Rosetta Antibody Design (RAbD): A General Framework for Computational Antibody Design

Jared Adolf-Bryfogle[1,2,3], Oleks Kalyuzhniy[3,4], Michael Kubitz[3], Brian D. Weitzner[5,6] Xiaozhen Hu[3], Yumiko Adachi[4], William R. Schief[3,4], Roland L. Dunbrack[1], Jr.*

[1] Institute for Cancer Research, Fox Chase Cancer Center, 333 Cottman Avenue, Philadelphia PA 19111, USA

[2] Program in Molecular and Cell Biology and Genetics, Drexel University College of Medicine, 245 N. 15th St. Philadelphia, PA 19102, USA

[3] The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla CA 92037, USA

[4] IAVI Neutralizing Antibody Center at TSRI, 10550 North Torrey Pines Road, La Jolla CA 92037, USA

[5] Department of Biochemistry, University of Washington, Seattle, WA 98195, USA

[6] Institute for Protein Design, University of Washington, Seattle, WA 98195, USA

*Correspondence to: Roland.Dunbrack@fccc.edu

Abstract

A structural-bioinformatics-based computational methodology has been developed for the design of antibodies to targets of interest. *RosettaAntibodyDesign (RAbD)* samples the diverse sequence, structure, and binding space of an antibody to an antigen in highly customizable protocols for the design of antibodies in a broad range of applications. The program samples antibody sequences and structures by grafting structures from updated canonical clusters of CDRs (North et al., *J. Mol. Biol.*, 406:228-256, 2011), performs sequence design according to amino acid sequence profiles of each cluster, and samples CDR backbones using a flexible-backbone design protocol incorporating cluster-based CDR constraints. We computationally benchmarked RAbD on a set of ten diverse antibody–antigen complexes, using two novel metrics, the design risk ratio and the antigen risk ratio, for measuring success in computational protein design. These metrics provide a measure of statistical significance typically absent in protein design benchmarking. Risk ratios for all CDRs were above 1.0 for cluster recovery in the presence of the antigen when compared against simulations in the absence of the antigen. We tested RAbD experimentally on both a lambda and kappa antibody–antigen complex, successfully improving their affinities 10 to 50 fold by replacing individual CDRs of the native antibody with new CDR lengths and clusters.

INTRODUCTION

Antibodies are a key component of the adaptive immune system and form the basis of its ability to detect and respond to foreign pathogens through binding of molecular epitopes. Antibodies are increasingly a focus of biomedical research for drug and vaccine development in addition to their numerous applications in biotechnology by private companies, government, and academia[1-7]. Experimentally, antibodies may be discovered and optimized through in vitro phage and yeast display[8,9], screening with large antibody libraries[10-13] and/or affinity maturation through error-prone PCR[14-16]. They may also be derived in vivo through a combination of animal immunization and antibody screening through ELISA or Western blots, and humanization of the animal antibody[17-19].

Although these methods have been successfully applied to create new antibodies, they can take many months to complete and can be prohibitively expensive. In addition, for many targets, these methods may not produce antibodies with desirable properties, because the antigen is difficult to target[20-22] or because the antibody is required to bind to a specific epitope for various functional reasons such as the neutralization of a target pathogen[23], initialization of a downstream signaling cascade[24], or the blocking of a binding protein from being able to engage the site[25]. We believe that computational design methods developed specifically for antibodies can be used in tandem with state-of-the-art experimental methods to save time, money, and increase our ability to design or enhance antibodies to many different targets.

Various computational methods including rational, structure-based design, protein design algorithms, and antibody-specific modeling techniques can aid in the design of antibodies[26-28]. General protein design methods have been applied to affinity maturation[29-31], improving stability[32-34], humanization[35,36], and the design of phage/yeast display libraries[37-40]. Three software programs have been developed specifically for antibody computational design. Maranas and colleagues have developed the *OptCDR*[41] and *OptMAVEn*[42] methods, which sample and combine elements of antibody structure in an effort to assemble antibodies to bind to novel epitopes. *OptCDR* uses clusters of the six CDRs to sample diverse backbone conformations of CDRs in the presence of a fixed antigen position followed by placement of side chains according to sequence preferences within each cluster, a residue-type and rotamer search from a backbone-dependent rotamer library[43], and a CHARMM-based energy function. The method has not been experimentally tested. *OptMAVEn* divides antibody structures in a manner inspired by V(D)J recombination: antibody heavy- and light-chain V regions, CDR3s, and post-CDR3 segments from the MAPS database[44]. *OptMAVEn* has been tested experimentally and was shown to successfully design antibodies against a very hydrophobic heptamer peptide antigen with a repetitive sequence (FYPYPYA), starting from the structure of an existing antibody bound to a dodecamer peptide containing this sequence (PDB 4HOH[45]; only the heptamer has coordinates and was used in the design process)[46].

Recently, Lapidoth et al. presented *AbDesign*[47] that follows a similar methodology to *OptMAVEn*, breaking up antibodies into V regions and CDR3 region by analogy to V(D)J recombination. As implemented in the Rosetta Software Suite, *AbDesign* combinatorially builds antibodies and performs sequence design from position-specific scoring matrices of

aligned antibody sequences of length-based clusters of the V regions and CDR3 regions. Their benchmarking consisted of reporting the Cα RMSD from native for each CDR of the top design for each of 9 antibodies. The program has not yet been tested experimentally.

Taking advantage of the influx of new structures of antibodies in the PDB, we presented a new clustering of all CDR structures in the Protein Data Bank (PDB), updating the Chothia classification developed in the 1980s and 1990s[48-51] with a dihedral angle metric, an affinity-propagation clustering algorithm, and a systematic nomenclature[52], which is now in common use. From this classification, we developed the PyIgClassify database[53], which is updated monthly and contains CDR sequences and cluster identifications for all antibodies in the PDB. PyIgClassify includes identification of species and germline V regions, and is provided as a relational database for use in antibody structure prediction and design.

We hypothesized that the clusters in PyIgClassify could form the core of a knowledge-based approach to antibody design. Using the data from PyIgClassify, our main approach to design is to graft CDRs from well-populated clusters onto the antibody and to sample the sequence and structure space of that CDR according to the observed variation in sequence and structure of that cluster in the database. Our goal was to create a generalized antibody design framework and program that can be applied to numerous types of antibody design projects from affinity maturation to de novo design.

To create a reliable antibody design framework from our structural bioinformatics efforts, we leveraged the Rosetta Software Suite[54], a collaborative research project across many independent labs around the world. Rosetta has been developed and used for a variety of modeling and design tasks, such as loop modeling[55,56], protein–protein docking[57,58], structure refinement[59-62], de novo protein design[63], enzyme design[64-66], and interface design[67-69]. Rosetta provides frameworks for sampling and optimizing the conformations of the backbone and side chains of a protein–protein complex while simultaneously changing the sequence at specified positions in the interface (in this case, primarily in the CDRs) in order to optimize the total energy of the system. The program and methodology we have developed is called *RosettaAntibodyDesign* or RAbD. In this paper, we describe RAbD and both experimental testing and extensive computational benchmarking.

A common method for computational benchmarking of protein design methods is the use of the concept of *sequence recovery*. Sequence recovery tests how the sequences in the final design models match the native sequence, calculated as percent sequence identity for all or the subset of the designable residues. Rosetta's sequence recovery tends to be in the 35–40% range for full design of monomeric proteins[70], since many surface positions are tolerant to amino acid substitution, and the benchmark protocols do not include functional interactions with other proteins, nucleic acids, or ligands.

However, since our antibody design protocol includes potential changes in the overall structure of the CDRs (by sampling different CDR lengths, clusters, and sequences), the standard sequence recovery metric is inadequate for testing computational antibody design. We have therefore expanded the concept of sequence recovery to include recovery of native CDR lengths and clusters. Although antibodies in the PDB are not likely to be the highest affinity possible to a given epitope, they bind strongly enough for crystallography. Thus, maximizing the recovery of CDR lengths, clusters, and sequences is a strategy to optimize sampling and scoring strategies for antibody design.

We have developed a novel way of assessing the statistical significance of these recovery metrics, which may be used in any protein design scenario. We have defined two design metrics – the *design risk ratio* and the *antigen risk ratio*. We define the *design risk ratio* as the ratio of the frequency of native CDR clusters, lengths, or residue identities in the top scoring designs divided by those sampled during the design trajectory. In this way, we can account for uneven sampling of the native structure and sequence during the design process. The *antigen risk ratio* is the ratio of the frequency of the native CDR length, cluster, or residue identities achieved in the top scoring decoys in independent antigen-present and antigen-absent simulations. This metric accounts for any bias Rosetta may have for the native CDRs even in the absence of the antigen, possibly because of favorable framework–CDR or CDR–CDR interactions.

In order to enable repeatable analysis and comparison of native, modeled, and designed antibody structures output by RAbD, we developed a set of antibody-specific *FeatureReporters* and *Features R Scripts* within the Rosetta Features Reporter framework[70-72]. These have enabled comparison of antibody design strategies and benchmarks, were used in the design of the antibodies in this paper, and have aided in the general optimization of the antibody design framework.

Finally, we show results where RAbD and the feature analysis reporters were used to experimentally improve binding affinity of antibodies from two different antibody–antigen complexes.

RESULTS

Antibody Design Framework and Program

We created a generalized framework and application for antibody design within the Rosetta software suite written in C++. This framework enables the tailored design of antibody CDRs, frameworks, and antigens using core components of the *RosettaAntibody* framework as its base[73-75]. As with other Rosetta design protocols, *RosettaAntibodyDesign* depends on a "Monte Carlo plus minimization" (MCM) procedure[76]. This means that at each stage of the simulation, a change in sequence and/or structure is sampled randomly, followed by energy minimization within the Rosetta force field. If the resulting minimized structure (a "decoy") has lower energy than the previous decoy in the protocol, then the new structure is accepted. If the energy of the new design is higher than the previous decoy, the new design is accepted with probability proportional to $\exp(-\Delta E/RT)$ where ΔE is the change in energy. The Monte Carlo algorithm is meant to sample the diverse sequence, structure, and binding space of an antibody-antigen complex (**Fig. 1; Figure S1 in Supporting Information)**.

The protocol begins with the three-dimensional structure of an antibody–antigen complex. This structure may be an experimental structure of an existing antibody in complex with its antigen, a predicted structure of an existing antibody docked computationally to its antigen, or even the best scoring result of low-resolution docking a large number of unrelated antibodies to a desired epitope on the structure of a target antigen as a prelude to de novo design. The variable domains of the input structure must first be renumbered according to the AHo numbering scheme[77], which can be accomplished through the PyIgClassify server[53]. On input into the program, Rosetta

assigns our CDR clusters using the same methodology as PyIgClassify. The *RosettaAntibodyDesign* protocol is then driven by a set of command-line options and a set of design instructions provided as an input file that controls which CDR(s) are designed and how. Details and example command lines and instruction files are provided in the Methods.

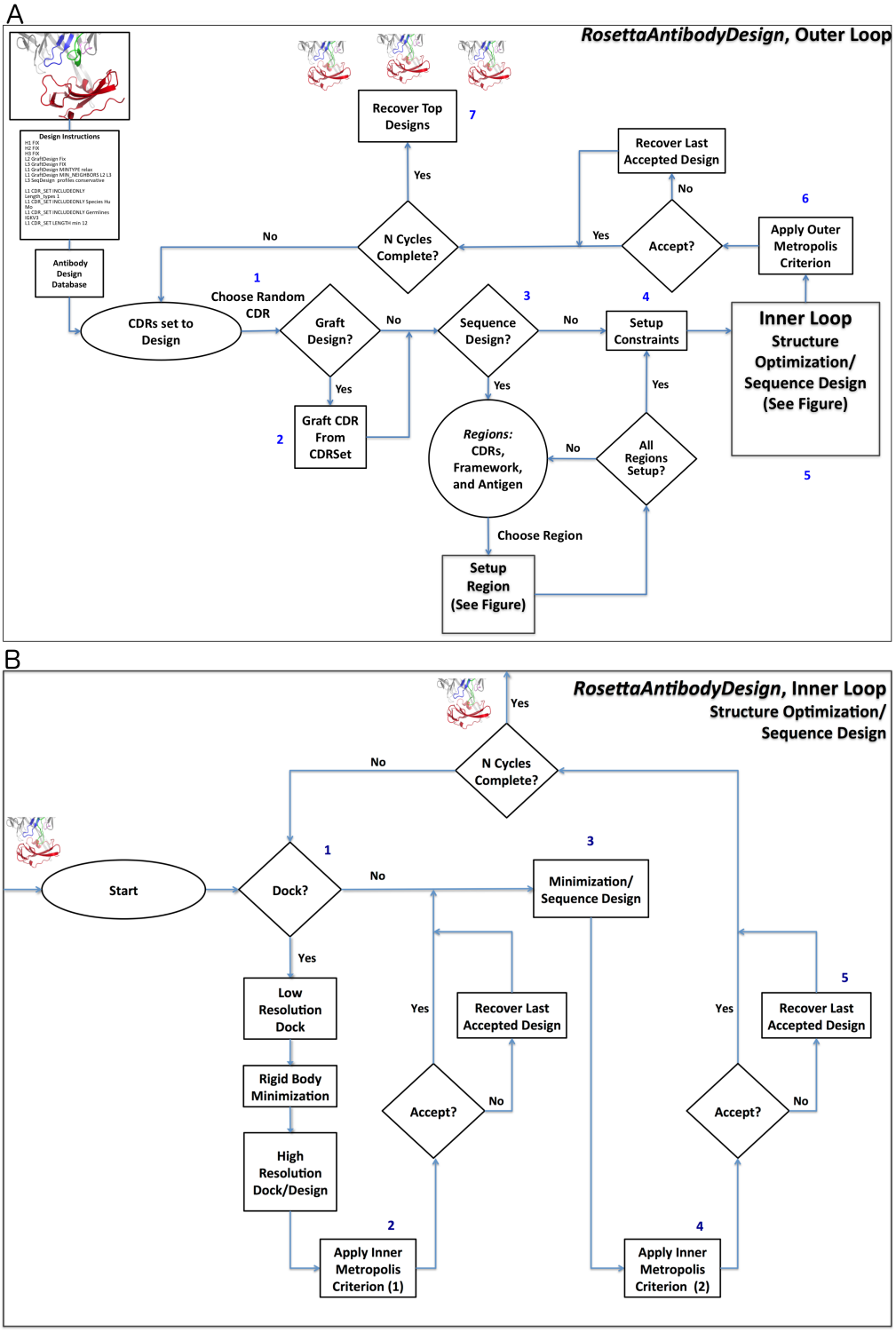

Figure 1. Schematic diagrams of *RosettaAntibodyDesign*. A. the outer loop: The protocol starts by **(1)** Choosing a CDR from those that are set to design [L1, L2, etc.] randomly according to set weights (default is equal weighting) and **(2)** grafting a random structure for that CDR from the CDRSet. **(3)** Regional Sequence Design is then setup for all designable regions and **(4)** structural constraints on the CDRs and SiteConstraints on the antibody-antigen orientation, if any, are set. **(5)** N Inner cycles are then completed, followed by **(6)** the application of the Monte Carlo criterion to either accept or reject the preliminary designs. **(7)** Finally, the lowest energy designs observed are output. **B. the inner loop**: **(1)** The antigen-antibody interface is first optionally optimized by running N cycles of *RosettaDock*[45]. Interface residues set to undergo sequence design will be designed. **(2)** The inner Monte Carlo criterion is then applied. The conformations of the CDR, its stem, and surrounding residues,and CDRs are then optimized according to the instruction file. **(3)** Residues from neighboring regions are designed if enabled (Figure 2 shows this packing/design shell). **(4)** The inner Monte Carlo criterion is then applied again and **(5)** the lowest energy decoy found in the inner loop is returned to the outer loop.

