Published in final edited form as:
Nat Protoc. 2017 February ; 12(2): 401–416. doi:10.1038/nprot.2016.180.

Modeling and docking antibody structures with Rosetta

Brian D. Weitzner[1,*], **Jeliazko R. Jeliazkov**[2,*], **Sergey Lyskov**[1,*], **Nicholas Marze**[1], **Daisuke Kuroda**[1,3], **Rahel Frick**[4,5], **Jared Adolf-Bryfogle**[6,7], **Naireeta Biswas**[1], **Roland L. Dunbrack Jr.**[7], and **Jeffrey J. Gray**[1,2,8,9]

[1]Department of Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, Maryland, USA

[2]Program in Molecular Biophysics, Johns Hopkins University, Baltimore, Maryland, USA

[3]Department of Analytical and Physical Chemistry, Showa University School of Pharmacy, Tokyo 142-8555, Japan

[4]Centre for Immune Regulation, Department of Biosciences, University of Oslo, Oslo, Norway

[5]Centre for Immune Regulation, Department of Immunology, Oslo University Hospital Rikshospitalet, Oslo, Norway

[6]Department of Immunology and Microbial Science, The Scripps Research Institute, La Jolla, California, USA

[7]Institute for Cancer Research, Fox Chase Cancer Center, Philadelphia, Pennsylvania, USA

[8]Institute for NanoBioTechnology, Johns Hopkins University, Baltimore, Maryland, USA

[9]Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins School of Medicine, Baltimore, Maryland, USA

Abstract

We describe Rosetta-based computational protocols for predicting the three-dimensional structure of an antibody from sequence (RosettaAntibody) and then docking the antibody to protein antigens (SnugDock). Antibody modeling leverages canonical loop conformations to graft large segments from experimentally-determined structures as well as (1) energetic calculations to minimize loops, (2) docking methodology to refine the V_L–V_H relative orientation, and (3) *de novo* prediction of the elusive complementarity determining region (CDR) H3 loop. To alleviate model uncertainty, antibody–antigen docking resamples CDR loop conformations and can use multiple models to represent an ensemble of conformations for the antibody, the antigen or both.

Correspondence should be addressed to J.J.G. (jgray@jhu.edu).
[*]Equal contribution authors

AUTHOR CONTRIBUTIONS
BDW, NM, SL, DK, JRJ, JAB, and JJG developed the current version of RosettaAntibody. SL developed ROSIE and implemented the RosettaAntibody and SnugDock server apps. BDW implemented SnugDock in Rosetta 3, JRJ benchmarked SnugDock's performance. RF and NB wrote the procedure, codified the manual intervention steps developed by BDW, NM, and DK, and recorded timing information. BDW, JRJ, NM, SL, DK, RF, JAB, NB and JJG wrote the manuscript.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests. All revenue generated by licensing Rosetta to for-profit entities is invested into the continued development of the software.

These protocols can be run fully-automated via the ROSIE web server (http://rosie.rosettacommons.org/) or manually on a computer with user control of individual steps. For best results, the protocol requires roughly 1,000 CPU-hours for antibody modeling and 250 CPU-hours for antibody–antigen docking. Tasks can be completed in under a day by using public supercomputers.

INTRODUCTION

The vertebrate adaptive immune system is capable of promoting cells to degranulate or phagocytose nearly any foreign pathogen by producing immunoglobulin G (IgG) proteins (antibodies) that recognize a specific region (epitope) of a pathogenic molecule (antigen). The ability to bind diverse antigens requires a diverse population of antibodies, which is achieved through complex processes in bone marrow and lymphatic tissues, namely V(D)J recombination and somatic hypermutation. The diversity of antibodies is astonishing; the size of the theoretical naïve antibody repertoire is estimated to be > 10^{13} in humans[1]. In addition to their biological importance, antibodies are routinely used in biotechnology as probes and diagnostics, and there are dozens of antibodies approved as therapeutics[2].

Next-generation sequencing techniques have enabled rapid determination of large numbers of antibody sequences[1]. A limitation of these approaches is that no information about the specific atomic contacts between the antibody and antigen can be gleaned from these data sets. Atomic detail is required to consider specific antibody–antigen interactions, for example, in order to develop therapeutic antibodies or vaccines that are mimetics of extremely infectious antigens[3]. Although there are experimental methods capable of generating structural models in atomic detail (X-ray crystallography, nuclear magnetic resonance [NMR], neutron diffraction, cryo-electron microscopy [cryo-EM]), not all protein structures can be determined with these methods, and limited resources make it impossible to determine the structures of all of the sequences identified in high-throughput sequencing experiments. To bridge the sequence–structure gap, one must employ computational structure prediction methods. Perhaps more importantly, structure prediction methods are useful in diagnostics and drug discovery to define epitopes and help infer biological or therapeutic mechanisms.

The function of an antibody arises from its three-dimensional structure. The IgG isoform, the most common type of naturally occurring antibodies, consists of two identical sets of heavy and light chains arranged into a “Y” shape, with the four polypeptide chains joined by disulfide linkages. The heavy chain contains four domains, three adjacent constant domains (C_H1, C_H2, C_H3) and one variable domain (V_H), and the light chain consists of a single constant domain (C_L) and a variable domain (V_L). The C_H1 and V_H domains interact with the C_L and V_L domains to form the antigen-binding fragment (F_{ab}) or the “arms” of the Y. Within the F_{ab}, both variable domains are directed away from the remaining heavy chain constant domains and make up the variable fragment (F_V). At the tip of the F_V are three complementarity determining region (CDR) loops on each chain (CDR L1–3 and CDR H1–3) that form the region of the antibody, called the paratope, that recognizes its target. This F_v structure is common to other antibody isoforms (IgA, IgE, etc.).

Antibody homology modeling

The F_V is the focal point of the recombination and hypermutation events; as such, the primary difference among antibodies is the conformation, structural context, and chemical identity of their CDR loops. For this reason, antibody structure prediction methods focus on modeling the F_V. The F_V can be split into two regions: framework regions, and CDR loops. The framework regions have a high degree of structural conservation, making it possible to generate accurate models of framework regions from template structures.

Similarly, analysis of antibody crystal structures has revealed that five of the six CDR loops (CDR L1–3, H1, H2) adopt a limited number of distinct structures, referred to as canonical loop conformations[4]. The canonical conformation of a particular CDR loop can typically be identified from its length and sequence. Like the framework regions, the CDRs L1–3, H1, and H2 are also modeled using template structures. Sequences that might not adopt canonical conformations, and may therefore yield inaccurate predictions, can be readily recognized by severe mismatches to the known patterns.

The remaining CDR loop, H3, does not adopt canonical conformations and must be modeled *de novo*. Additionally, the H3 loop lies at the interface of the two domains (V_H and V_L) and can interact with residues on either chain. To account for these interactions as well as the overall geometry of the paratope, the V_L–V_H orientation is optimized during H3 modeling. Accurately modeling CDR H3 and the V_L–V_H orientation are typically the most challenging and critically important aspects of antibody structure prediction[5–7].

Protein–protein docking

While accurate predictions of unbound antibody structures are informative, they are void of an important biological context: the antibody–antigen (Ab–Ag) interaction. High-resolution structures of Ab–Ag complexes give insight to the molecular mechanism by which antibodies function, a necessity for rational design of vaccines or antibody therapeutics. Structures of Ab–Ag complexes can be determined through experimental methods, however, just as with unbound antibodies, these methods are limited by their throughput and expense and are not viable for all proteins. When experimental methods cannot be used to determine complex structures, computational protein–protein interface prediction (docking) provides an alternative approach.

In general, computational docking approaches strive to sample all possible interactions between two proteins to discern the biologically-relevant interaction. Predicting a protein–protein interaction *de novo* is challenging due to the sheer number of possible docked conformations. However, the sample space can be made tractable with information about the interaction. In the case of Ab–Ag interactions, the search space is limited because the antibody paratope, comprised of the six CDR loops, is the binding site for the cognate antigen epitope.

The Rosetta SnugDock algorithm leverages the information about the flexible and/or uncertain regions of the antibody to perform robust Ab–Ag docking[8]. SnugDock simulates the induced-fit mechanism through simultaneous optimization of several degrees of freedom. It performs rigid-body docking of the multi-body (V_L–V_H)–Ag complex, as well as remodeling of the CDR H2 and H3 loops, the latter of which typically contributes a plurality of atomic contacts to the Ab–Ag interaction[9,10]. SnugDock can also simulate conformer selection by swapping either the antibody or the antigen with another member of a pre-generated structural ensemble. Because SnugDock samples most of the conformation space available to antibody paratopes, it can refine antibody homology models with inaccuracies in the difficult-to-predict V_L–V_H orientation and CDR H3 loop.

