SO3 *	A	MT-SP1: Membrane-type serine protease 1 / ST14: Suppressor of tumorigenicity 14 protein	A11		-	2,1
5F3B	C	Myostatin (Growth/differentiation factor 8)	RK35		-	1,8
3L95	X	NOTCH1: Neurogenic locus notch homolog protein 1	-		-	2,2
5CZV	A	Notch3 NRR: Notch 3 Neurogenic locus notch homolog protein 3	Fab 20350		Cancer - T-cell ALL	3,2
3SQO	A	PCSK9: Proprotein convertase substilisin/kexin type 9	J16		Arteriosclerosis	2,7
5B8C	C	PD1: Programmed cell death protein 1	Pembrolizumab		Cancer	2,5
4QCI	D	PDGF-B: Platelet-derived growth factor subunit B	-		-	2,3
4R9Y	D	PF4: Platelet factor 4 human	KKO		Autoimmune thrombotic disorder	4,1
2FD6	AU	PLAU + PLAUR: Urokinase-type plasminogen activator and receptor	ATN-615		-	1,9
4I18	R	PRLR: Prolactin receptor human	-		-	3,2
5FB8	C	pro-IL16	14.1		autoimmune diseases, ischemic reperfusion injury (IRI), and tissue transplant rejection	2,1
2ZCH	P	PSA: Prostate-specific antigen	8G8F5		Cancer	2,8
5D93	A	QSOX1: Sulfhydryl oxidase 1	-		Cancer - adenocarcinomas	2,3
4IJ3 *	A				-	2,7
1JPS	T	TF: tissue factor (cofactor for the serine protease F.VIIa)	D3h44		Platelets aggregation / Thrombic disorders	1,9
4M7L *	T	TF: tissue factor (cofactor for the serine protease F.VIIa)	10H10		Platelets aggregation / Thrombic disorders	3,4
5KN5	C	TGFα/Epiregulin	LY3016859		-	2,8
4KXZ	A	TGFβ-2 Transforming growth factor β-2	Fresolimumab		-	2,8
4KXZ *	B				-	2,8
4KXZ *	E				-	2,8

3WD5	A	TNFα	adalimumab		Inflammation / Autoimmunity	3,1
2XWT ⌘	C	TSHR: Thyrotropin receptor	K1-70		Autoimmunity - Thyroïd disease	1,9
3G04 *	C	TSHR: Thyrotropin receptor	M22 (thyroid stimulating)		Autoimmunity - Thyroïd disease	2,6
5J13	A	TSLP (Thymic stromal lymphopoietin)	Tezepelumab		Allergy	2,3
4K24	A	UPAR: Urokinase-type plasminogen activator	ATN-658		Cancer	4,5
1FNS	A	vWV: von Willebrand factor	-		Arterial thrombosis	2
2R4R	A	β2 adrenergic receptor	-		-	3,4
5MES	A	MCL1 (leukemia factor), chimeric mouse and human	-	Homo Sapiens Chimeric + Mus Musculus	Cancer	2,2
3CVH	ABC	H-2 class I histocompatibility antigen + beta2 microglobulin + OVA	25-D1.16	MHC Mus musculus + OVA Gallus Gallus	Anti p-MHC antibodies	2,9
1WEJ	F	CYC: Cytochrome C	E8	Equus caballus	-	1,8
3MJ9 ⌘	A	JAML: Junctional adhesion molecule-like	HL4E10		-	3
3UBX	A	CD1d (+ microbulin)	L363		TCR-like antibody	3,1
3R08	E	CD3ε: T-cell surface glycoprotein CD3 epsilon chain	2C11		-	4,1
2ARJ	RQ	CD8α	YTS 105.18		-	2,9
3B9K *	A	CD8α	-	Mus musculus	-	2,8
5C0N	A	FABP4: Fatty acid-binding protein, adipocyte	CA33		Diabetes / Fatty liver disease	3
4YUE	C	IL-2	S4B6		Transplantation / Autoimmunity / Cancer	2,2
3U9P	C	NGAL: Siderocalin (Neutrophil gelatinase-associated lipocalin)	-		Anti-infective	2,8
1KB5	AB	TCRαβ: fusion of light and heavy chains (scFv, clone KB5-C20 from cytotoxic T cells)	Desire-1		-	2,5
2B2X	A	Itga1 or VLA1: integrin alpha 1	AQC2		-	2,2
1MHP *	A	Itga1 or VLA1: integrin alpha 1, residues 169-360	AQC2	Rattus norvegicus	-	2,8
1PKQ	J	MOG: Myelin Oligodendrocyte Glycoprotein	8-18C5		Autoimmunity - Multiple Sclerosis	3
3KJ4	A	NgRI: Reticulon-4 receptor or Nogo-66 receptor	1D9		Spinal Cord Regeneration	3,1

Supplementary Table 2 (refers to Methods, Figure 3 and Figure 4) | Parameters of the shallow learning methods.
During training, unless stated otherwise, the default parameters for each model were used. We note that we did not perform any hyperparameter optimization. Therefore, the performance of these models may further be improved by additional steps of hyperparameter optimization.

Shallow learning model	Parameters used
Naive Bayes (NB)	priors=None, var_smoothing=1e-09
Support Vector Machine (SVM)	penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1000
Random Forest (RF)	n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None
Logistic Regression (LR)	penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None

Supplementary Table 3 (refers to Figure S11): Detail of the VAE architecture layer per layer.
The number of neurons in each layer is shown, with the type of activation function. As a comparison, other architectures with one less layer in both encoder and decoder were tested and yielded less clear delineation of clusters, written as [Encoder dense layer sizes (latent space dimensions used to represent mean and variance and first decoder layer size) Other decoder dense layer sizes]: [220-100-(50-50,40-40,30-30, or 20-20)-100-220]; [220-200-(50-50,40-40,30-30, or 20-20)-200-220]; [220-300-(50-50,40-40,30-30, or 20-20)-300-220]; [220-400-(50-50,40-40,30-30, or 20-20)-400-220]; [220-500-(50-50,40-40,30-30, or 20-20)-500-220]; [220-500-(50-50,40-40,30-30, or 20-20)-500-220] and [220-500-(20-20)-500-220].

Encoder	Decoder
Dense (220) - BatchNorm1d - ReLU	Dense (40) - BatchNorm1d - ReLU
Dense (300) - BatchNorm1d - ReLU	Dense (100) - BatchNorm1d - ReLU
Dense (100) - BatchNorm1d - ReLU	Dense (300) - BatchNorm1d - ReLU
Dense (40) for mean and dense (40) for variance - BatchNorm1d	Dense (220) - Softmax