Broadly, the RAbD protocol consists of alternating outer and inner Monte Carlo cycles. Each outer cycle (**Fig. 1A**) consists of randomly choosing a CDR (L1, L2, etc…) from those CDRs set to design, randomly choosing a cluster and then a structure from that cluster from the database according to the input instructions, and grafting that CDR's structure, onto the antibody framework in place of the existing CDR (*GraftDesign*). The program then performs *N* rounds of the inner cycle (**Fig. 1B**), consisting of sequence design (*SeqDesign*), energy minimization, and optional docking. Each inner cycle structurally optimizes the backbone and repacks side chains of the CDR chosen in the outer cycle as well as optional neighbors in order to optimize interactions of the CDR with the antigen and other CDRs **(Fig. 2).** Backbone dihedral angle constraints derived from the cluster data

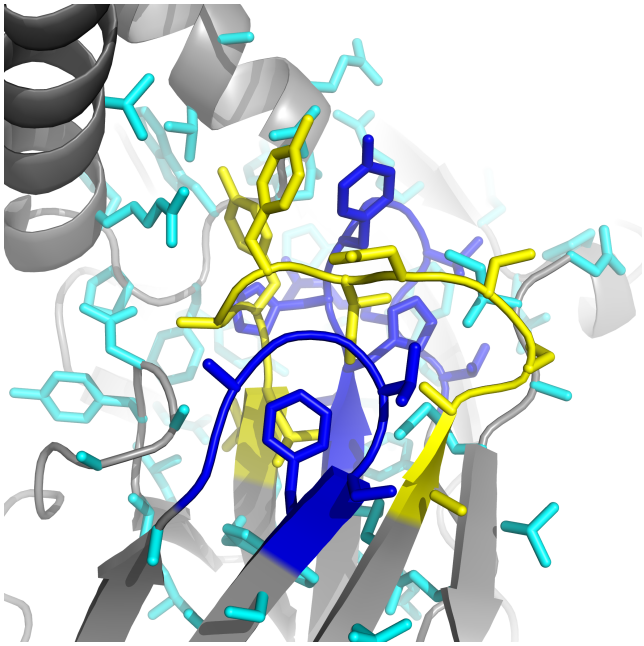

Figure 2: During the inner optimization cycle., a packing shell is created (cyan) around the chosen CDR (in this case, L1 in yellow), and its neighbors (in this case, L3 and the DE loop (L4) in blue). By default, 6 Å is used as the packing shell distance. During the inner loop, all side chains are optimized and amino acid changes are made to any CDRs or regions set to sequence. The chosen CDR and its neighbors additionally undergo backbone optimization during this stage according to the minimization type chosen (min or relax). If the relax minimization type is chosen, backbone movement and sequence design/packing are performed simultaneously (*RelaxedDesign*).

are applied to limit deleterious structural perturbations. Amino acid changes are typically sampled from profiles derived for each CDR cluster in PyIgClassify. Conservative amino acid substitutions (according to the BLOSUM62 substitution matrix) may be performed when too few sequences are available to produce a profile (e.g., for H3). After each inner cycle is completed, the new sequence and structure are accepted according to the Metropolis Monte Carlo criterion. After *N* rounds within the inner cycle, the program returns to the outer cycle, at which point the energy of the resulting design is compared to the previous design in the outer cycle. The new design is accepted or rejected according to the Monte Carlo criterion.

RAbD can be tailored for a variety of design projects and design strategies. Much of this is accomplished through the use of a set of command-line options and the CDR Instruction File. The CDR Instruction File (**Fig. 3**) uses a simple syntax and controls which CDRs are designed with graft and/or sequence design, what lengths and clusters of each CDR will be sampled (the *CDRSet*), and which structural optimizations are used to minimize the score of each design. Each instruction can be set for all of the CDRs using a specific keyword, or they can be set individually. For example, in a redesign project, we may want to design an antibody with a particular CDR that is longer than the existing CDR in order to create new contacts with the antigen that are not present in the starting structure. Alternatively, we may simply want to optimize the sequence of a particular CDR or set of CDRs using the cluster profiles from PyIgClassify. These examples can be accomplished easily through the CDR Instruction File, and this flexibility has been used to design the antibodies described below.

```
#Fix All CDRs
ALL FIX
DE FIX

#Allow only L1 and L4 to design
L1 ALLOW
L4 SEQDESIGN ALLOW

#Use the Relax mintype for design
ALL GRAFTDESIGN MINTYPE RELAX

#Manually set L1 neighbors to L3 and L4
L1 GRAFTDESIGN MIN_NEIGHBORS L3 L4

#Only include CDR clusters for grafting that have more than 10 members (those with profiles)
L1 CDRSET CLUSTER_CUTOFFS 10

#Exclude the CDRs of our starting antibody.
ALL CDRSET EXCLUDE PDBIDs 2J88
```

Figure 3 – Example CDR Instruction file used for successful 2J88 antibody design L14_7. Lines beginning with # are comments and are ignored by the program. Further details are provided in Methods.

A core component of the RAbD protocol is an SQLITE3 antibody design database that houses all structures, CDR-clustering information, species, germline, and sequence profile data used for design. A version is included in the public version of Rosetta that has all CDRs used in the original North–Dunbrack clustering, but up-to-date versions that reflect the current Protein Data Bank (PDB) can also be obtained from PyIgClassify

(http://dunbrack2.fccc.edu/PyIgClassify). In order to improve framework-CDR compatibility in the final designs, λ and κ type antibodies are designed by limiting the resulting *CDRSet* to only those CDRs derived from the same light chain type as the antibody undergoing design. In addition, framework residues are packed through a 6 Å packing shell around the CDR residues being optimized with optional framework design (**Fig. 2**).

For any design scenario specified by the Instruction File, the algorithm is run repeatedly, and the lowest energy models observed in each run of the program is output as a preliminary design or decoy. Hundreds or thousands of decoys may be output by the program; decoy discrimination, analysis, and selection are critical to the experimental success of the final designs.

Computational Benchmarking of RAbD

To develop metrics for recovery of CDR lengths and clusters, we must account for the fact that CDR lengths and clusters are not evenly distributed in nature, the PDB, or in the PyIgClassify database and are not necessarily sampled evenly during RAbD's Monte-Carlo trajectories. The primary antibody design methodology samples a particular CDR sequence and structure from the antibody database by first choosing a random CDR to design from the allowed CDRs, then a random cluster from the list of available clusters for that CDR, and then a random structure/sequence from within that cluster in order to more evenly design new clusters into the antibody of interest. The probability of choosing the native cluster and length during sampling directly influences the statistical significance of the final recovery of the native length and cluster.

To account for this phenomenon, we borrow a concept from statistical epidemiology, the Risk Ratio. The Risk Ratio (RR) is defined as the ratio of two frequencies (or proportions): the frequency of event X in situation A (e.g., disease progression while taking a drug) and the frequency of event X in situation B (e.g., disease progression with no drug treatment). The Risk Ratio is similar to the odds ratio, which is simply the ratio of the odds of X to not-X in situation A and the odds of X to not-X in situation B. However, the interpretation of the odds ratio is often misleading and used incorrectly to inflate a sense of benefit or risk[78]. In standard protein design scenarios, we may define the risk ratio as the frequency of the native sequence (or structure) in the top scoring designs divided by the frequency of the native sequence (or structure) sampled during the protocol.

If we perform design simulations on an existing high-affinity antibody–antigen complex, it is reasonable to suppose that a successful protocol will recover the native CDR lengths, conformations, and sequences of a high-affinity antibody more often than they are sampled. We define the design risk ratio (DRR) for CDR lengths and clusters as the frequency of the native length or cluster in the top scoring designs (the top decoys, one from each run of the program) divided by the frequency that the native length or cluster was sampled during the design simulations. We also define, the antigen risk ratio (ARR), which is the frequency of the native cluster length or CDR in the top scoring designs in the presence of the antigen divided by the frequency of the native in the top scoring designs from independent simulations performed in the absence of the antigen.

A diverse set of seven κ and three λ antibody–antigen complexes were used for the computational benchmarks (**Table 1; Tables S1 and S2**). This set of antibody–antigen complexes includes a diverse set of CDR lengths and clusters, with many of the clusters

commonly found in the PDB. Each crystal structure was minimized into the Rosetta energy function with tight coordinate constraints on both backbone and side-chain regions[61] (see Methods for protocol). In order to enable repeatable benchmarking, the antibody database included with Rosetta by default was used. This database consists of the CDRs used in the original North–Dunbrack clustering paper[52] as opposed to optional up-to-date versions from PyIgClassify (http://dunbrack2.fccc.edu/pyigclassify).

Table 1: Clusters and germlines of benchmark antibodies, as identified by PyIgClassify (See Methods)

PDB (chains)	VL germline	VH germline	L1	L2	L3	H1	H2	H3
1a2y (AB)	Mo_IGKV12-41*02	Mo_IGHV2-7-7*02	L1-11-2	L2-8-1	L3-9-cis7-2	H1-13-1	H2-9-1	H3-10-*
1fe8 (LH)	Mo_IGKV10-97*02	Mo_IGHV2-3*01	L1-11-1	L2-8-1	L3-9-cis7-1	H1-13-1	H2-9-1	H3-9-2
1ic7 (LH)	Mo_IGKV5-43*01	Mo_IGHV3-8*02	L1-11-1	L2-8-1	L3-9-cis7-1	H1-13-1	H2-9-1	H3-7-1
1n8z (AB)	Hu_IGKV1-39*01/ Mo_IGKV6-17*01	Hu_IGHV3-66*02/ Mo_IGHV14-3*02	L1-11-1	L2-8-1	L3-9-cis7-1	H1-13-1	H2-10-1	H3-13-*
1nsn (LH)	Mo_IGKV3-7*01	Mo_IGHV3-2*02	L1-15-1	L2-8-*	L3-9-cis7-1	H1-14-*	H2-9-1	H3-6-1
1tqb (CB)	Mo_IGKV1-135*01	Mo_IGHV9-3-1*01	L1-16-1	L2-8-1	L3-9-cis7-1	H1-13-1	H2-10-1	H3-6-1
3h3b (CC)	Mo_IGKV8-30*01	Mo_IGHV1-42*01	L1-17-1	L2-8-1	L3-9-cis7-1	H1-13-1	H2-10-1	H3-13-*
2xwt (BA)	Hu_IGLV1-51*01	Hu_IGHV5-51*03	L1-13-1	L2-8-2	L3-11-1	H1-13-1	H2-10-1	H3-12-*
3nh7 (LH)	Hu_IGLV3-1*01	Hu_IGHV3-48*03	L1-11-3	L2-8-5	L3-11-cis7-*	H1-13-1	H2-10-3	H3-13-*
4xnq (LH)	Hu_IGLV3-1*01	Hu_IGHV4-4*02	L1-11-3	L2-8-1	L3-9-1	H1-14-*	H2-9-1	H3-16-*

Germlines are as annotated by IMGT[79]. Hu=human; Mo=mouse. IGKV=kappa light chain gene; IGLV=lambda light chain gene; IGHV=heavy chain gene. CDRs that do not cluster or are more than 40 degrees away from the closest centroid (PyIgClassify measured dihedral distance) are represented by "*". CDR (Mo) and framework (Hu) germlines are given for 1n8z respectively.

We tested two types of minimization/design methods: 'min', which alternates between minimization and sequence design, and 'relax', which uses *RelaxedDesign* to perform flexible-backbone sequence design within the FastRelax algorithm. *RelaxedDesign* requires significantly longer computation time. All 5 non-H3 CDRs were allowed to *GraftDesign*, while all CDR sequences, including H3, were allowed to be sequence-designed either preferentially using derived CDR cluster profiles or conservative design where cluster sequence data were sparse. All 5 CDRs that were allowed to *GraftDesign* began each simulation with randomly inserted CDRs from the antibody design database. This benchmark is a rigorous test of the ability of RAbD to recapitulate native lengths and clusters during a de novo antibody design protocol to the same epitope of each antibody.

We define the "%Observed" as the rate at which the native length or cluster is sampled during the design trajectories. The 5 non-H3 CDRs are very different in terms of the diversity of lengths and clusters that are observed in the PDB[52], with L2 and H1 having more than 90% of CDRs in the PDB with a single length and conformation (clusters L2-8-1 and H1-13-1 respectively), while L1, L3, and H2 are diverse in both length and conformation. We ran design simulations for the antibodies in the benchmark set by

sampling the clusters of each CDR evenly (regardless of length) of all clusters represented in the database. Thus, the %Observed of native CDR lengths (**Fig. 4A**) is only 18% for the most length-diverse CDR, L1; followed by L3 and H2 (~40%), H1 (64%), and L2 (88%), which is the least diverse. The %Observed of native CDR clusters is only 6-12% for L1, L3, H1, and H2 and 24% for L2 (**Fig. 4B**).

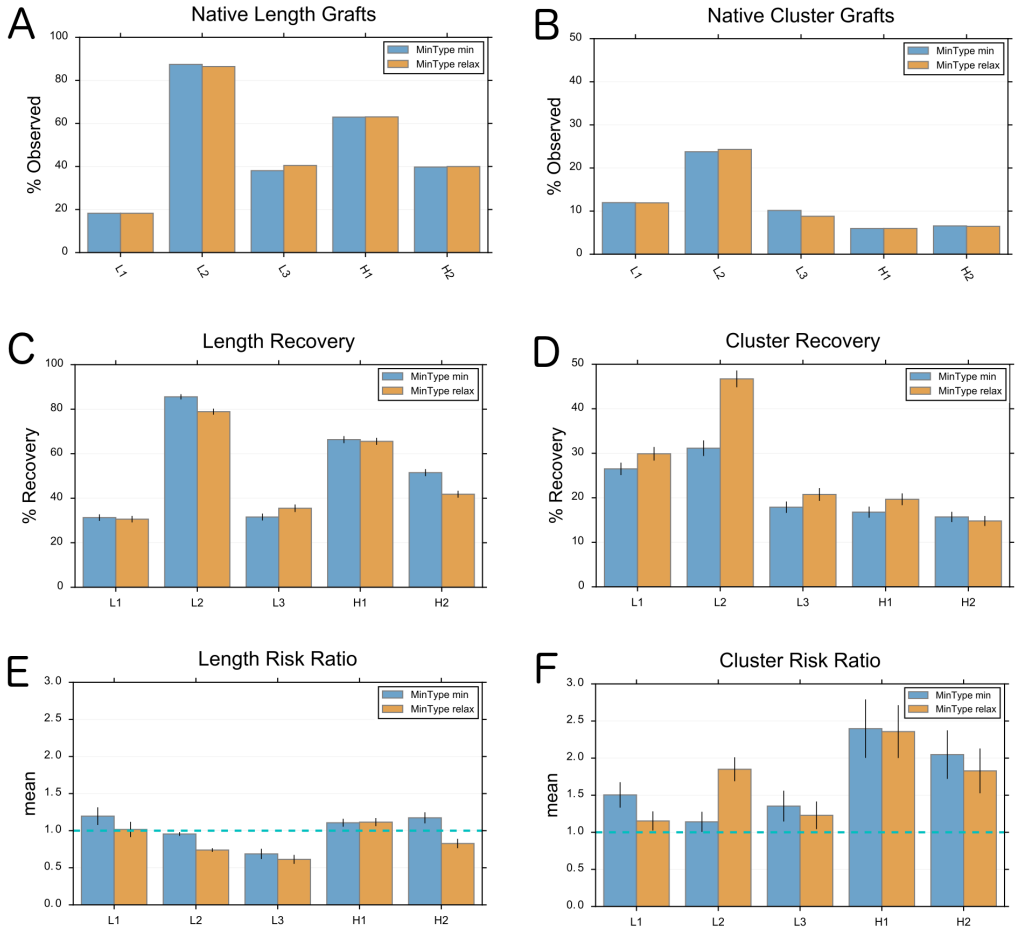

Figure 4: Computational Benchmarking showing recovery metrics on 10 antibodies for two types of minimization/sequence design protocols (min and relax) for each CDR that underwent GraftDesign in the RAbD design protocol. Results are shown for the min protocol (blue bars) and the relax protocol (orange bars). %observed lengths **(A)** and clusters **(B)**; %recovered lengths **(C)** and clusters **(D)**; Design risk ratios for CDR lengths **(E)** and clusters **(F).** Standard deviations are shown for the %recovery as the standard deviation of a binomial distribution. 95% confidence intervals for the Risk Ratio statistics are calculated as described in Methods.