When docking homology models, it is best if there is experimental evidence to suggest the general location of the epitope (within ~8 Å, approximately the correct side of the antigen domain), and in this protocol paper, we describe the local docking procedure in detail. If no information is available about the epitope, there are several programs that perform global docking or epitope prediction[11]. In particular, there are two fast-Fourier transform (FFT) rigid-body docking approaches that implement antibody-specific energy potentials: PIPER[12] with the antibody-ADARS potential[13], and ZDOCK[14] with the Antibody i-Patch potential[15]. FFT rigid-body approaches are fast, but they cannot account for antibody motions upon antigen binding or compensate for errors in the initial homology model; SnugDock is the only flexible-backbone antibody docking method. It can provide a global-antigen docking alternative but it is slower and, like others, can produce false-positive epitope predictions[8]. For local docking, SnugDock has been demonstrated to produce high-quality models when using an antibody homology model or crystal structure and the unbound antigen crystal structure as input[8]. In addition, SnugDock approaches used in the Critical Assessment of PRediction of Interactions (CAPRI) blind docking challenge[16] produced the best structure among all predictors for a flexible-loop target. SnugDock has been further assessed on a set of 15 antibody–protein-antigen targets using CAPRI rankings (Table 1). CAPRI uses star-based rankings (*** = high quality, ** = medium, * = acceptable, 0 = incorrect) that consolidate three similarity metrics: ligand-root-mean-squared deviation (RMSD), interface-RMSD, and fraction of native contacts recovered (f_{nat})[17]. Examining the highest attained CAPRI ranking among the ten lowest-energy docked models (starting with a homology modeled antibody), SnugDock currently produces 2***, 11**, and 2* models over 15 targets. These performance data are improved since the original SnugDock publication[8] due to updates in the energy function[18] and a switch to the kinematic loop closure (KIC) loop modeling method[19–21]. While SnugDock has not been benchmarked and was not originally intended to be used on peptide or small molecule antigens, there are no technical limitations to doing so; alternately FlexPepDock[22,23] (peptides) or RosettaLigand[24] (small molecules) can be used to capture the degrees of freedom of those antigens, albeit without sampling the antibody degrees of freedom.

Experimental design: Antibody homology modeling with RosettaAntibody (steps 1–11)

The protocol described in this paper enables a user to generate a structural model of an antibody from its sequence and a structural model of an antibody–antigen complex from structures of the antibody and its antigen (Fig. 1).

Generating a structural model of an antibody from sequence in RosettaAntibody uses homology modeling techniques, that is, it uses segments from known structures with similar sequences. As described in detail below, the input sequence is split into several components.

For each component, RosettaAntibody searches a curated database of known structures for the closest match by sequence and then assembles those structural segments into a model. That model is then used as the input for the next stage in which the CDR H3 loop is modeled and the V_L–V_H orientation is optimized.

Numbering the residues in the sequence—The RosettaAntibody protocol identifies the CDRs of the input antibody sequence through regular expression matching to the Kabat CDR definition[25], and it numbers the antibody residues according to the Chothia scheme[4].

Template selection—For each structural component considered (FR_L, FR_H, CDRs L1–3, H1–3), templates are selected by maximum sequence similarity using a BLAST-based method with custom databases constructed from high-quality structures in the PDB. Since the CDR identity and length each constrain the possible canonical CDR conformations, we use separate databases for each loop–length combination. For example, ten-residue H1 loops, eleven-residue H1 loops, and eleven-residue L1 loops are separate BLAST-formatted databases. This ensures a compatible canonical conformation is chosen for each CDR, although recently others have had success using different length loop templates, particularly when somatic hypermutation introduces indels[26].

The results for each structural component are sorted by BLAST bit score, and the sequence with best score is selected as the template.

Initial V_L–V_H orientations—The initial V_L–V_H orientation is also selected as a template by BLAST in the same way as the other structural components; in this case, the entire F_V sequence is used for the BLAST comparison. Unlike the other segments, ten V_L–V_H templates are selected to mitigate the weak correlation between sequence and orientation[27]. Starting from the list of all possible templates ordered by bit score, the best match is selected as the first template. To diversify the initial V_L–V_H orientations, all templates with similar V_L–V_H orientations (0.5 OCD, see Marze & Gray[27]) to this template are pruned from the list. The best match remaining in the list is selected as the second template, and candidate templates similar to the second template are now removed from the list. This winnowing is repeated to create ten distinct templates. One grafted model will be created from each of these ten initial V_L–V_H orientations.

Grafting CDR templates—Once the initial V_L–V_H orientations are set, the CDR templates are grafted onto each framework region by superposing the two overlapping residues on either side of the loop with their corresponding residues on the framework regions. The graft points are then adjusted using cycles of minimization, random torsional sampling, and Cyclic Coordinate Descent (CCD)[28,29] of the two stem residues to prevent unphysical bond lengths and angles from being incorporated into the model. Finally, the structure is relaxed[30,31] via iterations of side-chain optimization and gradient-based minimization while constraining the backbone and side-chain heavy atoms to find a native-like conformation at a local energy minimum in Rosetta's score function.

All-atom refinement of CDR H3 and the V_L–V_H orientation—The grafted models are crude and must be refined, particularly in the CDR H3 loop and the V_L–V_H orientation.

The H3 loop is first completely remodeled in the context of the antibody framework using the next-generation KIC (NGK) loop modeling protocol[21]. For speed, the H3 loop side chains are each reduced to a single low-resolution pseudo-atom, and to ensure sampling of the C-terminal kink conformation[32], atomic constraints are applied to the governing score function[33]. For subsequent high-resolution refinement, the all-atom CDR H3 side chains are recovered, all CDR side chains are repacked, and the CDR side chains and backbones are minimized. The V_L and the V_H domains are re-docked with a rigid-backbone RosettaDock protocol[34,35] to remove any clashes created by the new H3 conformation, and the antibody side chains are again repacked. Using NGK, H3 is refined again in the context of the updated V_L–V_H orientation. The CDRs are packed and minimized again, and the model is saved as a candidate structure. The first grafted model is used as the starting point for 1,000 refined models and the other grafted models are each used as the starting point for 200 refined models, for a total of 2,800 refined models. The models are sorted by Rosetta score, a proxy for the free energy, and thus low-scoring models indicate more favorable (better) energies. A subset of the low-scoring models can be selected (Box 1) as a set of final models or as an ensemble for docking or other downstream applications.

Box 1

Assessing antibody modeling and antibody–antigen docking results

The user must critically analyze computational models. To select high-quality models from a set produced by Rosetta, models should be evaluated by their energy, geometry, agreement with observations, and diversity.

Model Scores (Energy)

Model structures output by Rosetta are ranked according to score, and typically we suggest using the ensemble of the ten lowest-scoring structures. Scores can be examined for individual models, or for the whole set of models by plotting score versus RMSD (see Fig. S3 in Supplementary Tutorial). In most simulations, approximately 90% of the models will span a total score range of 30–50 Rosetta Energy Units (REU) or an interface score range from 0 to −12 REU (Talaris2014 score function[18]). Typically, about 1–5% of models will have scores ranging from within the bulk to 5–10 REU below the bulk, and the low scores (of either total or interface score) indicate that these are the models that Rosetta expects to be closest to the native structure. If the low-scoring models cluster in a single set within about an angstrom RMSD, this indicates that Rosetta has converged upon a set of closely related models. Deeper scoring wells and more densely populated wells provide higher confidence in the models. In simulations with multiple low-scoring structural clusters, each is similarly likely to be native-like.

Geometry

Assess the physical feasibility of the low-scoring models by eye in a molecular visualization package such as PyMOL. In rare cases, such as when template structures are unavailable, Rosetta may create obvious flaws such as polypeptide chain breaks or backbone clashes, particularly within the CDRs and at their graft points, so one should make a cursory examination of the model integrity. The accuracy of the non-H3 CDR loops can be assessed by comparing the CDR cluster of the grafted loop with the cluster of the input sequence as identified by North et al.[53] (see step 5). Likewise, the components of the V_L–V_H orientation can be checked to ensure they lie within the observed natural distributions (step 9); an exception to this rule can be made if the V_L–V_H orientation grafting templates and Rosetta sampling all lie far toward the edge of nature's distribution. For complex models, ensure that the lowest-scoring Ab–Ag models make good contacts between the antigen and the antibody paratope. Higher confidence can be assigned to complex models with large (~1200 Å^2), complementary interfaces[35], as well as those in which the H3 CDR loop makes several specific contacts. Any models discarded from the low-scoring set should be replaced with other low-scoring models.

Agreement with observations

Ensure that the models are consistent with any experimental observations. For example, if experimental data show that a particular residue, when mutated, eliminates binding, then ensure the paratope contacts at this site unless there is evidence for allosteric effects. Again, replace any discarded models with other low-scoring models.

Diversity

It can be useful to seek a diverse set of candidate models, for examples to enhance conformational sampling during ensemble docking or when there is no single low-scoring cluster of models. Thus, the model set might be amended to include low-energy models from different structural clusters.

Experimental design: Antibody–antigen docking with SnugDock (steps 12–17)

Computational docking can be used to generate models of Ab–Ag complexes. In general, docking entails (1) roughly identifying (within 8 Å) the interacting interface through either experiment or global docking and (2) refining the initial model through local docking. Below we describe local docking with SnugDock in detail.

Generating the starting model—SnugDock requires, as an input, a putative Ab–Ag complex that contains a reasonable interface[36]. The complex can be composed of single structures or sets of structures (ensembles, see Box 2). The interface defines the local search, between the antibody CDRs and the antigen. Initial models are often based on experimental results that identify interacting residues at the Ab–Ag interface, such as mutagenesis or chemical crosslinking assays. In the absence of experimental results, a global docking approach such as ZDOCK/iPatch[15] or PIPER/ADARS[13] can generate putative complexes for refinement. Global docking can also be achieved with SnugDock, albeit at a higher computational expense.

Box 2

Increasing sampling during docking by incorporating backbone structural ensembles

In Rosetta, an ensemble is a set of discrete conformations of a protein structure. SnugDock uses ensembles to approximate backbone conformational flexibility by sampling conformations from the ensemble during docking. Through this approach, not only does the protocol explore more conformational space than standard docking, but it can also compensate for model error, for example by using an ensemble of models produced by a modeling approach in a previous step such as RosettaAntibody.

Rosetta ensembles can be converted directly from NMR ensembles, or they can be generated using any method that induces structural diversity, such as molecular dynamics or various Rosetta refinement protocols. The ensembles typically span small structural variations of 1–2 Å backbone RMSD[34]. Rosetta's relax (unconstrained)[30,31] and KIC[19–21] protocols are suggested to generate docking ensembles for antigens. In addition, RosettaAntibody creates ensembles of antibodies by default. More on how to generating and docking ensembles can be found in Chaudury and Gray[34] and in Rosetta's documentation (https://www.rosettacommons.org/docs/latest/Home).

Antigen or antibody structures that have not been generated by a Rosetta protocol need to be refined before being placed in contact. Refinement, commonly referred to as the Relax protocol[30,31], entails iterations of side-chain optimization and gradient-based minimization in Rosetta's score function. The Relax protocol samples local conformational space around the starting structure to identify an energetic minimum in the score function. Through this process, Rosetta-identified non-idealities (such as van der Waals bumps) are abated. Once the partners have been refined (usually with the coordinates constrained to the starting position as in Nivón *et al.*[37]), a putative complex can be assembled and prepacked. Prepacking optimizes side-chain conformations to prevent biasing toward the input complex model's side-chain conformations, ensuring uniform scoring of all potential bound complex states.

Performing docking—SnugDock iteratively performs multi-body docking of both the Ab–Ag and V_L–V_H orientations and remodeling of the H2 and H3 CDR loops. Prior to docking, the antigen in the prepacked Ab–Ag complex is subject to three rigid-body perturbations: (1) a randomized "spin" about the Ab–Ag primary axis, uniformly sampled from [0,360°], (2) a small-magnitude random translation, with the magnitude sampled from a Gaussian distribution centered on 3 Å, and (3) a small-magnitude randomized "tilt" in a random direction off of the Ab–Ag primary axis, sampled from a Gaussian distribution centered on 8°. Docking operates in two phases: low-resolution mode, where side chains are represented by a single pseudoatom located at the centroid of the side-chain heavy atoms, and high-resolution mode, where all protein atoms are explicit. Low-resolution mode consists of two types of interspersed Monte Carlo moves: rigid-body Ab–Ag translation and rotation, and backbone ensemble conformer swaps. Additionally, at the end of low-resolution mode, the H2 & H3 loops are refined. High-resolution mode consists of a 50-step Monte Carlo trajectory where each move is selected from a set of five possible moves: rigid body Ab–Ag docking (40%), rigid body V_L–V_H docking (40%), CDR minimization (10%), H2 loop refinement (5%), and H3 loop refinement (5%), where the percentages indicate the probabilities of selecting each move. Each trajectory results in one model. Typically, SnugDock is used to generate a total of 1,000 models, with the low-scoring models most likely to be near the native conformation.

Incorporating experimental data into the simulation—Two main types of experimental data that inform the Ab–Ag binding mode can be incorporated into SnugDock. First, knowledge about specific residues or pairs of residues that interact across the interface can be used to guide docking. This information could, for example, be derived from alanine scanning or other mutagenesis experiments, as has been successfully done before[38]. Second, knowledge about the epitope and the overall Ab-Ag orientation can be incorporated. Complex structures have been successfully predicting using binding data derived from different experiments, including nuclear magnetic resonance (NMR) hydrogen–deuterium exchange, NMR chemical shift perturbation, low-resolution cryo-EM, or chemical crosslinking of the binding partners with subsequent analysis by mass spectrometry[39–41]. Other methods for epitope mapping may also be suitable.

Depending on the type of experimental data available, there are different ways of incorporating it into the docking simulation. High-confidence residue–residue interactions can be preserved with the use of atom pair constraints. Less-specific and poorly-characterized interactions (hydrophobic pockets, ambiguous H-bonds) can be loosely constrained with ambiguous and site constraints. Predicted epitopes and binding patches can be sampled by properly placing the SnugDock input structure and adjusting the size of the initial starting move. For further information on incorporating experimental constraints, see the Rosetta documentation (https://www.rosettacommons.org/docs/wiki/rosetta_basics/Incorporating-Experimental-Data).

Caveats, challenges and pitfalls

There are several caveats associated with computational modeling of antibodies and docking of antibodies and antigens. Keeping these caveats in mind, the user should critically assess each prediction (see Box 1). RosettaAntibody is a homology modeling approach and can be hampered by template availability. Challenging targets include heavily engineered antibodies, antibodies derived from a species that diversifies its antibodies through gene conversion, such as chickens or rabbits, or antibodies with flexible CDR H3 loops.

When templates exist, errors in the FR and CDR L1–3, H1, H2 loops are typically small (no greater than 1 Å backbone RMSD to native)[5]. In general, the V_L–V_H orientation is correctly captured by RosettaAntibody in 43 of 46 benchmark antibody targets[27]. On the other hand, the CDR H3 loop is modeled *de novo*, and loop model quality decreases with loop length. In the KIC loop benchmark,[21,42] loops of 12–17 residues are modeled to near 1 Å backbone RMSD relative to the native structure—the average human CDR H3 falls within that range with an average length of 15 residues (under the international ImMunoGeneticsDatabase [IMGT] definition of CDR H3)[43]. However, the benchmark is measured by modeling loops on crystallographic frameworks, whereas in a blind context CDR H3 loops are modeled on homology frameworks, which introduces uncertainty in the loop environment. Nevertheless, in a recent assessment[33] Rosetta Antibody produced models with CDR H3 loops within 1.59 Å backbone RMSD to native and sub-angstrom accuracy in all other regions.

RosettaAntibody models unbound, solution-state antibodies, and its predictions should be treated as such. Additionally, each RosettaAntibody structure is implicitly treated as rigid, and the user should be less confident in a model of a CDR H3 known or expected to be flexible. Flexibility can be approximated by considering an ensemble of models in downstream protocols. The ensemble approach has the dual benefit of accounting for uncertainty in our homology models.

Conversely, SnugDock models the antigen-bound state of antibodies. 37% of CDR H3 loops exhibit a conformational change greater than 1 Å upon antigen binding (this value is rarely greater than 1 Å for the other CDR loops and the V_L–V_H orientation)[44]. To account for motions upon binding as well as error introduced during antibody modeling, SnugDock samples alternate conformers from ensembles of antibodies and antigens, and it explicitly remodels the CDR H2 and H3 loops, docks the V_L–V_H chains, and minimizes the interface. Thus, SnugDock emulates the lock and key, conformer selection, and induced fit binding models of the antibody. SnugDock does not, however, explicitly sample backbone degrees of freedom of the antigen or of the other canonical CDRs of the antibody. If the unbound and bound conformations differ substantially or if the homology models are poor, it could be difficult or impossible to model the docked complex accurately[45]. Despite this complication, SnugDock has successfully predicted Ab–Ag complexes from homology models[8].

Comparable and Alternative Methods

Antibody Modeling—In addition to RosettaAntibody, there are three publicly accessible, fully automated web servers for antibody structure prediction: Kotai Antibody Builder[46,47], Prediction of ImmunoGlobulin Structures (PIGS)[48], and ABodyBuilder[49]. The performance of each method was discussed in the recent Antibody Modeling Assessment (AMA)[6], except for ABodyBuilder, which was developed and benchmarked on the AMA antibodies *ex post facto*. While similar, these approaches differ in some underlying methods for CDR template selection and loop modeling, resources needed, and best applications. For example, the Kotai Antibody Builder relies more heavily on sequence-based rules for template selection and for CDR H3 base geometry. PIGS favors the selection of CDR and framework templates from a single source structure when possible. PIGS does not include backbone refinement, and as a result it returns structures very rapidly with minimal computational cost. ABodyBuilder includes extensive refinement and has recent developments[49]; it is different in that it allows CDR templates of mismatched lengths[26] and exploits a full six-dimensional V_L-V_H determination strategy[7]. RosettaAntibody is unique in that its extensive conformational refinement focused in antibody degrees of freedom is designed and tested toward creating a structure that is at an energy minimum and appropriate for downstream applications including docking or design.

Antibody–Antigen docking—In addition to SnugDock, there are two freely available antibody docking approaches: PIPER[12] with the antibody-ADARS potential[13] and ZDOCK[14] with the Antibody i-Patch potential[15]. Both are rigid-body, FFT approaches, which do not capture side-chain or backbone flexibility as SnugDock does and with key differences in the formulation of the energy potentials and in the docking algorithms. Thus, these methods should be used to explore global conformation space rapidly, whereas SnugDock should be used for thorough local refinement. The performance of ZDOCK with the Antibody i-Patch potential has been benchmarked on the same set of complexes as SnugDock. Thus, a direct comparison is possible with the current implementation of

SnugDock, where i-Patch produces complexes of CAPRI criteria 0.7***/3.1**/7.0* (averaged over 20 simulations, best in top 10) and SnugDock produces complexes with CAPRI criteria of 2***/11**/2* (one simulation, best in top 10).