We define the "%Recovery" as the rate at which the native length or cluster is observed in the top scoring designs. For each of the 10 antibodies, we ran 100 design trajectories, each with 150 outer design cycles (**Fig. 1A**) for each experiment (representing a total of 15,000 full design cycles for each antibody) and analyzed the lengths and clusters of the final decoy from each of the 100 Monte Carlo simulations. The %Recovery is then the number of final decoys out of 100 runs that contain the native length (**Fig. 4C**) or cluster (**Fig. 4D**) for any given CDR. The %Recovery of length generally runs parallel with the %Observed with the least length-diverse CDRs (L2 and H1) having higher length recovery

than the others. The highest cluster %Recovery is for L1.

For benchmarking computational antibody design, we want to compare the frequency in which the native cluster/length is observed in the top scoring models (out of all the models constructed in the benchmark experiment) with the frequency with which it was sampled from random draws on the database (out of the total number of draws for that CDR type). The Design Risk Ratio (DRR) is then defined by Equation 1:

$$DRR = \frac{\%Recovery}{\%Observed} \quad (1)$$

The DRR allows us to disentangle the sampling and computational constraints of the method and allows us to determine the statistical significance of our recovery metrics (in the form of 95% confidence intervals; see Methods) over the uneven distribution of our database and sampling methodology. The DRRs for the length of CDRs are highest for L1 and H2 with values of ~1.2. We do not expect high DRRs for L2 and H1 since their length diversity in the PDB is very limited in the first place. The L3 DRR is lower than might be expected because of a narrow range of L3 lengths available for κ antibodies (**Table 1**).

The DRRs for the clusters are much higher. For both protocols, the cluster risk ratios are above 1.0 for all 5 non-H3 CDRs, and over 2.0 for H1 and H2. The results demonstrate the utility of the DRR in accounting for the different levels of diversity in length and cluster across the five CDRs in which *GraftDesign* was enabled and are directly correlated with the diversity in each CDR. It is notable that the risk ratios were high for the highly diverse CDRs (L1, L3, H2) where random sampling results in a lower number of trials in which the native cluster or length is grafted but the energy function is able to distinguish those native clusters – even while designing all five canonical CDRs together. This result may come from both more favorable interactions and higher shape complementarity with the antigen–antibody interface using the native cluster(s), as well as local CDR–CDR interactions, which help to enrich certain lengths and clusters together.

While the recovery and DRR results for the two optimization protocols (min and relax) are quite similar for both groups (except for the L2 DRR, where the relax protocol is significantly better) (**Fig. 4**), the respective structural properties derived from the implemented antibody and interface Feature Reporters (Methods) are quite different. **Figures S2-S6** show that low-energy decoys produced during the benchmarks with *relax*, as opposed to the faster *min* protocol, show better interface ΔG values and total energies, shape complementarity[80], total cross-interface hydrogen bonds, and total buried solvent accessible surface area (ΔSASA) at the interface, while being closer to the structural characteristics of the set of natives. For this comparison, each input native structure was relaxed with 100 output decoys using the same Pareto-optimal method[61] used on the starting antibodies in the benchmark.

There is a possibility that Rosetta scores some CDR structures more favorably than others because of internal interactions within the CDR or interactions with other CDRs or the framework. Some rare clusters may be high in energy or even artifacts of highly engineered antibodies or errors in structure determination. To investigate this, we performed both min and relax protocols without the antigen present in the simulations. We calculated an antigen risk ratio (ARR) from Equation 2 as the ratio of the frequency of the

native length or cluster in the final decoys from the antigen-present simulations and the frequency of the native in the final decoys from the antigen-absent simulations:

$$ARR = \frac{\%Recovery \text{ (with antigen)}}{\%Recovery \text{ (without antigen)}} \quad (2)$$

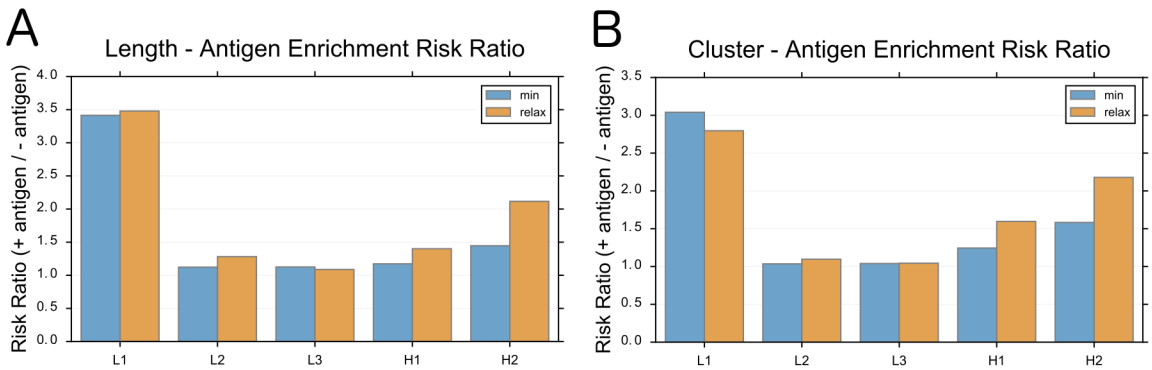

Figure 5: Risk Ratios of benchmarks showing the enrichment in the recovery of native lengths and clusters in the presence of the native antigen compared to simulations performed in its absence. **(A)** Length Risk Ratios from the min (blue bars) and relax (orange bars) design strategies. **(B)** Cluster Risk Ratios. A risk ratio greater than 1.0 indicates enrichment of the native length and cluster in the presence of the antigen over simulations performed in the absence of the antigen.

The length (**Fig. 5A**) and cluster (**Fig. 5B**) antigen risk ratios demonstrate that the native lengths and clusters are enriched particularly for the L1 and H2 CDRs in the presence of the antigen. The results are slightly better for the more computationally expensive relax protocol with the average antigen risk ratios of about 1.5 across all 5 CDRs for relax and 1.4 for the min protocol.

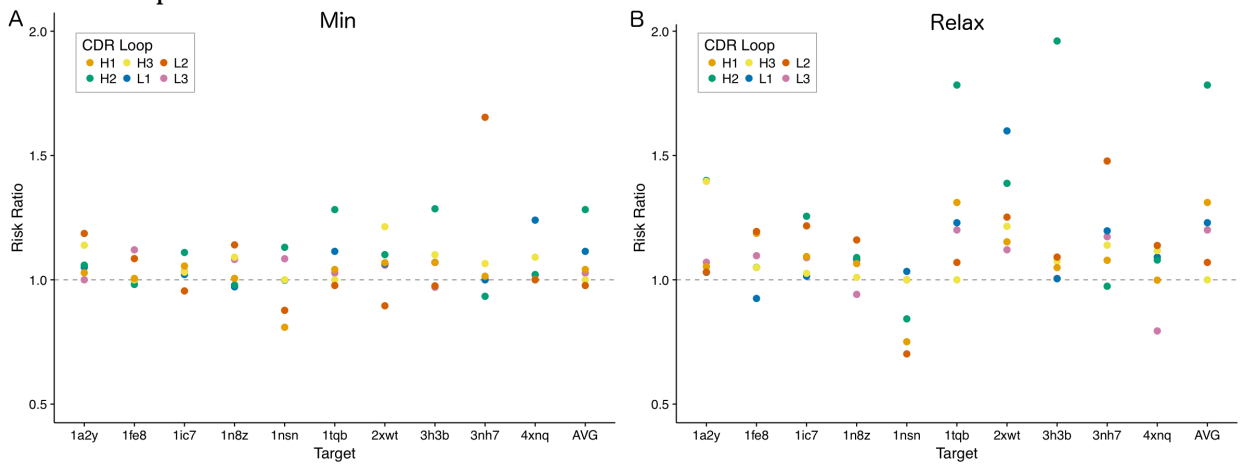

Figure 6. Sequence antigen risk ratios for residues of each antibody in contact with the antigen. Values greater than 1.0 indicate that the native residue types were present in the design simulations in the presence of the antigen more often than they were present in the design simulations in the absence of the antigen. **(A)** Sequence ARRs for the min protocol; **(B)** Sequence ARRs for the relax protocol. Some antibodies did not have contacts between individual CDRs and the antigen, and these CDRs are thus not represented in the figure for those antibodies.

To further isolate the effect of the antigen's presence during the design phase, we performed sequence design on one CDR at a time (6 per target) starting from the native sequence and structure with and without the antigen present using both the min and relax

protocols described above. In each of these twenty-four simulations, we produced 100 models for analysis. **Figure 6** shows the sequence antigen risk ratios, as formulated in equation 2, for antigen-contacting residues (defined by the native complex) in each target split up by CDR. On average, both protocols tend to recover native residues at antibody–antigen contact positions as demonstrated by ARRs > 1.0. The ARRs are better for the more compute-intensive relax protocol.

As a final control, we tested the DRRs for our various sequence design strategies – profiles (cluster profiles), conservative (substitutions derived from BLOSUM62 matrix; see Methods), and basic (all 20 amino acids). As can be seen in **Figures S7 and S8**, the RRs for profiles/basic, profiles/conservative, and conservative/basic are all > 1.0, showing that each of these guided design strategies work to enrich for native amino acids as expected when considering all designed positions (**Figure S7**) or just those positions in contact with the antigen (**Figure S8**).

Experimental Validation

Although computational benchmarking can be extremely useful in optimizing a protocol and its parameters for protein design, the true measure of new protein design methodologies is to test computationally derived sequences experimentally by expressing and purifying the proteins and testing them for desired functionality, including binding affinity and biophysical properties of the designed molecules.

We tested a common scenario for which RAbD was intended – improving the affinity of an existing antigen–antibody complex by grafting new CDRs in place of one or more of the native CDRs. To this end, we tested the ability of the RAbD program to create viable antibody designs that improve binding affinity in two antibody–antigen complexes: an HIV-neutralizing antibody known as CH103 (PDB ID: 4JAN)[81] that binds to the CD4 binding site of HIV gp120, and an antibody that binds to the enzyme hyaluronidase, which is the main allergen in bee venom (PDB ID: 2J88)[82]. These antibodies are not dominated by interactions of H3 with the antigen and use common canonical clusters for the CDRs at the binding interface. Using this knowledge and the general knowledge of CDR length and cluster variability, we designed both L1 and L3 together, or H2 in the CH103 antibody. For the 2J88 antibody, we designed either L1 and the light chain DE loop, or H2. The DE loop is a short loop between strands D and E of the variable domain β sheet (residues 82-89 in AHo numbering). The ability of RAbD to treat both the heavy- and light-chain DE loops as CDRs, which are typically considered framework regions, was added later in program development after the elucidation of the role of the loop in both antigen binding and stabilization, especially in regard to intra-CDR contacts with L1[83]. In light of this, with the L1 design of the 2J88 antibody, we enabled sequence design of this light chain DE loop, which we call L4.

The CDRs selected for design were set to undergo graft- and sequence-sampling with the relax protocol, allowing for new lengths and clusters in the final antibody design, while the framework residues and antigen residues held their starting amino acid identities. The crystal structures 2J88 and 4JAN were first minimized into the Rosetta energy function before being used as starting points for the designs. An instruction file was given for each CDR design strategy (L1 + L4, H2, and L1 + L3) with different algorithms and selection methods used to choose the final designs. Details of these settings, instruction files, and design and selection strategies can be found in Methods.

For 2J88, 30 designs were chosen, expressed, and purified with no detectable aggregation. These 30 were chosen with no visual or manual inspection, and based purely on a ranking of physical characteristics of each decoy determined by the *AntibodyFeatures* reporters for each design strategy and selection characteristic (Methods). Of these 30, 20 showed some degree of binding affinity through an acceptable kinetic sensorgram signal of at least μM binding consisting of L1 loops of length 11, 15, and 17 residues (wild-type: 11) in addition to a single H2 design of 12 residues (wild-type: 9) (**Figure 7A**, **Table S3**).