MATERIALS

EQUIPMENT

Homology modeling data

- Primary amino-acid sequence of the variable domain of the light and heavy chains.

Docking data

- File of the antigen structure, formatted in Protein Data Bank (PDB) standard.
- PDB-formatted file of the antibody structure or the homology modeling output, which consists of a single antibody with chains L and H.

  CRITICAL Both of these can be single structures or an ensemble of structures.

Software for running simulations via ROSIE web server

- Modern web browser
- RosettaAntibody and SnugDock can be run via a public webserver (http://rosie.rosettacommons.org), python bindings (PyRosetta, http://www.pyrosetta.org) and through local installations of Rosetta. Rosetta is distributed as source code and licenses are available from the RosettaCommons (http://www.rosettacommons.org) free of charge for academic and non-profit users. Rosetta can be installed on UNIX-like operating systems (including Mac OS X).

Hardware for running simulations manually (optional)

- Workstation with multi-core CPU(s) running a POSIX compliant operating system (*e.g.,* GNU/Linux, OS X) or a Linux-based cluster. Several public facilities are available. For example, the U.S. National Science Foundation’s provides clusters like Stampede through the Extreme Science and Engineering Discovery Environment (XSEDE, www.xsede.org). In Europe, the Partnership for Advanced Computing in Europe (PRACE, www.prace-ri.eu) provides access to clusters like JUQUEEN. Resources like the Norwegian Metacenter for Computational Science (Notur, www.notur.no) or Japan’s supercomputer facilities of National Institute of Genetics (sc.ddbj.nig.ac.jp) and of Human Genome Center at the University of Tokyo (hgc.jp) are also suitable.

Software for running simulations locally (optional)

- The Rosetta software suite, available at www.rosettacommons.org/software. Compilation instructions are available at www.rosettacommons.org/build.

Support for any issues encountered that are not covered in this manuscript can be addressed on the Rosetta user forums: www.rosettacommons.org/forum

?TROUBLESHOOTING

- BLAST+ (version 2.2.28 or later), available at ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/
- Text editor (*e.g.*, vim, emacs, nano)
- Optional: Python (www.python.org) or R (www.r-project.org) for analyzing results
- Optional: A molecular visualization package for viewing results and customizing starting structures for docking. Recommended packages include PyMOL (www.pymol.org)[50], UCSF Chimera (www.cgl.ucsf.edu/chimera)[51], and Kinemage (kinemage.biochem.duke.edu)[52]

PROCEDURE

CRITICAL The simplest way to create antibody and antibody–antigen complex structures is through the use of the ROSIE web server (rosie.rosettacommons.org)[53]. On ROSIE, the Antibody application uses the input antibody sequence to generate a homology model, and the SnugDock application uses the antibody model(s) and an antigen structures for docking. Both operations are entirely automated with a minimum of user input.

For greater control of the operation, we describe below the steps to run the protocols manually, including the key points for checking intermediate data and intervening with alternate choices. Users with structures of the unbound antibody and antigen can skip to docking stage (step 12). A detailed example, which can be run on a standard workstation, is supplied in the supplemental information (Supplementary Tutorial).

Antibody Homology Modeling TIMING Variable

1. ***Construction of a grafted F_V model (5 hrs)***: Set up your terminal. After installing BLAST+ and Rosetta (see Materials), launch an interactive terminal (*e.g.*, *Terminal* on mac or *xterm* on Linux) and set path variables to the executable programs needed as follows (bash syntax):

```
export ROSETTA=~/Rosetta
export ROSETTA3_DB=$ROSETTA/main/database
export ROSETTA_BIN=$ROSETTA/main/source/bin
export PATH=$PATH:$ROSETTA_BIN
```

In the first line above, replace " ~" with the parent directory where you installed Rosetta on your machine. Similarly, be sure the PATH variable includes the blastp program (e.g. `export PATH=$PATH:/path/to/blastp` where `/path/to/blastp` is replaced with the directory containing the *blastp* executable. These path settings may be added to a configuration file such as `.bashrc` so they are automatically set each time a terminal is open (logged into).

2 Create a working directory and navigate to it:

```
mkdir /path/to/my_dir
cd /path/to/my_dir
```

3 Obtain the amino acid sequences for the variable domain of your antibody (light chain and heavy chain) and save them in FASTA format (in your working directory) with the heavy and light chains noted in the comment lines, as follows:

```
> heavy
VKLEESGGGLVQPGGSMKLSCATSGFRFADYWMDWVRQSPEKGLEWVAEIRNKANNHATYYAESVKG
RF
TISRDDSKRRVYLQMNTLRAEDTGIYYCTLIAYBYPWFAYWGQGTLVTVS
> light
DVVMTQTPLSLPVSLGNQASISCRSSQSLVHSNGNTYLHWYLQKPGQSPKLLIYKVSNRFSGVPDRF
SG
SGSGTDFTLKISRVEAEDLGVYFCSQSTHVPFTFGSGTKLEIKR
```

4 Use Rosetta's grafting application to find suitable templates and graft them together to obtain a crude model of the antibody. Execute the application with the line below.

```
antibody. macosclangrelease \
          -fasta antibody_chains.fasta | tee grafting.log
```

The application will output a directory called `grafting`. The PDB-formatted files named `model-0.relaxed.pdb`, `model-1.relaxed.pdb`, …, `model-9.relaxed.pdb` will be your input for the H3 modeling. The " `| tee grafting.log`" part of the command records all the program output in the file `grafting.log` for later review. The " `\`" permits the command to be spread across multiple lines rather than just one.

?TROUBLESHOOTING

5 ***(Optional) Check grafted template structures (10 mins – 2 hrs)***: Assign the CDR loops in your models to the CDR loop clusters described by North et al.[54], using the same methodology as in [55], and check whether the chosen templates are suitable.

Run the cluster identification application as follows:

```
identify_cdr_clusters.macosclangrelease \
         –s grafting/model-*.relaxed.pdb \
         –out:file:score_only north_clusters.log
```

North *et al.* clustered all CDR loop structures by their backbone dihedral angles and named them by CDR type, loop length and cluster size (e.g. “H1-13-10” is the 10th most common conformation for 13-residue H1 loops). Occasionally, Rosetta chooses templates that are rare or inconsistent with the sequence preferences observed by North *et al.* For example, if Rosetta recommends the H1-13-10 cluster, the user might also consider the H1-13-1 cluster. Tables 3–7 of North *et al.* present consensus sequences for each cluster that can inform this decision.

Loops and clusters with proline residues are also worth a manual examination. Several clusters of North *et al.* are contingent on the presence of prolines in particular locations (e.g. L3-9-*cis*7-1 has a *cis*-proline at position 7). Because RosettaAntibody relies on BLAST to choose loop templates, occasionally a loop from an uncommon non-*cis*-proline cluster (e.g. L3-9-2) is chosen. In such cases it is best to manually select a loop template from the well-populated *cis*-proline cluster.

6 If desired, rerun grafting to replace a template with one from a manually-specified source structure. Use the `antibody` command line as above with an extra flag to specify a template. Follow the below example: to force Rosetta to use the CDR H1 loop from the PDB 1RZI as the template in the model, add the flag `–antibody:h1_template 1rzi`. Select templates for other regions accordingly:

```
antibody.macosclangrelease \
           -fasta antibody_chains.fasta \
           -antibody:h1_template 1rzi | tee graft.log
```

Flag	region
-antibody:l1_template **-antibody:l2_template** **-antibody:l3_template**	light chain CDR loops
-antibody:h1_template **-antibody:h2_template** **–antibody:h3_template**	heavy chain CDR loops
-antibody:light_heavy_template **-antibody:n_multi_templates 1**	V_L–V_H orientation
-antibody:frl_template **-antibody:frh_template**	Framework region of the light or heavy chain

7 ***H3 modeling (1 hr – 4 days)***: Copy the set of standard H3 modeling flags to your working directory and create a directory for the H3 modeling output:

```
cp $ROSETTA/tools/antibody/abH3.flags.
mkdir H3_modeling
```

8 Run Rosetta's `antibody_H3` application on the 10 models generated during grafting. This step requires 1,000 CPU hours and is often performed in parallel on a computer cluster (see Box 3).

For a Mac workstation, use the following command line:

```
antibody_H3. macosclangrelease \
         @abH3.flags \
          -s grafting/model-0. relaxed.pdb \
          -nstruct 1000 \
          -antibody:auto_generate_kink_constraint \
          -antibody:all_atom_mode_kink_constraint \
          -multiple_processes_writing_to_one_directory \
          -out:file:scorefile H3_modeling_scores.fasc \
          -out:path:pdb H3_modeling > h3_modeling-0.log 2>&1 &
```

`-s` specifies the input file (one of the grafted models generated in step 4.

`-nstruct` specifies the number of structures generated, which should be 1000 for `model.0.pdb` and 200 each for all other grafted models.

A specific numbering scheme can also be specific, see Box 4.

The expected output is the specified number of PDB files as well as a score file named `H3_modeling_scores.fasc`. All these files will appear in an output directory named `H3_modeling/`.

To trivially run in parallel, simply repeatedly execute the above command (changing input models, number of structures, and the output log as you wish). Each time the command is executed, an *antibody_H3* process is run in the background.

CRITICAL STEP Generating the 2,800 antibody structures takes approximately 2,500 CPU hours. Running 24 processes in parallel, on a modern 24-CPU workstation, expect ~4 days of run time. Distributing the work over nodes on a supercomputer can reduce this time to hours (see Materials).

9 ***(Optional) Check V_L-V_H orientation (5 mins)***: Check whether the V_L–V_H orientations of the antibody models are close to the orientations observed in antibody crystal structures found in the PDB. To do this, run the python script `plot_LHOC.py` using the following command line:

```
python $ROSETTA/main/source/scripts/python/public/
           plot_VL_VH_orientational_coordinates/plot_LHOC.py
```

This script will create a subfolder (`lhoc_analyis`) with separate plots for each of the four antibody Light–Heavy Orientational Coordinate frame (LHOC) metrics. Fig. 2 shows a representative plot of the heavy opening angle for two antibodies, one with a native-like distribution and another with a non-native distribution. Each plot shows the native distribution of V_L–V_H orientations (grey), the orientations sampled by Rosetta (black line) as well as the top 10 models (labeled diamonds) and the 10 different template structures generated during step 4 (dots). Antibody models that are outside the native distributions are unlikely to be correct.

10 ***Choose final antibody models (10 mins)***: Choose 10 of the antibody models as an ensemble for docking. The following criteria may be useful to consider as docking with ensembles aims to increase conformational diversity and sampling: Select models with the lowest total score – these are purportedly native-like; Select models with natural V_L–V_H orientation, falling within the observed distribution (grey); Select models derived from different templates to maintain diversity.

If all ten low-scoring models are outside the native distribution, consider returning to step 6 and manually select new templates for the relative orientation of the V_L and V_H chains by using the - `antibody:light_heavy_template` flag (*e.g.*, `antibody.macosclangrelease -antibody:light_heavy_template 1ABC`).

11 ***(Optional) Renumber antibody models (5 mins)***: Standard residue numbering facilitates comparison of different antibodies, but several different numbering schemes are used. RosettaAntibody uses the Chothia residue numbering scheme[4] by default, and other numbering schemes, such as Enhanced Chothia[56], AHo[57], IMGT[58], and Kabat[59], are specifiable with command options. To change residue numbering, we provide a conversion application. For example, to convert `best_antibody.pdb` from Chothia to AHo numbering, run:

```
antibody_numbering_converter.macosclangrelease \
           -s best_antibody.pdb \
           -input_ab_numbering Chothia \
           -output_ab_numbering AHO
```

Compatible numbering schemes and their eponymous options are `Chothia`, `Enhanced_Chothia`, `AHO`, `IMGT`, and `Kabat`. These options are also valid for other antibody-related Rosetta applications (e.g. SnugDock, below, or `identify_cdr_clusters`[55]).

Box 3

Using Rosetta on different platforms and running in parallel

Rosetta on different platforms

Throughout this protocol executables are suffixed by the platform and mode for which they were compiled (i.e. `antibody.macosclangrelease` indicates that the antibody executable was compiled on a MacOS operating system using the Clang compiler and it was compiled in release mode). The suffix is highlighted in orange throughout (`.macosclangrelease`). On other platforms you will replace this string with your operating system and compiler (for example, GNU/Linux platforms with gcc as the compiler will default to `.linuxgccrelease`). Additionally, the suffix is prefixed by `.mpi` (`.mpi.linuxgccrelease`) when the executable is built for the message passing interface (MPI) by an MPI compiler. MPI-compatible executables can communicate with one another for parallel processing, and some Rosetta executables use MPI non-trivially. However, most standard Rosetta applications are trivially parallelizable (“embarrassingly parallel”) and thus capable of running on both MPI and non-MPI systems.

Running in parallel

An example of how to locally run a non-MPI executable in parallel is given in step 8. In general, add the `-multiple_processes_writing_to_one_directory` flag to your command line, and then execute multiple instances of the process. This procedure works on a single desktop computer with multiple CPUs or remotely on a supercomputer cluster. However, running a Rosetta executable on a cluster strongly depends on the hardware configuration and available software (e.g. workload management software).

For example, to run a non-MPI executable via HTCondor: (1) save the standard command line as an executable bash script, (2) write a submit description file specifying the executable bash script and the number of processes to execute, and (3) use the `condor_submit` command with the description file as an argument to submit your jobs to the cluster.

On the other hand, MPI executables can be run in parallel locally by prepending the command line with the `mpirun –n XX` command, where XX is the number of processes to run, if your machine is configured to use the Open MPI library. Again, the exact depend on the specific cluster configuration. For example, to run an MPI executable on Stampede via the slurm workload manager: (1) save the standard command line as an executable bash script, (2) write a slurm batch script specifying the executable bash script and the number of tasks, and (3) use the `sbatch` command with the bash script as an argument to submit your jobs to the cluster.

Box 4

Antibody Numbering

RosettaAntibody uses the Chothia numbering scheme by default, though many other numbering schemes are used in the literature[4,56–59].

The following table lists the I/O options that can be given to most of RosettaAntibody apps and the currently implemented numbering schemes.

Flag	Accepted Schemes
-input_ab_scheme	Chothia Enhanced_Chothia
-output_ab_scheme	AHO IMGT Kabat

If you would like a decoy (or all of them) converted into a particular scheme post-modeling, an app is provided with the syntax given below. Note that the Chothia Scheme is the default input numbering scheme for RosettaAntibody and is only given here in the option as an example.

```
antibody_numbering_converter.macosclangrelease \
          -s best_antibody.pdb \
          -input_ab_numbering Chothia \
          -output_ab_numbering AHO
```

Antibody-Antigen Docking TIMING Variable

12 ***Clean antigen (or antibody) PDB (1 hr)***: Prepare the antigen and antibody for docking. Format your antigen (and antibody if you are not using a homology model produced by Rosetta Antibody) PDB file so it can be read by Rosetta. Run the following script:

```
$ROSETTA/tools/protein_tools/scripts/clean_pdb.py antigen.pdb C
```

Where `antigen.pdb` is a PDB file of your antigen and `C` is the one-letter chain identifier(s) for the antigen chain(s) in the PDB file.

13 ***(Optional) Refine antibody in Rosetta's score function (10 min)***: If you are not using an antibody model produced by Rosetta, you must refine the antibody structure by running the relax application. The command line is:

```
relax.macosclangrelease \
         -s antibody. pdb \
         -relax:constrain_relax_to_start_coords \
         -relax:ramp_constraints false \
         -ex1 \
         -ex2 \
         -use_input_sc \
         -flip_HNQ \
         -no_optH false
```

You may also wish to generate an ensemble of antibody structures, see Box 2.

14 ***Prepacking (30 min)***: Generate a PDB file that contains both your antibody and your antigen in the following order: light chain of your antibody (L), heavy chain of your antibody (H), and antigen (A). There are several ways to create and modify a PDB file. This can be done using either PyMOL (Option A), or by using the command line and a text editor such as Vim (Option B).

a. PyMOL

i. Load the antibody in a PyMOL session.

ii. If it is a model from Rosetta Antibody, the chains will already be labeled as H and L. Otherwise, use the `alter` command to change the chain ID of a selection:

```
alter chain A, chain='H'
alter chain B, chain='L'
```

iii. Load the antigen into the same PyMOL session.

iv. Change the antigen chain ID in a similar fashion. **CRITICAL STEP** if antigen chains share an ID with the antibody, you will have to be more specific with your selections (*e.g.,* `alter chain H and antigen, chain='A'`).

v. Save both objects in the same PDB file:

```
save complex.pdb, chains L+H+A
```

b. Command line and text editor

i. Concatenate the antibody and antigen pdbs:

```
cat antibody.pdb antigen.pdb > complex.pdb
```

ii. Open the file using the text editor (e.g. Vim) and alter the chain IDs. First, navigate to the chain ID column. Then engage blockwise visual mode (Ctrl-V), select the entire chain column for a specific ID (e.g. A), and delete it using the delete operator (d). Next select the column again using blockwise visual mode and insert the new chain ID (shift+I, “H”, Esc). Repeat this process for each chain.

iii. Reorder the chains. Using visual mode (V), select entire chains, and cut (d) and paste (p).

15 Load the `complex.pdb` in PyMOL and reorient the antibody and antigen using the RotO, MovO, and MvOZ editing commands until they are in contact. Alternatively, one can also use the translate command (*i.e.* `translate [x,y,z], selection`). If you know an approximate binding location, adjust the orientation accordingly. Save both objects in the same PDB file:

```
save antibody_antigen_start.pdb, chains L+H+A
```

16 To ensure low-energy starting side-chain conformations, prepack the monomers:

```
docking_prepack_protocol. macosclangrelease \
           -in:file:s antibody_antigen_start. pdb \
           -ex1 \
           -ex2 \
           -partners LH_A \
           -ensemble1 antibody_ensemble.list \
           -ensemble2 antigen_ensemble.list \
           -docking:dock_rtmin
```

`antibody_ensemble.list` is a text file that contains filenames with absolute paths to the ten antibody models selected after antibody modeling. In the case that you have a single crystal structure, you can omit the `–ensemble1` flag.

If antigen flexibility is expected, a family of structures can be created with other Rosetta applications (see Box 2). The text file `antigen_ensemble.list` will contain the filenames of your antigen (using absolute paths). NMR starting structures must be split (i.e. each model should be in its own PDB file). To use a single antigen structure, omit the `–ensemble2` flag.