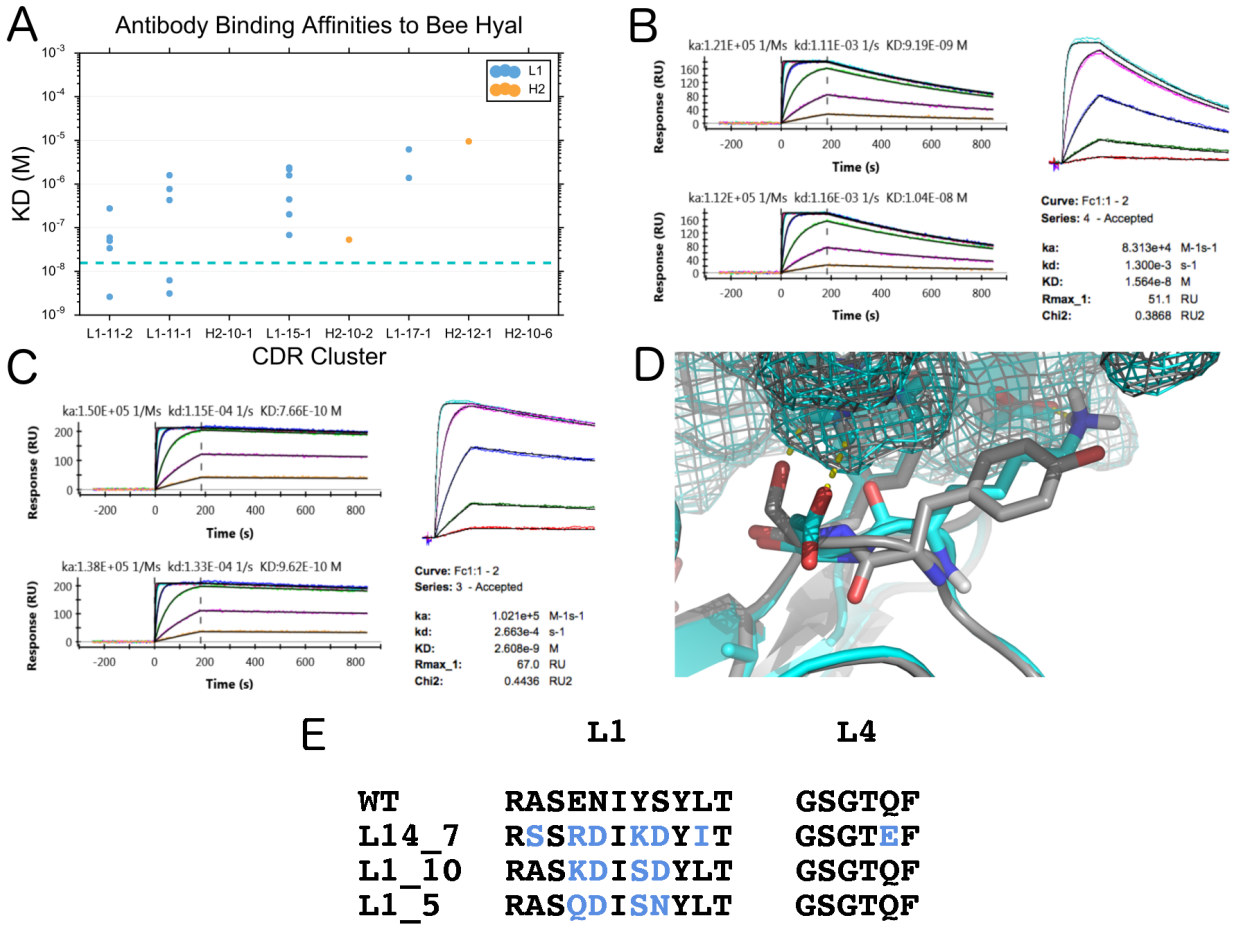

Figure 7: Designed antibodies against bee hyaluronidase. (A) Apparent Binding Affinity (K_D) of expressed and tested antibody designs for bee hyaluronidase (PDB: 2J88), grouped by designed CDR cluster, as determined using Surface Plasmon Resonance (SPR) on a Biacore 4000. The dotted blue line represents the binding affinity of the native antibody on the Biacore machine (1.57×10^{-8} nM). Binding affinity is shown for the 26 designs that had detectable binding affinity (out of 30 tested). The native CDRs are L1-11-2 and H2-9-1. **(B)** Kinetic sensorgrams of WT 2J88 Antibody to Bee Hyaluronidase. Two repeats of *XPR* (left)*; Biacore 4000* (right). **(C)** Kinetic sensorgrams of design L14_7 to Bee Hyaluronidase Two repeats of *XPR* (left)*; Biacore 4000* (right). **(D)** Model of the interface changes in design *L14-7,* with designed L1 cluster L1-11-2 (cyan), superimposed onto the WT antibody from PDB ID 2J88 (gray). **(E)** Designed Sequences vs WT.

Three of these designs had improved binding affinity over the wild-type (WT), which binds at 9.2 nM (**Figure 7B)**, with the best design exhibiting a 12-fold improvement over wild-type with a K_D of 770 pM, as determined by Surface Plasmon Resonance (SPR) on a ProteOn XPR (**Figure 7C**). This design, designated as *L14_7*, had a different L1 cluster (L1-

11-2) than the wild-type (L1-11-1), with six amino acid differences in the L1 sequence, and a single amino acid difference in the L4 loop (**Figure 7D, 7E**), which makes important contacts with the new L1 loop in the design model. Of the L1/L4 design group where docking was enabled, L14_7 had the lowest computational ΔG after filtering out the worst 90% of the designs by total Rosetta energy. The other two designs with better affinity than the native, L1_10 (**Figure S9**) and L1_5 (**Figure S10**) contained the same cluster as the native, but with 4 amino acid changes in L1 out of 11 positions. Kinetic studies of these designs and WT were done on both a Biacore 4000 and an XPR for a total of 3 replicates. Binding affinity was improved against WT for each of these designs in each replicate. Although the rigorous dualspace FastRelax protocol[60] was used in decoy discrimination and selection, the only antibodies to bind better than WT were selected from the non-relaxed group (**Figure S11**).

Thermostability (Tm) of WT and these designs were similar, as measured by Differential Scanning Calorimetry (DSC). While both the WT and designs had two Tm peaks, the major peak of the WT 2J88 antibody was measured at 75.0 °C, while the designs L14_7, L1_10, and L1_5 had similar Tm values of 71.9, 71.5, and 72.1 °C respectively (**Figure S12**).

Mutational analysis was done to delineate hotspot residues at the antibody-antigen interface. Rosetta was used to guide this manual analysis through the use of the PyRosetta Toolkit[84] and FoldIt Standalone[85] GUIs. Residue 7 (Lys) of the L1-11 loop of the L14_7 design was selected for mutation due to its proximity to the antigen (making hydrogen bonds to the antigen in the design model – **Figure 6D**) and favorable Rosetta energy. This position was mutated back to its sequence-aligned WT residue (K38Y). Binding affinity worsened by approximately 3-4 fold as determined by SPR on a Biacore 4000 **(Figure S13)**. The reverse experiment was done on the WT antibody for position 38 (Y38K) and the mutant exhibited improved binding (14.2 nM to 4.3 nM) as expected **(Figure S14)**. Finally, as a proof-of-concept, we improved one of the weaker-binding designs (L1_4), which harbors a very long L1-17-1 loop by 2.5x, through a single S->V mutation at position 36 in the L1 loop (655 nM -> 261 nM) **(Figure S15**).

We chose 27 design variants of the antibody CH103 to express and test for binding to HIV gp120 by using the AntibodyFeatures reporters to rank and select prospective decoys. Of the 27 designs, 21 designs could be purified and tested for binding affinity against a panel of gp120 from different strains of HIV. Of these, 7 designs could bind one or more of the gp120 strains as determined through SPR on a Biacore 4000 (**Figure 8a** and **Table S4**). One of these antibodies, *H2-6*, improved binding affinity to most of the gp120s tested, with a 54-fold improvement to Core-Bal (91 nM to 1.7 nM) and a 40-fold improvement to HXB2 (52 nM to 1.32 nM) (**Figure 8B, 8C, Figure S16**). H2-6 had the least number of buried unsatisfied hydrogen bonds in the interface in the H2 design, profile-only group (Methods). This antibody design had a longer H2 loop (cluster H2-10-6) than the native (cluster H2-9-1), came from the unrelated mouse antibody from PDB 1OAQ[86] in the antibody design database, and is significantly different than the WT CDR (**Figure 8D, 8E**).

We performed mutational analysis on the CH103 designs and WT antibodies using a similar methodology to the mutants created for the 2J88 design project. Based on the sequence alignment of the H2 loops from the H2-6 design and the WT antibody (**Figure 8E**) and structural observation using the Rosetta GUIs, we mutated two hypothesized hotspot residues within the H2 loop at positions 3 and 8 of the designed length-10 CDR in the H2-6

design (AHo numbering 59 and 67 respectively) to the aligned WT residue (Y->F and Y->E respectively). Binding affinity was measured for Core HXB2 and Core Bal using SPR on a ProteON XPR. Notably, the position 67 mutant decreased binding significantly (1.7 nM to 30.9 nM for HXB2; 8.8 nM to 212 nM for Core Bal), while the position 59 mutants had a smaller effect (1.7 nM to 2.2 nM for HXB2; 8.8 nM to 12.8 nM for Core Bal) **(Figure S17)**. The reverse experiment was also done, where the H2-6 residues at the same positions were placed into the WT antibody. This reverse experiment confirmed position 67 as a hotspot residue **(Figure S18)**; Kd for HXB2 improved 60-fold from 138 nM to 2.3 nM, and Kd for Core Bal improved 93 fold from 1.1 μM to 11.7 nM.

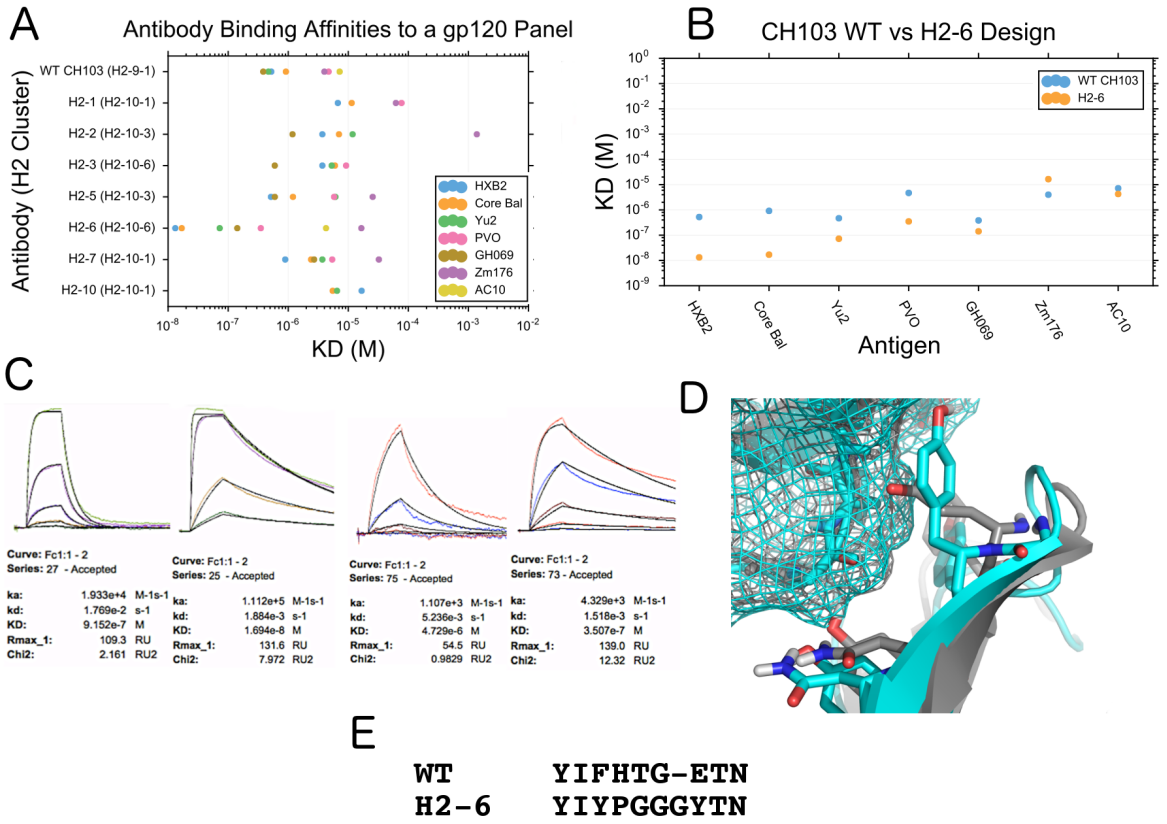

Figure 8: Binding of designed antibodies to HIV gp120. (A) Apparent binding affinity (KD) of WT CH103 antibody and designed antibodies to a panel of gp120 antigens. Here, 30 designs were expressed and tested, where 7 had detectable binding to these gp120s. **(B)** Binding affinity (KD) of the designed antibody, H2-6, versus the wild-type antibody CH103. **(C)** Kinetic sensograms of CH103 WT and design H2-6 to two select GP120s, Core Bal and PVO as determined through a Biacore 4000. **(D)** Model of the interface changes in design *H2-6,* with designed H2 cluster H2-10-1 (cyan), superimposed onto the WT antibody from PDB ID 4JAN (gray) **(E)** Alignment of H2-6 and the WT antibody CH103 from PDB ID 4JAN.

To investigate the role of glycans in CH103 binding, we created glycan knockouts in both the native antibody (**Figure S19**) and the ZM176 strain gp120 (**Figure S20**). Native antibodies do not usually have N-linked glycosylation sites near the paratope and in all cases except AC10, the glycan-knockout antibodies did not affect binding affinity significantly. However, multiple antibody designs were sensitive to the 463 and/or 386 glycans of gp120, which are in structural proximity to the antibody binding site. A single

antibody design that included L1 and L3 CDRs design together was able to bind, but only with the potential 386 glycan knocked out. Meanwhile, two designs bound significantly better to ZM176 when the 386 glycan was knocked out. These glycan knockout studies show the importance of glycan considerations for some antigens and for antibody-design in general.

The results shown here demonstrate that RAbD can be used to successfully improve the binding affinity of antibodies, and that those designs can have different CDR lengths and clusters from the starting antibody.

DISCUSSION

The knowledge-based RosettaAntibodyDesign framework and application was developed to enable reliable, customizable structure-based antibody design for a wide variety of design goals and strategies based on a comprehensive clustering of antibody CDR structures[52]. In order to test the ability of RAbD to produce native-like antibody designs before it was used experimentally, we performed rigorous computational benchmarking using novel recovery metrics, the design risk ratio (DRR) and the antigen risk ratio (ARR), which provided needed statistical significance and improvement over random sampling. The results showed that RAbD was able to enrich for native lengths and clusters – even with the large structural diversity of our underlying antibody design database and flat sampling over CDR clusters, while recovering native-like physical characteristics of the interface and antibody. We applied RAbD to two different antigen-antibody systems where the ability to tailor the program to a specific need and the use of our knowledge-based approach to both antibody design and selection, led to successful experimental designs that improve binding affinity significantly using different CDR lengths and/or clusters.

RAbD is most similar to methods previously developed by two other research groups: *OptCDR* and *OptMAVEn*, developed by Maranas et al., and *AbDesign* developed by Fleishman et al. These authors also present computational benchmarking of their methods, and our benchmarking procedures, metrics, and results can be compared with theirs.

For *OptCDR*, Pantazas and Maranas constructed 100 decoys for 254 antibodies with CDRs borrowed from other antibody structures and used a simple scoring system that penalized steric conflicts of CDR backbone atoms with any atom of the antigen (within the van der Waals distance), favored interactions with a flat score between the sum of van der Waals radii and 8 Å between CDR backbone atoms and atoms of the antigen, and a zero score for longer distances. The native CDR coordinates had better scores than the constructed decoys on average. This is not surprising since almost all of the decoys would have at least one non-native CDR length or cluster, and the antibody as a whole would score worse than the exact native structure, which would have zero clash score and a favorable contact profile. They did not evaluate whether decoys with similar CDR lengths or clusters as the native scored well, as we have done with the length and cluster design risk ratios.

In a second computational experiment, they tested a set of 95 experimentally characterized mutants of a single antibody (anti-VLA1, PDB: 1MHP), 12% of which had improved affinity experimentally[87]. They claim 78% binary total accuracy (Q2) on this set, and a 50% positive predictive value (PPV), which is the percentage of their positive predictions that are true positives (improved affinities). It is impossible to discern a consistent and complete set of evaluative measures typically used in binary prediction

methods (TPR, TNR, PPV, and NPV, balanced accuracy, etc.)[88] from these limited pieces of data. Finally, they performed a sequence design test and found that the native sequence scored better than all decoy sequences for two thirds of 38 test cases, although this does not indicate that the method could find these native sequences from scratch, which is the typical sequence recovery metric in protein design. They did not provide any measures of statistical significance of these results.