17 ***Docking (1–15 hrs)***: Dock the antibody to the antigen. As in step 8, this is an expensive computational step and you have the option of running a single process, multiple processes on one machine, or splitting the job across processors on a supercomputer (see Box 3). Using the executable for an MPI-based computing cluster with 300 processes as an example, the command line for docking is:

```
mpirun -n 300 snugdock.mpi. linuxgccrelease \
            -s antibody_antigen_start.prepack.pdb \
            -ensemble1 antibody_ensemble.list \
            -ensemble2 antigen_ensemble.list \
            -antibody:auto_generate_kink_constraint \
            -antibody:all_atom_mode_kink_constraint
            -nstruct 1000
```

? TROUBLESHOOTING

TIMING

Here we report the time to generate a single, docked model from antibody sequence and antigen crystal structure. Typically, however, thousands of models are generated, so we also indicate the timing for the full, recommended simulations. These time estimates were computed on a 2 × 2.4 GHz Quad-Core Intel Xeon processor; timing will vary for other computer configurations. Furthermore, wait times for resources with queues are not factored. Historically, wait times on range from 0 to 15 days. Detailed information can be found on http://rosie.graylab.jhu.edu/about.

Step	Human Time	CPU Time per Model	Total CPU Time
(1–4) Construction of grafted Fv models	5 min	20 min	200 min
(5) Check grafted models	10 min	<1 min	10 min
(7–8) H3 modeling	5 min	20 min	1000 hrs
(9) Check V_L–V_H orientation	5 min	<1 min	10 min
(10) Choose models	10 min	5 min	5 min
(11) Renumber antibody models	5 min	<1 min	5 min
(12) Prepare antibody and antigen for docking	5 min	15 min	15 min
(13) Refine antibody in Rosetta’s score function	5 min	20 min	20 min
(14–16) Prepacking	5 min	30 min	30 min
(17) Docking	5 min	15 min	250 hrs

TROUBLESHOOTING

For troubleshooting advice, see Table 2

ANTICIPATED RESULTS

The antibody structure prediction and docking methods described in this paper each produce a set of structural models that have been evaluated by a score function. In the case of antibody structure prediction, we have found through benchmarking and participation in the AMA that the accuracy of frameworks and non-H3 CDR loops can typically be expected to be within 1.0 Å RMSD of the coordinates in a crystal structure. When the model deviates more than 1.0 Å in RMSD from crystallographic coordinates it is usually because there is not a suitable known template in the PDB. These situations should become increasingly rare as more structures are deposited into the PDB, although heavily engineered antibodies should always be modeled with care.

The H3 loop accuracy is variable and depends both on length and V_L–V_H orientation. Loop length is an important factor in the accuracy of *de novo* loop modeling methods because the search space increases exponentially with each additional residue in the loop. We expect accurate models of CDR H3 loops of length 14 or less[33], but the lowest-scoring model may not be the most accurate. We therefore recommend using all ten models for downstream analysis. In AMA-II, we found that non-native V_L–V_H orientations can lead to explicit interactions between the light chain and the CDR H3 loop that are indistinguishable from native interactions[5]. Using multiple V_L–V_H orientation templates[27] allows broader exploration of conformational space, sampling more low-scoring wells. Models generated from at least three different templates should be used to maximize the chance of capturing the native V_L–V_H orientation.

Through benchmarking Ab–Ag docking, we have found that the accuracy of a complex model depends on the starting configuration of the partners and the accuracy of the models for each partner. SnugDock samples local conformation space, thus a good starting structure (within 8 Å) generally results in sampling a near-native conformation. Equally important is the quality of the initial unbound models; near-native models enable increased docking performance (see Table 1: B-B rigid body-docking vs. U-U rigid-body docking). We have found that docking a homology modeled antibody to the crystal structure of the unbound antigen typically results in at least one model of acceptable quality in the ten low-scoring models (Table 1).

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors wish to thank Arvind Sivasubramanian, Aroop Sircar, and Sidhartha Chaudhury for their development of the original RosettaAntibody, SnugDock, and EnsembleDock methods. Jianqing Xu refactored the antibody code. We also thank the members of the RosettaCommons for the continued development of the Rosetta Software Suite. ROSIE simulations are carried out, in part, within the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. BW, NM, JRJ, RLD and JJG are supported by National Institutes of Health Grant R01 GM078221. SL is supported by National Institutes of Health Grant R01 GM73151. DK is supported by the DARPA Antibody Technology Program (HR-0011-10-1-0052) and the Japan Society for the Promotion of Science (grant number 15H06606). RF is supported by the South-Eastern Norway Regional Health Authority (grant number 850703-6051-39788). JA and RLD are supported by National Institutes of Health Grants R01 GM111819 and R01 GM084453.

References

1. Georgiou G, Ippolito GC, Beausang J, Busse CE, Wardemann H, Quake SR. The promise and challenge of high-throughput sequencing of the antibody repertoire. Nat Biotechnol. 2014; 32:158–168. [PubMed: 24441474]
2. Reichert JM. Antibodies to watch in 2016. MAbs. 2016; 8:197–204. [PubMed: 26651519]
3. Correia BE, Bates JT, Loomis RJ, Baneyx G, Carrico C, Jardine JG, Rupert P, Correnti C, Kalyuzhniy O, Vittal V, et al. Proof of principle for epitope-focused vaccine design. Nature. 2014; 507:201–6. [PubMed: 24499818]

4. Al-Lazikani B, Lesk AM, Chothia C. Standard conformations for the canonical structures of immunoglobulins. J Mol Biol. 1997; 273:927–948. [PubMed: 9367782]

5. Weitzner BD, Kuroda D, Marze N, Xu J, Gray JJ. Blind prediction performance of RosettaAntibody 3.0: Grafting, relaxation, kinematic loop modeling, and full CDR optimization. Proteins Struct Funct Bioinforma. 2014; 82:1611–1623.

6. Almagro JC, Teplyakov A, Luo J, Sweet RW, Kodangattil S, Hernandez-Guzman F, Gilliland GL. Second antibody modeling assessment (AMA-II). Proteins Struct Funct Bioinforma. 2014; 82:1553–1562.

7. Bujotzek A, Dunbar J, Lipsmeier F, Schäfer W, Antes I, Deane CM, Georges G. Prediction of VH-VL domain orientation for antibody variable domain modeling. Proteins Struct Funct Bioinforma. 2015; 83:681–695.

8. Sircar A, Gray JJ. SnugDock: Paratope structural optimization during antibody-antigen docking compensates for errors in antibody homology models. PLoS Comput Biol. 2010; 6:e1000644. [PubMed: 20098500]

9. Alzari PM, Lascombe MB, Poljak RJ. Three-Dimensional Structure of Antibodies. Annu Rev Immunol. 1988; 6:555–580. [PubMed: 2454644]

10. Kunik V, Ofran Y. The indistinguishability of epitopes from protein surface is explained by the distinct binding preferences of each of the six antigen-binding loops. Protein Eng Des Sel. 2013; 26:599–609. [PubMed: 23754530]

11. Ponomarenko JV, Bourne PE. Antibody-protein interactions: benchmark datasets and prediction tools evaluation. BMC Struct Biol. 2007; 7:64. [PubMed: 17910770]

12. Kozakov D, Brenke R, Comeau SR, Vajda S. PIPER: An FFT-based protein docking program with pairwise potentials. Proteins Struct Funct Genet. 2006; 65:392–406. [PubMed: 16933295]

13. Brenke R, Hall DR, Chuang GY, Comeau SR, Bohnuud T, Beglov D, Schueler-Furman O, Vajda S, Kozakov D. Application of asymmetric statistical potentials to antibody-protein docking. Bioinformatics. 2012; 28:2608–2614. [PubMed: 23053206]

14. Chen R, Li L, Weng Z. ZDOCK: An initial-stage protein-docking algorithm. Proteins Struct Funct Genet. 2003; 52:80–87. [PubMed: 12784371]

15. Krawczyk K, Baker T, Shi J, Deane CM. Antibody i-Patch prediction of the antibody binding site improves rigid local antibody-antigen docking. Protein Eng Des Sel. 2013; 26:621–629. [PubMed: 24006373]

16. Sircar A, Chaudhury S, Kilambi KP, Berrondo M, Gray JJ. A generalized approach to sampling backbone conformations with RosettaDock for CAPRI rounds 13–19. Proteins Struct Funct Bioinforma. 2010; 78:3115–3123.

17. Méndez R, Leplae R, Lensink MF, Wodak SJ. Assessment of CAPRI predictions in rounds 3–5 shows progress in docking procedures. Proteins Struct Funct Bioinforma. 2005; 60:150–169.

18. O'Meara MJ, Leaver-Fay A, Tyka MD, Stein A, Houlihan K, Dimaio F, Bradley P, Kortemme T, Baker D, Snoeyink J, et al. Combined covalent-electrostatic model of hydrogen bonding improves structure prediction with Rosetta. J Chem Theory Comput. 2015; 11:609–622. [PubMed: 25866491]

19. Coutsias EA, Seok C, Jacobson MP, Dill KA. A kinematic view of loop closure. J Comput Chem. 2004; 25:510–528. [PubMed: 14735570]

20. Mandell DJ, Coutsias EA, Kortemme T. Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. Nat Methods. 2009; 6:551–552. [PubMed: 19644455]

21. Stein A, Kortemme T. Improvements to Robotics-Inspired Conformational Sampling in Rosetta. PLoS One. 2013; 8:e63090. [PubMed: 23704889]

22. Raveh B, London N, Schueler-Furman O. Sub-angstrom modeling of complexes between flexible peptides and globular proteins. Proteins Struct Funct Bioinforma. 2010; 78:2029–2040.

23. London N, Raveh B, Cohen E, Fathi G, Schueler-Furman O. Rosetta FlexPepDock web server - High resolution modeling of peptide-protein interactions. Nucleic Acids Res. 2011; 39:W249–W253. [PubMed: 21622962]

24. Meiler J, Baker D. ROSETTALIGAND: Protein-small molecule docking with full side-chain flexibility. Proteins Struct Funct Genet. 2006; 65:538–548. [PubMed: 16972285]

25. Johnson G, Wu TT. Kabat database and its applications: 30 years after the first variability plot. Nucleic Acids Res. 2000; 28:214–8. [PubMed: 10592229]

26. Nowak J, Baker T, Georges G, Kelm S, Klostermann S, Shi J, Sridharan S, Deane CM. Length-independent structural similarities enrich the antibody CDR canonical class model. MAbs. 2016; 8:751–60. [PubMed: 26963563]

27. Marze NA, Gray JJ. Improved prediction of antibody VL–VH orientation. Protein Eng Des Sel. 2016:gzw013. In Press.

28. Canutescu AA, Dunbrack RL. Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Sci. 2003; 12:963–72. [PubMed: 12717019]

29. Wang C, Bradley P, Baker D. Protein–Protein Docking with Backbone Flexibility. J Mol Biol. 2007; 373:503–519. [PubMed: 17825317]

30. Bradley P, Misura KMS, Baker D. Toward High-Resolution de Novo Structure Prediction for Small Proteins. Science (80-). 2005; 309:1868–1871.

31. Misura KMS, Baker D. Progress and challenges in high-resolution refinement of protein structure models. Proteins Struct Funct Genet. 2005; 59:15–29. [PubMed: 15690346]

32. Weitzner BD, Dunbrack RL, Gray JJ, Al-Lazikani B, Lesk AM, Chothia C, Almagro JC, Beavers MP, Hernandez-Guzman F, Maier J, et al. The Origin of CDR H3 Structural Diversity. Structure. 2015; 23:302–311. [PubMed: 25579815]

33. Weitzner BD, Gray JJ. Accurate structure prediction of CDR H3 loops enabled by a novel structure-based C-terminal 'kink' constraint. J Immunol. 2016 To Appear.

34. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D. Protein–Protein Docking with Simultaneous Optimization of Rigid-body Displacement and Side-chain Conformations. J Mol Biol. 2003; 331:281–299. [PubMed: 12875852]

35. Chaudhury S, Gray JJ. Conformer Selection and Induced Fit in Flexible Backbone Protein–Protein Docking Using Computational and NMR Ensembles. J Mol Biol. 2008; 381:1068–1087. [PubMed: 18640688]

36. Kuroda D, Gray JJ. Shape complementarity and hydrogen bond preferences in protein-protein interfaces: implications for antibody modeling and protein-protein docking. Bioinformatics. 2016; :btw197.doi: 10.1093/bioinformatics/btw197

37. Nivon LG, Moretti R, Baker D. A Pareto-Optimal Refinement Method for Protein Design Scaffolds. PLoS One. 2013; 8:e59004. [PubMed: 23565140]

38. Sivasubramanian A, Chao G, Pressler HM, Wittrup KD, Gray JJ. Structural model of the mAb 806-EGFR complex using computational docking followed by computational and experimental mutagenesis. Structure. 2006; 14:401–414. [PubMed: 16531225]

39. Simonelli L, Beltramello M, Yudina Z, Macagno A, Calzolai L, Varani L. Rapid Structural Characterization of Human Antibody-Antigen Complexes through Experimentally Validated Computational Docking. J Mol Biol. 2010; 396:1491–1507. [PubMed: 20053355]

40. Blech M, Seeliger D, Kistler B, Bauer MMT, Hafner M, Horer S, Zeeb M, Nar H, Park JE, Hörer S. Molecular structure of human GM-CSF in complex with a disease-associated anti-human GM-CSF autoantibody and its potential biological implications. Biochem J. 2012; 447:205–215. [PubMed: 22839360]

41. Thornburg NJ, Nannemann DP, Blum DL, Belser JA, Tumpey TM, Deshpande S, Fritz GA, Sapparapu G, Krause JC, Lee JH, et al. Human antibodies that neutralize respiratory droplet transmissible H5N1 infuenza viruses. J Clin Invest. 2013; 123:4405–4409. [PubMed: 23999429]

42. Ó Conchúir S, Barlow KA, Pache RA, Ollikainen N, Kundert K, O'Meara MJ, Smith CA, Kortemme T. A Web resource for standardized benchmark datasets, metrics, and rosetta protocols for macromolecular modeling and design. PLoS One. 2015; 10:e0130433. [PubMed: 26335248]

43. Zemlin M, Klinger M, Link J, Zemlin C, Bauer K, Engler JA, Schroeder HW, Kirkham PM. Expressed Murine and Human CDR-H3 Intervals of Equal Length Exhibit Distinct Repertoires that Differ in their Amino Acid Composition and Predicted Range of Structures. J Mol Biol. 2003; 334:733–749. [PubMed: 14636599]

44. Sela-Culang I, Alon S, Ofran Y. A Systematic Comparison of Free and Bound Antibodies Reveals Binding-Related Conformational Changes. J Immunol. 2012; 189:4890–4899. [PubMed: 23066154]

45. Kuroda D, Gray JJ. Pushing the backbone in protein-protein docking. Structure. 2016 In Press.

46. Yamashita K, Ikeda K, Amada K, Liang S, Tsuchiya Y, Nakamura H, Shirai H, Standley DM. Kotai Antibody Builder: Automated high-resolution structural modeling of antibodies. Bioinformatics. 2014; 30:3279–3280. [PubMed: 25064566]

47. Shirai H, Ikeda K, Yamashita K, Tsuchiya Y, Sarmiento J, Liang S, Morokata T, Mizuguchi K, Higo J, Standley DM, et al. High-resolution modeling of antibody structures by a combination of bioinformatics, expert knowledge, and molecular simulations. Proteins Struct Funct Bioinforma. 2014; 82:1624–1635.

48. Marcatili P, Olimpieri PP, Chailyan A, Tramontano A. Antibody structural modeling with prediction of immunoglobulin structure (PIGS). Nat Protoc. 2014; 9:2771–83. [PubMed: 25375991]

49. Leem J, Dunbar J, Georges G, Shi J, Deane CM. ABodyBuilder: automated antibody structure prediction with data-driven accuracy estimation. MAbs. 2016; :00–00. DOI: 10.1080/19420862.2016.1205773

50. Schrödinger, L. The PyMOL Molecular Graphics System, Version 1.8. 2015.

51. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE. UCSF Chimera--a visualization system for exploratory research and analysis. J Comput Chem. 2004; 25:1605–1612. [PubMed: 15264254]

52. Chen VB, Davis IW, Richardson DC. KiNG (Kinemage, Next Generation): A versatile interactive molecular and scientific visualization program. Protein Sci. 2009; 18:2403–2409. [PubMed: 19768809]

53. Lyskov S, Chou FC, Conchúir S, ÓDer BS, Drew K, Kuroda D, Xu J, Weitzner BD, Renfrew PD, Sripakdeevong P, et al. Serverification of Molecular Modeling Applications: The Rosetta Online Server That Includes Everyone (ROSIE). PLoS One. 2013; 8:e63906. [PubMed: 23717507]

54. North B, Lehmann A, Dunbrack RL. A new clustering of antibody CDR loop conformations. J Mol Biol. 2011; 406:228–256. [PubMed: 21035459]

55. Adolf-Bryfogle J, Xu Q, North B, Lehmann A, Dunbrack RL. PyIgClassify: a database of antibody CDR structural classifications. Nucleic Acids Res. 2015; 43:D432–8. [PubMed: 25392411]

56. Abhinandan KR, Martin ACR. Analysis and improvements to Kabat and structurally correct numbering of antibody variable domains. Mol Immunol. 2008; 45:3832–3839. [PubMed: 18614234]

57. Honegger A, Plückthun A. Yet another numbering scheme for immunoglobulin variable domains: an automatic modeling and analysis tool. J Mol Biol. 2001; 309:657–70. [PubMed: 11397087]

58. Lefranc MP, Pommié C, Ruiz M, Giuducelli V, Foulquier E, Truong L, Thouvenin-Contet V, Lefranc G. IMGT unique numbering fro immunoglobulin and T cell receptor variable domains and Ig superfamily V-like domains. Dev Comp Immunol. 2003; 27:55–77. [PubMed: 12477501]

59. Kabat, EA., Te Wu, T., Foeller, C., Perry, HM., Gottesman, KS. Sequences of Proteins of Immunological Interest. NIH Publication; 1991.