OptMAVEn[42] is based on the MAPS database developed by the same authors. For the heavy chain, κ light chain, and λ light chain, MAPS contains separate PDB files for the structures of V regions (through the beginning of CDR3), CDR3 segments, and post-CDR3 segments ("J regions"). If we count unique sequences: for the variable regions, there are 60 heavy, 34 κ, and 21 λ segments in MAPS; for CDR3, there are 413 heavy, 199 κ, and 39 λ sequences; and for the J regions, there are 3 heavy, 4 κ, and 6 λ sequences. By contrast, RAbD grafts CDRs onto the starting framework, which may have already been optimized for stability or other purposes. RAbD can mix CDR1s and CDR2s from different sources, rather than restricting them to a given V region from the PDB, as *OptMAVEn* does. If RAbD uses only the data from our original clustering, there are 226 L1 sequences, 168 L2 sequences, 260 L3 sequences, 265 H1 sequences, 282 H2 sequences, and 279 H3 sequences, which can be used in any combination. In the current PDB, with the same filters, there are approximately 3 times as many sequences for each CDR.

In their computational testing of *OptMAVEn*, Li et al. demonstrated that their grid search over antigen positions and orientations is able to sample (but not rank or score) structures relatively similar to the native structure for 120 antigen-antibody complexes (antigen protein lengths of 4 to 148 amino acids). This is useful to know but does not represent a recovery metric of any sort. They utilized *OptMAVEn* to produce designs for the same benchmark set and were able to produce designed antibodies with lower calculated interaction energies than the native for 42% of the cases, but this does not show that such antibodies would in fact bind better than the native, nor does it show that the native CDR lengths or conformations or sequences were obtained more frequently than one would expect, as our DRR does. For two antibodies, the authors were able to show that they could recover 20% (HIV VRC01) and 35% (Influenza CH65) of mutations from low-affinity antibodies to high-affinity, matured antibodies. By contrast, in our sequence design benchmark, sequence recovery of the native residues in contact with the antigen was about 80% for the min protocol and 57% for the relax protocol.

For their *AbDesign* method, Lapidoth et al. clustered the V regions of antibodies (up to the beginning of CDR3 of each variable domain) purely by the combination of lengths present at CDR1 and CDR2. They clustered CDR3 for each domain by length and RMSD. The input data consisted of 788 heavy-chain domains and 785 κ light-chain domains (no λ chains were included), broadly clustered into 5 V_{κ}, 2 L3, 9 V_H, and 50 H3 conformations.

AbDesign was computationally tested on 9 antibodies[47]. The authors compared features such as shape complementarity, buried surface areas, and interaction scores with the native antibodies. In terms of recovery of the native structure, they calculated Cα RMSDs to native for each of the CDRs of the top scoring design for each of the 9 antibodies. For all of the non-H3 CDRs and 6 of the H3s, the designs had CDR lengths that matched the native antibodies. For 36 of the 45 non-H3 CDRs (80%), the Cα RMSDs were better than 1.0 Å. It is difficult to assess the significance of these results, because the source database for

sampling in *AbDesign* is dominated by the same CDR lengths and clusters found in the native antibodies in the benchmark. We searched PyIgClassify for the 9 antibodies in this benchmark for their clusters according to our nomenclature. Since H3 does not cluster well beyond length 8, we report only the lengths of H3. The results are as follows (cluster and representation in the 9 antibodies in their benchmark set): L1-11-1 (5/9); L1-11-2 (3/9); L1-12-1 (1/9); L1-16-1 (1/9); L2-8-1 (9/9); L3-9-cis7-1 (7/9); L3-9-cis7-2 (1/9); L3-9-1 (1/9); H1-13-1 (8/9); H1-13-3 (1/9); H2-10-1 (7/9); H2-10-3 (1/9); H2-9-1 (1/9); H3-9 (1/9); H3-10 (4/9); H3-11 (2/9); H3-12 (2/9). L1-11-1 and L1-11-2 are very similar to each other (<0.5 Å RMSD). As it turns out, the 4 non-H3 CDRs with the largest RMSDs to native (>1.8 Å) are those with less common clusters or lengths: L1-12-1 (1IQD, 1.85 Å RMSD), L3-9-1 (1IQD, 2.12 Å RMSD), H1-13-3 (2CMR, 2.04 Å RMSD), and H2-10-3 (1P2C, 1.90 Å RMSD). This indicates that *AbDesign* may be dominated by its sampling database in a way that makes the benchmarking data difficult to interpret. The design risk ratio we developed in this work solves this problem by demonstrating the increase in recovery over the sampling rate of any particular conformation in the database. Similarly, the antigen risk ratio demonstrates that the sampling and scoring is able to choose native-like CDRs when the antigen is present in the simulations. Finally, Lapidoth et al. achieved a sequence identity of 32% for residues in the binding site of the antibodies in their benchmark, compared to RAbD's values of 80% for our min protocol and 57% for the relax protocol in our benchmark.

A major challenge moving forward, especially in regard to true de novo design, is the difficulty in effective sampling and design of the H3 loop, owing to its extreme variability and lack of clustering. To aid in this, H3-specific design strategies in the program can include up-weighting H3 graft sampling from other CDRs, restricting the use of H3 loops in the *CDRSet* to kinked-only (See Methods)[89,90], and reducing the search space to only short loop lengths that cluster well; however, much more work is needed to benchmark and test H3-specific design. Recently, we have shown that H3-like loops can be found in non-antibody proteins[89]. These loop structures could be used to supplement existing H3 structures in our antibody database as additional templates for H3 *GraftDesign*. The specific design of antibody H3 loops will be a major challenge in the next phase of antibody design methodology development, but using the RAbD framework and new methods for antibody design benchmarking outlined here should aid in this challenge.

These promising computational and experimental results show that RAbD is able to design antibodies with similar features to the native antibodies and antibodies with improved affinity. It can be used for a variety of antibody design tasks through the use of its highly customizable interface. RAbD represents a generalized framework and program for antibody design and makes many antibody design projects feasible that are either difficult or prohibitive using historical, traditional means, making computational antibody design a tangible reality.

Methods

1. Experimental Methods

Transection and Expression

All antibody designs were expressed as IgGs in 293F cells using the pFUSE vectors (pFUSEss-CHIg-hG1 (human heavy), pFUSEss-CLIg-hL2 (human lambda), and pFUSEss-

CLIg-hk (human kappa)). Bee Hyaluronidase and gp120 antigens were expressed in 293F cells using pHLsec vectors. Opti-Mem media and FreeStyle293 Expression media were first warmed to 37 °C. 293F cells were checked for viability at 95% and at a concentration greater than or equal to $2.4x10^6$ cells/ml. 6 mls of OptiMem were mixed with 125 µg of heavy chain DNA, and 125 µg of light chain DNA in one 15 ml conical tube, and 250 µg of fectin in the other. After a five minute incubation, the DNA tube was poured into the fectin tube and was left to incubate for 21 minutes. The 293F cells were then diluted to $1.2x10^6$ cells/ml, added to a 500 ml shaker flask, and the fectin/DNA mixture was added to the flask. The 500 ml flask was incubated in a 37 °C, 8.0% CO_2, 80% humidity shaking incubator for four days. The supernatant was harvested on the fifth day using 500 ml centrifuge tubes and spinning for 20 minutes at 4000 rpm. The supernatant was then filtered in a 500 ml filter unit.

Purification

Antibodies were purified by first using 1 ml of GE rProtein A Sepharose Fast Flow resin in a chromatography column, and then washing with 10 ml dH_2O and 10 ml PBS. The antibody supernatant was poured onto the column and then washed with 10 ml of PDB, followed by 10 ml of 0.5 M NaCl in PDB, and then another 10 ml of PBS after all supernatant had passed through the column. The antibody was then eluted with 6 ml of Thermo Scientific IgG Elution Buffer into a 50 ml conical tube of 0.5 ml, 1M Tris-HCl. The eluted antibody was then placed into a Slide-A-Lyzer cassette and dialyzed against PBS with three changes. After dialysis, the antibody solution was filtered using a 0.22 micron syringe filer and the OD was checked to obtain the final concentration of antibody.

Binding assays.

Kinetics and affinity of antibody-antigen interactions were determined on a Biacore 4000 (GE Healthcare) using Series S Sensor Chip CM5 (BR-1005-30, GE Healthcare) and 1x HBS-EP+ pH 7.4 running buffer (20x stock from Teknova, Cat. No H8022) supplemented with BSA at 1 mg/ml. We followed Human Antibody Capture Kit instructions (Cat. No BR-1008-39 from GE Healthcare) to prepare chip surface for ligands capture. In a typical experiment about 9000 RU of capture antibody was amine-coupled in appropriate flow cells of CM5 Chip. 3M Magnesium Chloride was used as a regeneration solution with 180 seconds contact time and injected once per each cycle. Raw sensorgrams were analyzed using Evaluation software (GE Healthcare), double referencing, Equilibrium or Kinetic with Langmuir model or both where applicable. Analyte concentrations were measured on NanoDrop 2000c Spectrophotometer using Absorption signal at 280 nm.

Antibody-Antigen binding kinetics were confirmed on a ProteOn XPR36 (Bio-Rad) using GLC Sensor Chip (Bio-Rad) and 1x HBS-EP+ pH 7.4 running buffer (20x stock from Teknova, Cat. No H8022) supplemented with BSA at 1mg/ml. We followed Human Antibody Capture Kit instructions (Cat. No BR-1008-39 from GE) to prepare chip surface for ligand capture. In a typical experiment, about 6000 RU of capture antibody was amine-coupled in all 6 flow cells of GLC Chip. 3M Magnesium Chloride was our regeneration solution with 180 seconds contact time and injected four times per each cycle. Raw sensorgrams were analyzed using ProteOn Manager software (Bio-Rad), interspot and column double referencing, Equilibrium or Kinetic with Langmuir model or both where

applicable. Analyte concentrations were measured on NanoDrop 2000c Spectrophotometer using Absorption signal at 280 nm.

Thermostability assays

Differential scanning calorimetry (DSC) experiments were performed on a MicroCal VP-Capillary differential scanning calorimeter (Malvern Instruments). The HEPES buffered saline (HBS) buffer was used for baseline scans and the protein samples were diluted into HBS buffer to adjust to 0.6 mg/ml. The system was set to equilibrate at 20 °C for 15 min and then heat up until a temperature of 125 °C was reached at a scan rate of 90 °C/h. Buffer correction, normalization, and baseline subtraction were applied during data analysis using Origin 7.0 software. The non-two-state model was used for data fitting.

2. *RosettaAntibodyDesign* Program

The RAbD protocol consists of two major steps after identification of the CDR clusters in the starting antibody structure using the dihedral angle distance metric and the same methodology as PyIgClassify[53]. The outer loop consists of: (1) choosing a CDR from those CDRs set to design (which can be weighted to up-sample or down-sample certain CDRs); (2) optionally performing graft-based, whole-CDR sampling (*GraftDesign*) using a randomly chosen CDR cluster and then a randomly chosen structure from the set of structures defined for that CDR in the CDR Instruction File (*CDRSet)* (**Fig. 1A**); and (3) performing *N* inner cycles (**Fig. 1B**).

There are three simple algorithms that control how the CDR structure is chosen from the database during the *GraftDesign* stage. The default is to choose a CDR cluster from the list of available clusters and then choose a structure from the list (*even_cluster_mc*). One can also choose a structure from all available structures, which then samples according to the prevalence of that length and cluster of the PDB (*gen_mc*), or evenly sample the length and cluster by first choosing a length from the available lengths, then a cluster from the clusters available in that length, and finally a structure from that cluster (*even_length_cluster_mc*). Finally, when designing a single CDR, the *deterministic_graft* algorithm can be used to graft every structure available. Availability (*CDRSet*) of lengths, clusters, and particular structures can be controlled through the CDR Instruction File as described below. These algorithms can be useful in tailoring design to specific situations.

The inner loop consists of applying a minimization type, optional docking, and sequence design (*SeqDesign*), preferentially using CDR cluster-based profiles when clusters are large enough; when they are not, a conservative design strategy is based on profiles derived from the BLOSUM62[91] matrix is used. Each cycle structurally optimizes the CDR chosen in the outer cycle and optionally its neighbors in order to optimize CDR-CDR interactions with CDR-cluster derived circular harmonic constraints on all ϕ and ψ dihedral angles. During *all* side-chain optimization (packing) and sequence design calls (including during docking), a packing shell is created within 6 Å (by default) of any residue included in this minimization. If any of the residues in the packing shell or the CDRs are set to Sequence Design, they will be designed, and all side chains within this shell will be optimized, including CDRs, the framework, and the antigen (**Fig. 2**). A Rosetta resfile can be used to further limit design to only specific residues a user is interested in.

Most Rosetta protocols utilize Monte Carlo + minimization algorithms to optimize sequence and structure effectively. By allowing occasional increases in energy, we enable

structures to overcome energy barriers to escape local energy wells in order to drive the energy down further. In order to traverse the energy landscape more effectively, a separate Monte Carlo object controls the progression of the preliminary designs for both the outer and inner components of the algorithm.

RAbD can be tailored for a variety of design projects and design strategies. Much of this is accomplished through the use of the CDR Instruction File, an abundant set of command-line options, and object-oriented code design, which enables RosettaScript-able[92] framework components.

Antibody Design Database

RosettaAntibodyDesign uses an SQLITE3 database to house all antibody and CDR data needed for the program, including full structural coordinates of CDRs, CDR length, cluster, species, and germline identifications, as well as CDR cluster sequence profile data. There are two versions of the database that can be used by RAbD – a database included with Rosetta comprising all antibodies and CDRs included in the original North/Dunbrack clustering[52], and a database that can be obtained from PyIgClassify[53], which is updated once a month and reflects data from current PDB.

While the North/Dunbrack clustering data is culled according to the original clustering (by resolution, R-factor, and non-redundancy), the up-to-date database is culled in a similar fashion, consisting of only non-redundant CDR data at a 2.8 Å resolution and 0.3 R factor cutoff. In order to cull for non-redundancy in the CDR loop, the CDR is selected in the order of: highest Resolution --> lowest R factor --> lowest normalized distance to the cluster centroid. CDR cluster outliers are then removed as described in the Supplemental Methods. These databases are used in all aspects of the antibody design algorithm, including the *GraftDesign* step, which uses the raw coordinates in the database and the *SeqDesign* step, which uses an additional table for CDR cluster profiles (residue probabilities at each position) created from the culled data. Broadly, the CDR Instruction File helps enables additional culling during the *GraftDesign* step, controlling which lengths, clusters, species, germlines, and structures, are used or left out.

DE Loop

The span of framework residues with AHo numbering 82-89 comprise what is commonly referred to as the "DE loop". These variable residues form a loop physically in contact with CDR1 and are occasionally observed making significant contacts with antigen[52,83]. In *RosettaAntibodyDesign*, we denote the DE loop region as L4 and H4 for the light and heavy chains respectively. The typical kappa L4 is different in sequence and conformation than the typical lambda L4, so that lambda L4s should be used with lambda L1 CDRs and frameworks[83] and kappa L4s should be used with kappa L1 CDRs and frameworks. Both loops can be considered CDRs in the application and can be specified just as any other CDR except for the *GraftDesign* stage.

Sequence Design and Side-chain Optimization

In Rosetta, the optimization of side-chain conformations is referred to as packing (or repacking). Packing in Rosetta is stochastic for speed, and each packing round consists of traversing the set of residues to be optimized randomly until no residues are left in the pool and selecting the best rotamer of all rotamers defined for the given residue[63].

Design is accomplished in the packing algorithm by sampling all rotamers (using the 2010 Dunbrack Rotamer Library[93]) of a specified set of residue types allowed at a given position. Control of which residues are packed/designed, which set of rotamers are used, and what residue types are allowed in the design pool for each position can be controlled through an object called the *TaskFactory*.

In general, we use a probability distribution of residue types for each CDR position, embedded in the antibody design database. In the RAbD framework, it is how we sample from the profile of a given CDR cluster type. Each time packing is applied, a residue type for each position is chosen based on distributions from the antibody design database. If this residue is different from the starting residue, it is added to the design types. This process can be performed repeatedly to increase the sampling according to the distributions; residues types with probability of zero can be sampled at some given probability to increase diversity. This methodology helps to maintain the residue profile of a given CDR cluster. This methodology (a *TaskOperation* called the *ResidueProbDesignOperation* if needed for scripting purposes) is in use in the Antibody Design framework by default if enough statistics for that CDR cluster are present.

Alternatively, we use a set of conservative mutations as design types for each specified position The conservative mutations for each residue type are composed of the substitutions for each residue which score >= 0 in one of the BLOSUM matrices[91]. All BLOSUM matrices can be used for conservative design, and are specified through the use of an option. The numbers of the matrix (such as BLOSUM62) indicate the sequence similarity cutoffs used to derive the BLOSUM matrices, with higher numbers being a more conservative set of mutations. By default, the conservative mutations from the BLOSUM62 matrix are used to strike a balance between variability and conservation. This methodology, incorporated as the *ConservativeDesignOperation TaskOperation*, is the default fallback sequence design strategy but can be used generally instead of profile-based sequence design.

By default we disable design for prolines, cysteine residues involved in disulfide bonds, and the H3 kink-determining stem region (first 2 and last 3 residues of H3)[89,90] in order to limit large, unproductive perturbations of the CDR loops from disruptive sequence changes.

Users may further disallow amino acid types for all positions through a command-line option, for specific CDRs through the CDR instruction file, and for specific positions through the use of the Rosetta *resfile* format. The resfile can also be used to disable specific positions from design or packing altogether.

Within the protocol, both antibody-antigen interface residues and neighbor residues **(Figure 2)** that are computed for side-chain packing and design are updated on-the-fly before each packing/design step. This allows the algorithm to continually adapt to the changing environment and is accomplished through Rosetta's graph-based neighbor detection algorithms and careful coding using the *TaskFactory* machinery.

In order to achieve flexible-backbone design and environmental adaptation of the packing/design algorithm as described above, we updated the *FastRelax* algorithm[59] to enable sequence design during backbone and side-chain optimization using custom *TaskFactories*. These changes to *FastRelax*, which we call *RelaxedDesign*, were used in the optimization step of Jacobs et al to general success[94].

Grafting

In order to sample whole structures of CDR conformations from our design database, a way to graft them onto a given antibody was needed. We required an extremely quick algorithm to enable large conformational space to be sampled in a single run of the program, as well as accurate enough to minimally perturb the CDR region without leaving breaks in the structure or non-realistic peptide bond lengths and angles.

We started with an updated version of a grafting algorithm created for the Anchored Design Protocol[95]. This graft algorithm harnesses the Cyclic Coordinate Descent (CCD) algorithm of Canutescu and Dunbrack to insert one region of a protein into another[96]. CCD is an algorithm originally created for loop modeling where a robotics-inspired algorithm iteratively calculates specific changes in ϕ/ψ angles needed to 'close' a loop during loop modeling. Rosetta uses CCD internally for many of its loop modeling protocols[55,56] as well as to generally close a region of a protein where a break in the peptide-bond connection has occurred. The Anchored Graft protocol originally used for Anchored Design (now refactored as the *AnchoredGraftMover*) uses residues on either end of a region to be inserted as the CCD solution-space, essentially creating a large 'arm', where all residues of the arm are held fixed in dihedral space. The algorithm iterates through residue perturbation using a *SmallMover* (which randomly changes dihedral angles within some range), loop closure with CCD as described above, and energy minimization of the terminal ends.

The arm is held fixed in dihedral space during the entirety of the algorithm and the residues of the pose are turned into a centroid representation to increase the speed of the protocol. The centroid representation of a protein structure can be thought of as a coarse-grained model where the side chains are essentially replaced with different sized balls. The centroid model is the main representation used by low-resolution loop modeling and low-resolution docking within Rosetta.

The run time of the initial grafting protocol was significantly higher than what we needed for the program. We created the *CCDEndsGraftMover* to rectify this, which uses the same general algorithm as the *AnchoredGraftMover*, with different regions used for minimization and CCD. The *CCDEndsGraftMover* uses three extra residues on either end of the region to be inserted to first superimpose the CDR onto the given antibody. Those residues are then removed from the insert and three residues of the framework on either end are used as CCD arms to close the loop in addition to single residues on either end of the loop. Both grafting classes use a set of smooth centroid scoring statistics (F. DiMaio, unpublished), which results in a much smoother energy distribution of final models and better structure optimization.

The algorithm in the antibody design framework uses a combination of these graft algorithms and attempts to use the *CCDEndsGraftMover* first, using the *AnchoredGraftMover*, for any loop that is not closable using the *CCDEndsGraftMover* algorithm (mainly due to differences in the distance between terminal ends of the CDRs, such as L1). A graft is considered closed if the peptide bond C-N distance is less than 1.5 Å and both the Cα-C-N and C-N-Cα angles are less then 15 degrees away from the ideal min and max values determined by Berkholz et al.[97] (114.5°, 119.5° and 120°, 126° respectively). Most CDR grafts can be completed in less than a second.

When both terminal ends are closed during the protocol, we continue the design

protocol. This graft adaptation algorithm closes 100% of CDR grafts, and is now also used for grafting within the antibody.cc application of RosettaAntibody[75].

Optional Dock/Design

Integrated sampling of the antibody-antigen orientation during design uses the underlying framework and docking algorithms of *RosettaDock*[45,98-101]. A 'dock cycle' consists of a low-resolution docking step, side-chain repacking of the interface residues (defined as residues of the antibody or antigen that are within the set interface distance of each other (8 Å default), minimization of the rigid body orientation (the *'jump'* in Rosetta parlance*)* between the antigen and the antibody, and a shortened high-resolution dock consisting of 3 outer cycles and 10 inner cycles as opposed to 4 and 45, which is used for a full *RosettaDock* high-resolution run. During high-resolution docking, the current interface side chains are optimized, while any CDRs or specific residues set to design are designed. In this way, sequence design and antibody-antigen orientation optimization are coupled in the same vein as sequence design is accomplished during CDR structural optimization using *RelaxedDesign*.

Dihedral, Epitope, and Paratope Constraints

Several constraint types are used by the Antibody Design framework to limit unproductive structural perturbations of the CDR regions and the relative orientation of the antibody-antigen interface while docking in the program. There are many constraint types with associated function types implemented in Rosetta. These constraints are added to the structure and are evaluated via terms added to the Rosetta energy function. The Rosetta energy minimizer (which optimizes the conformation of the structure by finding the local energy minimum) can use these constraints to find optimal values that help to satisfy all the energy terms including the constraints. Within the Antibody Design framework, the weight of these constraints can be set from command-line options, in addition to whether these constraints are used throughout the protocol (where they also act as structural filters) or only in certain situations like minimization or docking (where they act only to guide the structure to an optimum conformation that satisfies the constraints).

The set of general Antibody constraint movers that were implemented consist of the *CDRDihedralConstraintMover, ParatopeSiteConstraintMover,* and the *ParatopeEpitopeSiteConstraintMover.* These Movers (a mover applies some change to a Pose or structure[102]) can be fine-tuned for specific design strategies using a number of user-accessible options and RosettaScripts[92].

The *CDRDihedralConstraintMover* places Circular Harmonic constraints on each ϕ and ψ dihedral angle of a given CDR as cluster-specific and general-use constraints. The equation for the Circular Harmonic constraint is as follows where x_0 is the starting dihedral angle, x is the changed dihedral angle, and σ is the standard deviation of x:

$$f(x) = \left(\frac{NearestAngleRadians(x, x_0) - x_0}{\sigma} \right)^2$$

Constraints are added to help Rosetta keep a particular loop structure during any backbone optimization. Dihedral constraints are used instead of coordinate constraints (which try to keep each atom at a particular Cartesian coordinate) in order to allow more natural, hinge-like motion of the CDR loops. Users of the protocol who wish to design antibodies without these constraints can set the weight of the dihedral constraint to zero via a command-line option.

The cluster-specific constraints have the value of x_0 and standard deviation for each backbone ϕ and ψ dihedral angle at the angle mean and standard deviation of the members of the cluster. These constraints are output by PyIgClassify using a high-quality set of non-redundant data, by all 3 CDRs per chain (meaning all three CDRs of each antibody in the database are concatenated and aligned to determine redundancy). The default behavior of the *CDRDihedralConstraintMover* is to add these constraints for a particular CDR only if there are enough members in the cluster to have reliable data. This principle of a data cutoff for our knowledge-based elements in the Antibody Design framework is a common theme. If data are scarce, then general dihedral constraints are added, with means at the current angles and a standard deviation that was originally compiled by taking the mean of the standard deviations of all CDR clusters. These angles can be set via command-line options. By default, we use these general dihedral constraints for H3, since it does not cluster well.

For the cluster-specific constraints (and other places in the protocol), we generally filter out outliers in the data as described below. This can be turned off through the use of an option that will load a different set of constraints compiled with structures that are not filtered for outliers. This can be useful if using outliers elsewhere in the protocol.

SiteConstraints are a set of atom-pair constraints that evaluate whether a residue interacts with some other chain or region - roughly, that it is (or is not) in a binding site. More specifically, if we have a *SiteConstraint* on a particular residue, that *SiteConstraint* consists of a set of distance constraints on the Cα atom from that residue to the Cα atom of all other residues in a set, typically the set being specific residues on another chain or chains. After each constraint is evaluated, *only* the constraint giving the lowest score is used as the *SiteConstraint* energy for that residue. These *SiteConstraints* use a Flat Harmonic function by default: $f(x)=k(x - x_0)^2$ if $|(x - x_0)| > T$ with T being the tolerance value; $f(x) = 0$ if $|(x - x_0)| <= T$. Values of the standard deviation are set at 1 Å, while the tolerance is set at the interface distance of the protocol (8 Å default), which means that there is no penalty for the *SiteConstraint* except at distances greater than this distance.

The *ParatopeSiteConstraintMover* adds *SiteConstraints* between each CDR residue and the antigen. This helps to keep the CDR paratope at the interface during docking; without it, docking can use the whole of the antibody surface instead of just the paratope and this can be seen in resulting models. These paratope constraints are added automatically in the program, and the CDRs of the paratope can be controlled through an option.

The *ParatopeEpitopeSiteConstraintMover* adds *SiteConstraints* from the epitope residues to the paratope residues and from the paratope to the epitope. Target epitope residues can be specified via command-line or automatically detected via the set interface distance. These constraints are off by default, but if they are enabled, they are set instead of the *ParatopeSiteConstraintMover* and help to keep the paratope and the epitope in contact

during design when the docking component of the algorithm is enabled.

CDR Design Instructions File

The CDR Instruction File handles CDR-level control of the algorithm and design. It is used to create the *CDRSet* for sampling whole CDRs from the antibody design database as well as fine-tuning the minimization steps and sequence design strategies. The CDR Instruction File allows a user to tailor how the program works for a variety of design projects and design strategies and generally controls which CDRs will be designed both for Graft-based and Sequence-based design (Fig. 3). Full details of the file and commands available are available in the *RosettaAntibodyDesign manual* (https://www.rosettacommons.org/docs/latest/application_documentation/antibody/RosettaAntibodyDesign).

Commands are broken into four types, each controlling different aspects of the protocol: *CDRSet*, *GraftDesign*, *SeqDesign*, and *MinProtocol.* The first column on each line specifies which CDR the line's commands will be attributed to. For each option, 'ALL' can be given to control all of the CDRs at once. ALL can be used to set all CDRs to something and then one or more CDRs can be set to something else in succeeding lines. Commands are not lower- or upper- case-sensitive. A '#' character at the beginning of a line is a file comment and is skipped.

CDRSet Instructions

The *CDRSet* instructions tell the program which CDR lengths, clusters, and specific structures to include or exclude from the antibody design database for graft-based design. By default, every length is enabled up to length 30. Additionally, some command-line options are also used to control the final *CDRSet*. These options include the specification of the light-chain gene (kappa or lambda) to limit light-chain grafts to those that originate from that gene (which is recommended for stability purposes).

The majority of H3 structures contain a "kink" at the C-terminus involving a Cα-Cα-Cα-Cα dihedral around 0° for the last three residues of H3 and the conserved tryptophan residue immediately following H3. More than 80% of H3 structures contain this kink, whose function in part is to break the β sheet and allow the H3 CDR to form diverse non-β structures[89]. H3-specific control is available, such as limiting the H3 *CDRSet* to kinked-only structures. The kink option is useful if H3 is being sequence designed, as some mutations in kinked H3s may make the H3 adopt an extended β-strand-like conformation and vice-versa. In addition, by default, we disable sequence design of the H3 stem region, which is known to influence the H3-kink[89,90].

Sequence Design Instructions

The *SeqDesign* instruction type is used to control the sequence design aspect of the program. In addition to controlling which CDRs undergo sequence design, the *SeqDesign* options control which strategy to use when doing sequence design (primary strategy), and which strategy to use if the primary strategy cannot be used for that CDR (fallback strategy), such as conservative design or no design.

Minimization

The type of minimization and which CDRs are minimized as neighbors to other CDRs during the protocol can be specified through the *MinProtocol* section. Many minimization types are implemented including the backrub motion[62,103]; however, they each require different computational resources and may require an increased number of inner cycles in order to structurally optimize the CDRs sufficiently.

3. Computational Benchmarking

In order to reduce artifacts, all benchmarking complexes and starting complexes for antibody design were first minimized into the Rosetta energy function using the Pareto-optimal protocol of Nivon et al.[61], which essentially relaxes[59] the starting structure using restraints on the backbone and side-chain atoms to strike a balance between deviation from the starting structure and minimization of the energy. This Pareto-optimal method produces models with all-atom Root Mean Square Deviations (RMSD) to the starting structures at a mean of 0.176 Å[61]**.** For all starting structures, the lowest-energy model of ten decoys was used as the starting structure. An example command to run this protocol and the flags is given in the Supplemental Methods.

For the antibody design benchmarks, the antibody design database included with Rosetta was used, which reflects data from the original North/Dunbrack clustering[52], while each antibody design project tested in experiments used up-to-date antibody design databases output by PyIgClassify[53]. All benchmarking and antibody designs used the Rosetta Talaris2014 energy function.

All antibodies were renumbered into the AHo numbering scheme[77] using PyIgClassify. The *CDRClusterFeatures* and *AntibodyFeatures* reporters were used to determine CDR length and cluster information and physical characteristics of the decoys for benchmarking and design selection. In general, analysis was done using the Feature Reporters to analyze decoys and create databases with physical data, and the public, open-source Jade repository (https://github.com/SchiefLab/Jade) was used for benchmarking calculations and selections. The color_cdrs.pml script was used to for antibody visualization and structural analysis in PyMol:

https://github.com/SchiefLab/Jade/blob/master/apps/color_cdrs.pml

Benchmarking

Antibody-protein complexes used for benchmarking consist of seven kappa and three lambda structures , which were chosen to have resolution better than 2.8 Å and R-factor < 0.3, as well as by their diversity in CDR lengths and clusters.