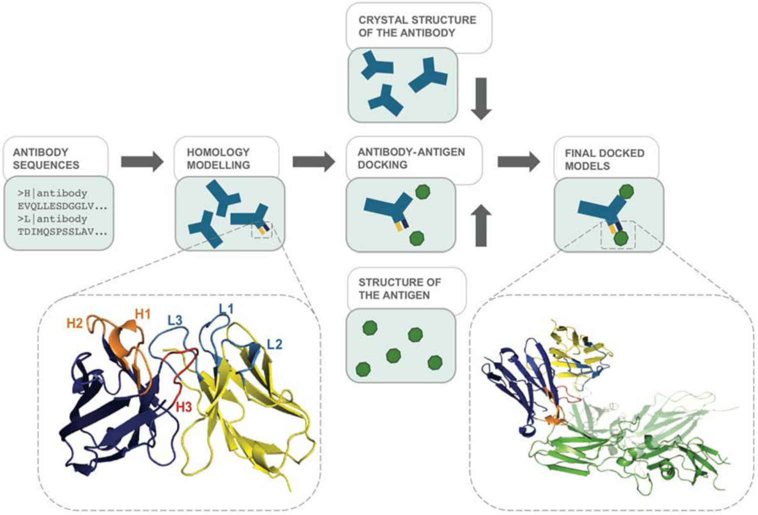

Figure 1.

A schematic of the modeling protocols (full flow charts for Rosetta Antibody and Rosetta SnugDock are available in the original publications). The structure on the left shows the Fv antibody domains predicted by homology modeling (heavy chain in dark blue with CDR H1 and H2 loops in orange and CDR H3 loop in red; light chain in yellow with its CDR loops in light blue). The structure on the right depicts an antibody–antigen structure output by docking (antigen in green).

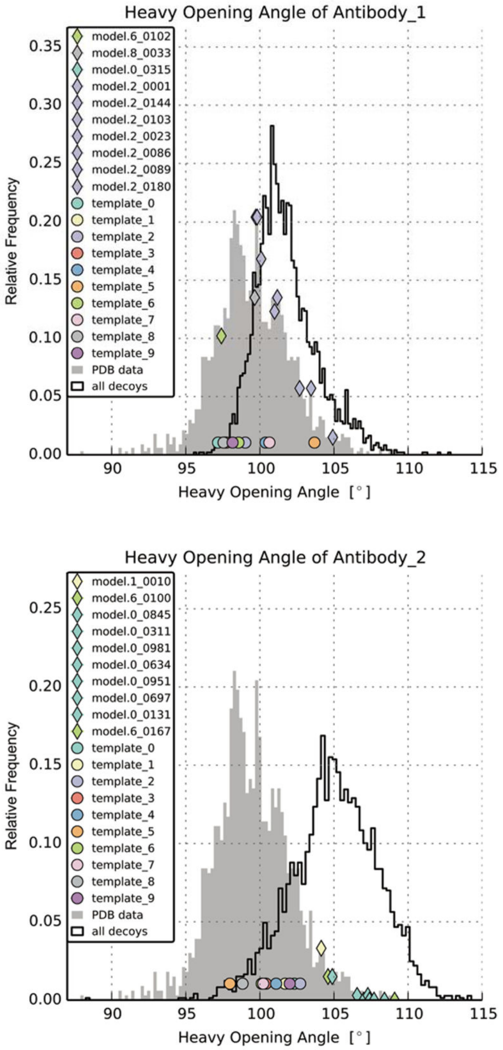

Figure 2.

Example output of plot_LHOC.py. The two plots show distributions of the Heavy Opening Angle[27] as obtained by plot_LHOC.py for two different antibodies. The 10 distinct light-heavy orientation templates are represented by the circles. The ten top-scoring models after H3 loop modeling are represented by the diamonds with the fill color corresponding to the starting template; in the legend, these points are ordered from smallest to largest metric value. For Antibody_1, the angles sampled by Rosetta overlap with the angles observed in antibody crystal structures. The ten top-scoring models are close to the center of the distribution. In Antibody_2, most of the angles sampled are found rarely or not at all in antibody crystal structures. The ten top-scoring models are also shifted to larger angles than typically found in antibodies. For Antibody_2, the user might consider trying alternate light-heavy orientation templates (Step 10).

Table 1

Local Ab–Ag docking benchmark results

Co-crystal PDB IDs indicate the native complex. PDB IDs listed under the “Type” column also indicate the use of unbound (U) or bound (B) component structures (as available). Model quality is defined by the CAPRI ranking criteria represented by a number of stars or a zero (0). Three, two, and one star(s) indicate high, medium, and acceptable quality, respectively, and a zero indicates incorrect models. A high quality model meets the criterion ($f_{nat} \geq 0.5$ && (Lrmsd ≤ 1.0 Å || Irmsd ≤ 1.0 Å)), a medium quality model meets the criterion ($f_{nat} \geq 0.3$ && (Lrmsd ≤ 5.0 Å || Irmsd ≤ 2.0 Å) && quality ≠ high), an acceptable quality model meets the criterion ($f_{nat} \geq 0.5$ && (Lrmsd ≤ 10.0 Å || Irmsd ≤ 4.0 Å) && quality ≠ (high || medium)), and an incorrect model meets the criterion (quality ≠ (high || medium || acceptable)). For the “Rigid-Body” and “SnugDock” columns, the quality of the lowest-scoring model, by interface energy, is reported (an “f” indicates a strong energy funnel defined as five or more of the ten lowest-scoring models being medium quality or better). Ensemble SnugDock simulations were run with multi-template grafting and a CDR H3 kink constraint. CAPRI Summary lines summarize model quality for all targets. CAPRI Summary Top 10 takes the highest-quality model from the ten lowest-scoring models.

Co-crystal (PDB ID)	Type (Ab-Ag)	CDR H3 Length	Rigid-Body Dock Xtal	Rigid-Body Dock Model	SnugDock	Ensemble SnugDock
1mlc	U(1mlb)-U(1lza)	7	0	0	**	*
1ahw	U(1fgn)-U(1boy)	8	0	*	*	**
1jps	U(1jpt)-U(1tfh)	8	**	*	*	**
1wej	U(1qbl)-U(1hrc)	8	0	0	0	*
1vfb	U(1vfa)-U(8lyz)	8	*	*	0	*
1bql	B-U(1dkj)	7	*** f	0	*	0
1k4c	B-U(1jvm)	9	** f	0	*	**
2jel	B-U(1poh)	9	*** f	0	**	*
1jhl	B-U(1ghl)	9	*** f	0	0	**
1nca	B-U(7nn9)	11	*** f	0	0	*
2bdn	B-B	8	*** f	*	*	*
1ynt	B-B	9	*** f	*** f	** f	*** f
2aep	B-B	9	*** f	**	*	0
2b2x	B-B	10	*** f	*	0	**
1ztx	B-B	10	*** f	*	** f	0
	CAPRI Summary Top Model		9***/2**/1*	1***/1**/6*	0***/4**/6*	1***/5**/6*
	CAPRI Summary Top 10 Models		9***/4**/2*	1***/6**/8*	1***/10**/4*	2***/11**/2*
	No. of Funnels		10	1	2	1

Table 2

Troubleshooting table

STEP	PROBLEM	POSSIBLE REASON	SOLUTION
EQUIPMENT	Rosetta does not compile.	Likely to be related to the specific computer operating system and configuration	Seek help on the Rosetta forums, www.rosettacommons.org/forum
4	Rosetta Antibody encounters error “ `sh: blastp: command not found”`	The blastp executable is not installed or not in in your $PATH	On the command line, try ‘ `which blastp`‘ to check if your system has it installed. If needed, download and install BLAST or/and add blastp to your PATH (`export PATH=$PATH:/path/to/blastp/`). You can also specify the path using the command line flag `-antibody:blastp /my/path`
4	Rosetta Antibody encounters encounters “ `BLAST Database error`”	The blastp database is not specified, and Rosetta Antibody is not finding it in the default location (`$ROSETTA/tools/antibody/blast_database/`)	Specify the grafting database location with `-antibody:grafting_database /database/location`
4	Rosetta Antibody produces BLAST output (e.g. `grafting/orientation.align`) but does not produce structural models (e.g. `model.0.pdb`)	Your version of BLAST+ may be out of date.	Download a compatible version of BLAST+ (version 2.2.28 or later). See Materials section.
4	Regular expression failure for CDR identification	Mutations in regions of the chain that Rosetta expects to be conserved prevent the sequence from being split into structural segments correctly.	Check your antibody sequence against the printed regular expression used to detect the CDR. To accommodate unusual sequences, the regular expressions can be altered by changing the file `database/protocol_data/antibody/cdr_regex.txt`.
17	SnugDock reports “ERROR: Could not find disulfide partner for residue 23”	A disulfide bond was disrupted during docking.	You can disable disulfide bond detection with the flag `-detect_disulf false`
17	SnugDock reports “ERROR: ReturnSidechainMover used with poses of different sequence; aborting”	The structures in the ensemble are not consistent.	Make sure that all sturctures have identical length chains and that if there are multiple chains, those chains appear in a consistent order.
17	SnugDock reports error `“chains are not named correctly or are not in the expected order”`	Input PDB does not contain chains in correct order (light, heavy, then antigen) or chain IDs are not L, H, and A.	Adjust chain order in input PDB or specify chain IDs with the `–partners AB_C` flag, where A, B and C are the light, heavy, and antigen chain IDs, respectively.
2–17	Other Rosetta errors.		Seek help on the Rosetta forums, www.rosettacommons.org/forum
2–17	Common fixes		• Check for misspellings • Check paths are correct • Check FASTA formatting • Check PDB formatting