For each input antibody, we ran a total of 100 simulations, and the best decoy observed during each design run was output. The benchmarking was run on a compute cluster in parallel via MPI. The RunRosettaMPIBenchmarks.py script of the Jade github repository was used to help launch and configure benchmarks on the cluster (https://github.com/SchiefLab/Jade).

The number of outer cycles for each parallel run was set to 100, so each input antibody underwent 10,000 total design cycles for each experimental group. The groups included the 'min', 'relax', and both with or without antigen (which is done through the -remove_antigen flag). The 'min' group used Rosetta's standard dihedral-angle-space structure minimization as the *MinProtocol*, while the 'relax' group used *RelaxedDesign*. The

inner cycles were set to 1 for the *RelaxedDesign* group, while the min group used a setting of two, which essentially means that it undergoes a cycle of min->pack/design->min->pack/design.

All 5 non-H3 CDRs were allowed to undergo *GraftDesign*, while all 6 CDRs went through *SeqDesign*. The starting CDR for each non-H3 CDR was removed at the start of the program and a random CDR from *CDRSet* was grafted on through the option *–random_start*. The gene type for each set of designs was specified in order to limit the *GraftDesign CDRSet* to either lambda or kappa antibody data through the option *–light_chain*.

Average risk ratios and their confidence intervals

The proper way to average ratios, especially when some of the values are above 1.0 and some are below 1.0, is to take the geometric mean. The reason is that the ratios less than 1.0 are bounded by 0.0 while the ratios greater than 1.0 are unbounded, creating a non-normal distribution. Taking the logs of the ratios creates a normal distribution (if the data are well behaved)[104]. Risk ratios for antibody CDRs are averaged by taking the geometric mean of the risk ratios for each CDR type over the 10 antibodies in the benchmark set. The risk ratio for each CDR for each antibody is *RR=R*ec/*Obs* where *Rec* is the rate of recovery for the native CDR in the top designs output by the 100 simulations and *Obs* is the rate of sampling of the native CDR during the 100 simulations. Instead, we take:

$$RR_{ave} = 100 \times \exp\left(\frac{1}{10} \sum_{i=1,10} \log RR_i \right)$$

where RR_i is the risk ratio of the native CDR in antibody *i* in the benchmark.

The confidence intervals are calculated as described by Gertsman[105]. If $RR = p_{Rec}/p_{Obs}$, then:

$$se_\log = \sqrt{\frac{(1-p_{Rec})}{N_{Rec}p_{Rec}} + \frac{(1-p_{Obs})}{N_{Obs}p_{Obs}}}$$

$$\text{Confidence interval (log)} = \ln(RR_{ave}) \pm 1.96\ se_\log$$

$$\text{Confidence interval (non-log)} = \left\{ \exp\left(\ln(RR_{ave}) - 1.96\ se_\log\right),\ \exp\left(\ln(RR_{ave}) - 1.96\ se_\log\right) \right\}$$

4. Designs for experimental testing of computational antibody design

2J88 Design

The starting antibody complex, was obtained from the Protein Data Bank with ID 2J88[106], renumbered using PyIgClassify, and minimized into the Rosetta energy function as described above. To begin design, we used a multiple-strategy approach including with or without docking and the explicit use of only CDR clusters which have cluster profiles (more than 10 non-redundant members in the database), as well as differential CDR design for both graft-based and sequence-based design (H2 vs. L1). When designing the L1 loop, we also included a strategy in which we allowed L4 to undergo sequence design, for a total of 6 antibody design strategies. The WT L1 or H2 CDR was removed and a random CDR from the database was grafted in order to start design with the non-native CDR, as well as start with a potentially higher-energy structure. All CDR structures from the WT 2J88 antibody were left out of the *CDRSet*.

For the strategies in which docking was enabled, epitope constraints were used and

the inner cycles option was set to two in order to optimize the docking structure, so that the sequence of steps is min->dock->min->dock, taking the lowest energy structure observed during both inner cycles as described above.

A total of 1000 top decoys were output as separate Monte Carlo trajectories in parallel for each design strategy using a compute cluster via MPI, with 100 outer cycles for each parallel run, for a total of 100,000 design rounds per strategy. The RunRosettaMPI Jade application was used to aid the cluster run (https://github.com/SchiefLab/Jade#runrosettampi). Each decoy output was also relaxed using the dualspace relax protocol[60] (using the option *–run_relax),* so that there were nonrelaxed and relaxed versions of the top 1000 designs for decoy discrimination purposes. The command to run the application, flags, and CDR Instructions are given in the Supplemental Methods.

Decoys were analyzed by the Rosetta Feature reporter framework in the exact same manner as the benchmarking (commands and flags above). The features databases were then used in the Jade Antibody Design GUI in order to sort them for selection (**Figure S21**) (https://github.com/SchiefLab/Jade#rabd_jade).

For both relaxed and unrelaxed sets of decoys and each antibody design strategy, we sorted the models according to their computed interface energy (dG) after culling to only the top 10% of the models by total energy (dG_top_p_total), or by the lowest density of unsaturated hydrogen bonds per interface area (delta_unsats_per_1000_dSASA)[107] for a total of 24 sorted groups (6 design strategies * 2 decoy discrimination methods * 2 sorting methods).

For the three design strategies where docking was enabled and sorted by unsaturated hydrogen bond density (3 design strategies * 2 decoy discrimination methods *1 sorting strategy), the best two models had antibodies that were too far from the native binding site, even with the use of epitope *SiteConstraints*. This could be due to not using the constraint energy as a filter in this case, only to guide the design –i.e., not in the sorting of the total energy. Due to this, these 6 groups were left out for a total of 18 groups. The best two models from the sorted, unrelaxed groups (9 groups, 18 designs) and the best model of the sorted, relaxed groups were expressed (9 groups, 9 designs) (i.e., chosen with no human intervention). Three other models of the sorted relaxed groups for L1 design were added to the expression group, as these were the second-best scoring models of the L1 relaxed groups, for a total of 30 antibodies selected for expression (**Figure S17**).

L1/dock off/dG_top_p_total sort – design *L1_4*
L1/dock off/delta_unsats sort – design *L1-8*
L1/dock on/dG_top_p_total sort – design *L14-5*

Sequences were obtained from the decoys and processed for inclusion into the expression vector sequences using the *get_seq* application of Jade (https://github.com/SchiefLab/Jade#get_seq)

CH103 Design

The starting antibody complex was obtained from the Protein Data Bank with ID 4JAN[81], renumbered using PyIgClassify, and minimized into the Rosetta energy function as described above. A total of four antibody design strategies were used where either H2 or

L1+L3 were designed and the *CDRSet* included only clusters with enough data to use profiles. 250 top decoys were output for parallel Monte Carlo trajectories in parallel for each antibody design strategy with the outer cycle rounds set to 200, for a total of 50, 000 design cycles per design strategy. Commands, flags, and CDR Instructions are given in the Supplemental Methods.

Decoys were analyzed with both the RosettaFeatures reporters and physical data and sorted as described for 2J88. In addition to sorting by the top dG of the top 10% of total energy (*dG_top_Ptotal*) and density of unsaturated hydrogen bonds per interface area (*delta_unsats_per_1000_dSASA*), we sorted by the Lawrence and Colman Shape Complementarity score[80] (*sc_value*) through the Jade RAbD GUI. Sorts were done for both relaxed and nonrelaxed decoys to aid in decoy discrimination. The sorts were done for individual antibody design strategies and all combined for a total of 28 sorted groups. Jade was used to output PyMol sessions of each group. The top 10 designs from each group were visually analyzed in PyMol and 27 designs were selected based on physical characteristics such as good shape complementarity, hydrogen bonding, interface, and total energies, as well as cluster and sequence redundancy in the designs. Generally, the top design selected from each sort was expressed, unless it was redundant or the structure held some abnormality, such as extremely complex hydrogen bond networks or bad shape complementarity. Sequences were obtained from the decoys and processed for inclusion into the expression vector sequences using the *get_seq* application of Jade.

Availability

RosettaAntibodyDesign is distributed with the Rosetta Software Suite (www.rosettacommons.org) and is included with Rosetta versions starting at 3.8. All RosettaAntibodyDesign framework classes are available for scripting within the *RosettaScripts* framework[92], including the main application. Up-to-date antibody design databases for use with RosettaAntibodyDesign can be obtained from PyIgClassify (http://dunbrack2.fccc.edu/PyIgClassify/default.aspx). Documentation on the use of *RosettaAntibodyDesign* can be found with the RosettaCommons documentation, (https://www.rosettacommons.org/docs/latest/application_documentation/antibody/RosettaAntibodyDesign).

Acknowledgments. This work was performed with support from NIH Grants R01 GM111819 (R.L.D.), R35 GM122517 (R.L.D.), P01 AI094419 (W.R.S.). JAB received support from NIAID training grant T32 AI007244.

REFERENCES

1. Diamandis EP, Christopoulos TK (1991). The biotin-(strept) avidin system: principles and applications in biotechnology. *Clinical chemistry* **37**: 625-636.
2. Holliger P, Hudson PJ (2005). Engineered antibody fragments and the rise of single domains. *Nature biotechnology* **23**: 1126-1136.
3. Hoogenboom HR (1997). Designing and optimizing library selection strategies for generating high-affinity antibodies. *Trends in biotechnology* **15**: 62-70.
4. Adams GP, Weiner LM (2005). Monoclonal antibody therapy of cancer. *Nature biotechnology* **23**: 1147-1157.
5. Wu AM, Senter PD (2005). Arming antibodies: prospects and challenges for immunoconjugates. *Nature biotechnology* **23**: 1137-1146.
6. Adams GP, Schier R, McCall AM, Simmons HH, Horak EM, Alpaugh RK, Marks JD, Weiner LM (2001). High affinity restricts the localization and tumor penetration of single-chain fv antibody molecules. *Cancer research* **61**: 4750-4755.
7. Yang L, Mao H, Wang YA, Cao Z, Peng X, Wang X, Duan H, Ni C, Yuan Q, Adams G (2009). Single chain epidermal growth factor receptor antibody conjugated nanoparticles for in vivo tumor targeting and imaging. *Small* **5**: 235-243.
8. Smith GP (1985). Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface. *Science* **228**: 1315-1317.
9. McCafferty J, Griffiths AD, Winter G, Chiswell DJ (1990). Phage antibodies: filamentous phage displaying antibody variable domains. *Nature* **348**: 552-554.
10. Clackson T, Hoogenboom HR, Griffiths AD, Winter G (1991). Making antibody fragments using phage display libraries. *Nature* **352**: 624-628.
11. Winter G, Griffiths AD, Hawkins RE, Hoogenboom HR (1994). Making antibodies by phage display technology. *Annual review of immunology* **12**: 433-455.
12. Hoogenboom HR, Chames P (2000). Natural and designer binding sites made by phage display technology. *Immunology today* **21**: 371-378.
13. Bratkovič T (2010). Progress in phage display: evolution of the technique and its applications. *Cellular and molecular life sciences* **67**: 749-767.
14. Hawkins RE, Russell SJ, Winter G (1992). Selection of phage antibodies by binding affinity: mimicking affinity maturation. *Journal of molecular biology* **226**: 889-896.
15. Daugherty PS, Chen G, Olsen MJ, Iverson BL, Georgiou G (1998). Antibody affinity maturation using bacterial surface display. *Protein engineering* **11**: 825-832.
16. Luginbühl B, Kanyo Z, Jones RM, Fletterick RJ, Prusiner SB, Cohen FE, Williamson RA, Burton DR, Plückthun A (2006). Directed evolution of an anti-prion protein scFv fragment to an affinity of 1 pM and its structural interpretation. *Journal of molecular biology* **363**: 75-97.
17. Lonberg N (2008). Human monoclonal antibodies from transgenic mice. Therapeutic Antibodies: Springer. pp. 69-97.
18. Milstein C (1999). The hybridoma revolution: an offshoot of basic research. *Bioessays* **21**: 966-973.
19. Lonberg N, Huszar D (1995). Human antibodies from transgenic mice. *International reviews of immunology* **13**: 65-93.

20. Sergeeva A, He H, Ruisaard K, St John L, Alatrash G, Clise-Dwyer K, Li D, Patenia R, Hong R, Sukhumalchandra P, You M, Gagea M, Ma Q, Molldrem J (2016). Activity of 8F4, a T-cell receptor-like anti-PR1/HLA-A2 antibody, against primary human AML in vivo. *Leukemia* **30**: 1475-1484.
21. Kawashima Y, Kurima K, Pan B, Griffith AJ, Holt JR (2015). Transmembrane channel-like (TMC) genes are required for auditory and vestibular mechanosensation. *European Journal of Physiology* **467**: 85-94.
22. Tiller KE, Tessier PM (2015). Advances in antibody design. *Annual review of biomedical engineering* **17**: 191-216.
23. Corti D, Lanzavecchia A (2013). Broadly neutralizing antiviral antibodies. *Annual review of immunology* **31**: 705-742.
24. Yuan Q-a, Simmons HH, Robinson MK, Russeva M, Marasco WA, Adams GP (2006). Development of engineered antibodies specific for the Müllerian inhibiting substance type II receptor: a promising candidate for targeted therapy of ovarian cancer. *Molecular cancer therapeutics* **5**: 2096-2105.
25. Ohaegbulam KC, Assal A, Lazar-Molnar E, Yao Y, Zang X (2015). Human cancer immunotherapy with antibodies to the PD-1 and PD-L1 pathway. *Trends in molecular medicine* **21**: 24-33.
26. Kuroda D, Shirai H, Jacobson MP, Nakamura H (2012). Computer-aided antibody design. *Protein Eng Des Sel* **25**: 507-521.
27. Marshall SA, Lazar GA, Chirino AJ, Desjarlais JR (2003). Rational design and engineering of therapeutic proteins. *Drug discovery today* **8**: 212-221.
28. Caravella JA, Wang D, Glaser SM, Lugovskoy A (2010). Structure-guided design of antibodies. *Current computer-aided drug design* **6**: 128-138.
29. Lippow SM, Wittrup KD, Tidor B (2007). Computational design of antibody-affinity improvement beyond in vivo maturation. *Nat Biotechnol* **25**: 1171-1176.
30. Barderas R, Desmet J, Timmerman P, Meloen R, Casal JI (2008). Affinity maturation of antibodies assisted by in silico modeling. *Proceedings of the National Academy of Sciences* **105**: 9029-9034.
31. Karanicolas J, Kuhlman B (2009). Computational design of affinity and specificity at protein–protein interfaces. *Current opinion in structural biology* **19**: 458-463.
32. Chennamsetty N, Voynov V, Kayser V, Helk B, Trout BL (2009). Design of therapeutic proteins with enhanced stability. *Proceedings of the National Academy of Sciences* **106**: 11937-11942.
33. Miklos AE, Kluwe C, Der BS, Pai S, Sircar A, Hughes RA, Berrondo M, Xu J, Codrea V, Buckley PE (2012). Structure-based design of supercharged, highly thermoresistant antibodies. *Chemistry & biology* **19**: 449-455.
34. Der BS, Kluwe C, Miklos AE, Jacak R, Lyskov S, Gray JJ, Georgiou G, Ellington AD, Kuhlman B (2013). Alternative computational protocols for supercharging protein surfaces for reversible unfolding and retention of stability. *PLOS ONE* **8**: e64363.
35. Pulito VL, Roberts VA, Adair JR, Rothermel AL, Collins AM, Varga SS, Martocello C, Bodmer M, Jolliffe LK, Zivin RA (1996). Humanization and molecular modeling of the anti-CD4 monoclonal antibody, OKT4A. *The Journal of Immunology* **156**: 2840-2850.
36. Lazar GA, Desjarlais JR, Jacinto J, Karki S, Hammond PW (2007). A molecular immunology approach to antibody humanization and functional optimization. *Molecular immunology* **44**: 1986-1998.

37. Qiao C, Lv M, Li X, Geng J, Li Y, Zhang J, Lin Z, Feng J, Shen B (2013). Affinity maturation of antiHER2 monoclonal antibody MIL5 using an epitope-specific synthetic phage library by computational design. *Journal of Biomolecular Structure and Dynamics* **31**: 511-521.
38. Jacobs TM, Yumerefendi H, Kuhlman B, Leaver-Fay A (2014). SwiftLib: rapid degenerate-codon-library optimization through dynamic programming. *Nucleic acids research*: gku1323.
39. Ponsel D, Neugebauer J, Ladetzki-Baehs K, Tissot K (2011). High affinity, developability and functional size: the holy grail of combinatorial antibody library generation. *Molecules* **16**: 3675-3700.
40. Chaparro‐Riggers JF, Polizzi KM, Bommarius AS (2007). Better library design: data‐driven protein engineering. *Biotechnology journal* **2**: 180-191.
41. Pantazes RJ, Maranas CD (2010). OptCDR: a general computational method for the design of antibody complementarity determining regions for targeted epitope binding. *Protein Eng Des Sel* **23**: 849-858.
42. Li T, Pantazes RJ, Maranas CD (2014). OptMAVEn–A New Framework for the de novo Design of Antibody Variable Region Models Targeting Specific Antigen Epitopes. *PloS one* **9**: e105954.
43. Dunbrack RL, Jr., Cohen FE (1997). Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci* **6**: 1661-1681.
44. Pantazes RJ, Maranas CD (2013). MAPs: a database of modular antibody parts for predicting tertiary structures and designing affinity matured antibodies. *BMC Bioinformatics* **14**: 168.
45. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D (2003). Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of molecular biology* **331**: 281-299.
46. Poosarla VG, Li T, Goh BC, Schulten K, Wood TK, Maranas CD (2017). Computational de novo design of antibodies binding to a peptide with high affinity. *Biotechnology and Bioengineering* **6**: 1331-1342.
47. Lapidoth GD, Baran D, Pszolla GM, Norn C, Alon A, Tyka MD, Fleishman SJ (2015). AbDesign: An algorithm for combinatorial backbone design guided by natural conformations and sequences. *Proteins* **83**: 1385-1406.
48. Chothia C, Lesk AM (1987). Canonical structures for the hypervariable regions of immunoglobulins. *J Mol Biol* **196**: 901-917.
49. Al-Lazikani B, Lesk AM, Chothia C (1997). Standard conformations for the canonical structures of immunoglobulins. *J Mol Biol* **273**: 927-948.
50. Martin AC, Thornton JM (1996). Structural families in loops of homologous proteins: automatic classification, modelling and application to antibodies. *J Mol Biol* **263**: 800-815.
51. Morea V, Tramontano A, Rustici M, Chothia C, Lesk AM (1998). Conformations of the third hypervariable region in the VH domain of immunoglobulins. *J Mol Biol* **275**: 269-294.
52. North B, Lehmann A, Dunbrack RL, Jr. (2011). A new clustering of antibody CDR loop conformations. *J Mol Biol* **406**: 228-256.

53. Adolf-Bryfogle J, Xu Q, North B, Lehmann A, Dunbrack RL, Jr. (2015). PyIgClassify: a database of antibody CDR structural classifications. *Nucleic Acids Res* **43**: D432-438.
54. Das R, Baker D (2008). Macromolecular modeling with rosetta. *Annual Review of Biochemistry* **77**: 363-382.
55. Mandell DJ, Coutsias EA, Kortemme T (2009). Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nat Methods* **6**: 551-552.
56. Das R (2013). Atomic-accuracy prediction of protein loop structures through an RNA-inspired ansatz. *PloS one* **8**: e74830.
57. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D (2003). Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J Mol Biol* **331**: 281-299.
58. Wang C, Bradley P, Baker D (2007). Protein-protein docking with backbone flexibility. *J Mol Biol* **373**: 503-519.
59. Tyka MD, Keedy DA, Andre I, Dimaio F, Song Y, Richardson DC, Richardson JS, Baker D (2011). Alternate states of proteins revealed by detailed energy landscape mapping. *J Mol Biol* **405**: 607-618.
60. Conway P, Tyka MD, DiMaio F, Konerding DE, Baker D (2014). Relaxation of backbone bond geometry improves protein energy landscape modeling. *Protein Sci* **23**: 47-55.
61. Nivon LG, Moretti R, Baker D (2013). A Pareto-optimal refinement method for protein design scaffolds. *PLoS One* **8**: e59004.
62. Smith CA, Kortemme T (2008). Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *J Mol Biol* **380**: 742-756.
63. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, Baker D (2003). Design of a novel globular protein fold with atomic-level accuracy. *Science* **302**: 1364-1368.
64. Röthlisberger D, Khersonsky O, Wollacott AM, Jiang L, DeChancie J, Betker J, Gallaher JL, Althoff EA, Zanghellini A, Dym O (2008). Kemp elimination catalysts by computational enzyme design. *Nature* **453**: 190-195.
65. Jiang L, Althoff EA, Clemente FR, Doyle L, Röthlisberger D, Zanghellini A, Gallaher JL, Betker JL, Tanaka F, Barbas CF (2008). De novo computational design of retro-aldol enzymes. *science* **319**: 1387-1391.
66. Richter F, Leaver-Fay A, Khare SD, Bjelic S, Baker D (2011). De novo enzyme design using Rosetta3. *PLoS One* **6**: e19230.
67. Guntas G, Purbeck C, Kuhlman B (2010). Engineering a protein-protein interface using a computationally designed library. *Proc Natl Acad Sci U S A* **107**: 19296-19301.
68. Fleishman SJ, Corn JE, Strauch EM, Whitehead TA, Karanicolas J, Baker D (2011). Hotspot-centric de novo design of protein binders. *J Mol Biol* **413**: 1047-1062.
69. Fleishman SJ, Whitehead TA, Ekiert DC, Dreyfus C, Corn JE, Strauch EM, Wilson IA, Baker D (2011). Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science* **332**: 816-821.
70. Leaver-Fay A, O'Meara MJ, Tyka M, Jacak R, Song Y, Kellogg EH, Thompson J, Davis IW, Pache RA, Lyskov S, Gray JJ, Kortemme T, Richardson JS, Havranek JJ, Snoeyink J, Baker D, Kuhlman B (2013). Scientific benchmarks for guiding macromolecular energy function improvement. *Methods Enzymol* **523**: 109-143.

71. Ihaka R, Gentleman R (1996). R: A language for data analysis and graphics. *J Comp Graphical Stat* **5**: 299-314.
72. R Core Team (2015) R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.
73. Sircar A, Kim ET, Gray JJ (2009). RosettaAntibody: antibody variable region homology modeling server. *Nucleic Acids Res* **37**: W474-479.
74. Weitzner BD, Kuroda D, Marze N, Xu J, Gray JJ (2014). Blind prediction performance of RosettaAntibody 3.0: grafting, relaxation, kinematic loop modeling, and full CDR optimization. *Proteins* **82**: 1611-1623.
75. Weitzner BD, Jeliazkov JR, Lyskov S, Marze N, Kuroda D, Frick R, Adolf-Bryfogle J, Biswas N, Dunbrack Jr RL, Gray JJ (2017). Modeling and docking of antibody structures with Rosetta. *Nature protocols* **12**: 401-416.
76. Nayeem A, Vila J, Scheraga HA (1991). A comparative study of the simulated‐annealing and Monte Carlo‐with‐minimization approaches to the minimum‐energy structures of polypeptides:[Met]‐enkephalin. *Journal of Computational Chemistry* **12**: 594-605.
77. Honegger A, Plückthun A (2001). Yet another numbering scheme for immunoglobulin variable domains: an automatic modeling and analysis tool. *J Mol Biol* **309**: 657-670.
78. Davies HTO, Crombie IK, Tavakoli M (1998). When can odds ratios mislead? *Bmj* **316**: 989-991.
79. Giudicelli V, Duroux P, Ginestoux C, Folch G, Jabado-Michaloud J, Chaume D, Lefranc MP (2006). IMGT/LIGM-DB, the IMGT comprehensive database of immunoglobulin and T cell receptor nucleotide sequences. *Nucleic Acids Res* **34**: D781-784.
80. Lawrence MC, Colman PM (1993). Shape complementarity at protein/protein interfaces. *Journal of molecular biology* **234**: 946-950.
81. Liao HX, Lynch R, Zhou T, Gao F, Alam SM, Boyd SD, Fire AZ, Roskin KM, Schramm CA, Zhang Z, Zhu J, Shapiro L, Program NCS, Mullikin JC, Gnanakaran S, Hraber P, Wiehe K, Kelsoe G, Yang G, Xia SM, Montefiori DC, Parks R, Lloyd KE, Scearce RM, Soderberg KA, Cohen M, Kamanga G, Louder MK, Tran LM, Chen Y, Cai F, Chen S, Moquin S, Du X, Joyce MG, Srivatsan S, Zhang B, Zheng A, Shaw GM, Hahn BH, Kepler TB, Korber BT, Kwong PD, Mascola JR, Haynes BF (2013). Co-evolution of a broadly neutralizing HIV-1 antibody and founder virus. *Nature* **496**: 469-476.
82. Padavattan S, Schirmer T, Schmidt M, Akdis C, Valenta R, Mittermann I, Soldatova L, Slater J, Mueller U, Markovic-Housley Z (2007). Identification of a B-cell epitope of hyaluronidase, a major bee venom allergen, from its crystal structure in complex with a specific Fab. *J Mol Biol* **368**: 742-752.
83. Lehmann A, Wixted JH, Shapovalov MV, Roder H, Dunbrack RL, Jr., Robinson MK (2015). Stability engineering of anti-EGFR scFv antibodies by rational design of a lambda-to-kappa swap of the VL framework using a structure-guided approach. *MAbs* **7**: 1058-1071.
84. Adolf-Bryfogle J, Dunbrack Jr RL (2013). The PyRosetta Toolkit: a graphical user interface for the Rosetta software suite. *PloS one* **8**: e66856.
85. Kleffner R, Flatten J, Leaver-Fay A, Baker D, Siegel JB, Khatib F, Cooper S (2017). Foldit Standalone: a video game-derived protein structure manipulation interface using Rosetta. *Bioinformatics*: btx283.

86. James LC, Roversi P, Tawfik DS (2003). Antibody multispecificity mediated by conformational diversity. *Science* **299**: 1362-1367.
87. Clark LA, Boriack‐Sjodin PA, Eldredge J, Fitch C, Friedman B, Hanf KJ, Jarpe M, Liparoto SF, Li Y, Lugovskoy A, Miller S, Rushe M, Sherman W, Simon K, Van Vlijmen H (2006). Affinity enhancement of an in vivo matured therapeutic antibody using structure‐based computational design. *Protein science* **15**: 949-960.
88. Wei Q, Dunbrack Jr RL (2013). The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLOS ONE* **8**: e67863.
89. Weitzner BD, Dunbrack RL, Jr., Gray JJ (2015). The origin of CDR H3 structural diversity. *Structure* **23**: 302-311.
90. Weitzner BD, Gray JJ (2017). Accurate Structure Prediction of CDR H3 Loops Enabled by a Novel Structure-Based C-Terminal Constraint. *J Immunol* **198**: 505-515.
91. Henikoff S, Henikoff JG (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA* **89**: 10915-10919.
92. Fleishman SJ, Leaver-Fay A, Corn JE, Strauch EM, Khare SD, Koga N, Ashworth J, Murphy P, Richter F, Lemmon G, Meiler J, Baker D (2011). RosettaScripts: a scripting language interface to the Rosetta macromolecular modeling suite. *PLoS One* **6**: e20161.
93. Shapovalov MV, Dunbrack RL, Jr. (2011). A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure* **19**: 844-858.
94. Jacobs TM, Williams B, Williams T, Xu X, Eletsky A, Federizon JF, Szyperski T, Kuhlman B (2016). Design of structurally distinct proteins using strategies inspired by evolution. *Science* **352**: 687-690.
95. Lewis SM, Kuhlman BA (2011). Anchored design of protein-protein interfaces. *PloS one* **6**: e20872.
96. Canutescu AA, Dunbrack RL, Jr. (2003). Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Sci* **12**: 963-972.
97. Berkholz DS, Shapovalov MV, Dunbrack RL, Jr., Karplus PA (2009). Conformation dependence of backbone geometry in proteins. *Structure* **17**: 1316-1325.
98. Chaudhury S, Lyskov S, Gray JJ (2011). PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics* **26**: 689-691.
99. Kilambi KP, Pacella MS, Xu J, Labonte JW, Porter JR, Muthu P, Drew K, Kuroda D, Schueler-Furman O, Bonneau R, Gray JJ (2013). Extending RosettaDock with water, sugar, and pH for prediction of complex structures and affinities for CAPRI rounds 20-27. *Proteins* **81**: 2201-2209.
100. Marze NA, Jeliazkov JR, Roy Burman SS, Boyken SE, DiMaio F, Gray JJ (2016). Modeling oblong proteins and water-mediated interfaces with RosettaDock in CAPRI rounds 28-35. *Proteins*.
101. Schueler‐Furman O, Wang C, Baker D (2005). Progress in protein–protein docking: Atomic resolution predictions in the CAPRI experiment using RosettaDock with an improved treatment of side‐chain flexibility. *Proteins: Structure, Function, and Bioinformatics* **60**: 187-194.

102. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R, Kaufman K, Renfrew PD, Smith CA, Sheffler W, Davis IW, Cooper S, Treuille A, Mandell DJ, Richter F, Ban YE, Fleishman SJ, Corn JE, Kim DE, Lyskov S, Berrondo M, Mentzer S, Popovic Z, Havranek JJ, Karanicolas J, Das R, Meiler J, Kortemme T, Gray JJ, Kuhlman B, Baker D, Bradley P (2011). Rosetta3 an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol* **487**: 545-574.
103. Davis IW, Arendall WB, 3rd, Richardson DC, Richardson JS (2006). The backrub motion: how protein backbone shrugs when a sidechain dances. *Structure* **14**: 265-274.
104. Mukherjee C, White H, Wuyts M (1998) Econometrics and data analysis for developing countries. London: Routledge.
105. Gertsman BB (2013) Epidemiology Kept Simple: An Introduction to Traditional and Modern Epidemiology (3rd Edition). Oxford: John Wiley and Sons.
106. Padavattan S, Schirmer T, Schmidt M, Akdis C, Valenta R, Mittermann I, Soldatova L, Slater J, Mueller U, Markovic-Housley Z (2007). Identification of a B-cell epitope of hyaluronidase, a major bee venom allergen, from its crystal structure in complex with a specific Fab. *Journal of molecular biology* **368**: 742-752.
107. Stranges PB, Kuhlman B (2013). A comparison of successful and failed protein interface designs highlights the challenges of designing buried hydrogen bonds. *Protein Science* **22**: 74-82.