



HHS Public Access

Author manuscript

J Comput Chem. Author manuscript; available in PMC 2018 February 15.

Published in final edited form as:

J Comput Chem. 2017 February 15; 38(5): 276–287. doi:10.1002/jcc.24679.

Residue-Centric Modeling and Design of Saccharide and Glycoconjugate Structures

Jason W. Labonte^[a], Jared Adolf-Bryfogle^[b], William R. Schief^{[b],[c]}, and Jeffrey J. Gray^[a]

^[a]Department of Chemical & Biomolecular Engineering, Johns Hopkins University, 3400 N. Charles St., Baltimore, Maryland, U.S.A. 21218

^[b]Department of Immunology and Microbial Science and IAVI Neutralizing Antibody Center, The Scripps Research Institute, La Jolla, CA 92037

^[c]The Ragon Institute of Massachusetts General Hospital, Massachusetts Institute of Technology and Harvard University, Cambridge, MA 02139

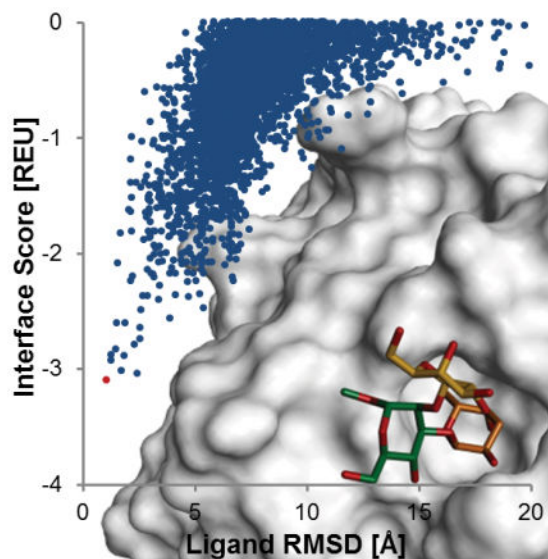
Graphical Abstract

Carbohydrates are present everywhere in nature, possessing a vast array of structural diversity, yet historically, they have been challenging to model. RosettaCarbohydrate is a new tool for researchers studying the form and function of carbohydrate structures. The framework integrates with Rosetta's successful modeling and design suite and addresses challenges unique to glycans. This article describes the development of the framework and highlights its applications, including loop modeling and glyco-ligand docking.

Correspondence to: Jeffrey J. Gray.

Author Contributions

J.W.L.: Conceived, designed, and wrote source code for the RosettaCarbohydrate framework; prepared and ran the docking benchmark; analyzed data; prepared figures; authored the paper. J.A.-B.: Contributed improvements to the RosettaCarbohydrate framework; conceived, designed, and wrote source code for the LinkageConformerMover and related classes; revised the manuscript. W.R.S.: Directed the project; conceived and designed the LinkageConformerMover; revised the manuscript. J.J.G.: Conceived and directed the project; analyzed data; revised the manuscript.



Introduction

Carbohydrates are the most abundant class of molecule on earth,^[1] functioning as oligosaccharide small molecules, polysaccharide structures, and glycoconjugates in a wide array of processes crucial for life.^[2] The diversity of glycoforms is enormous. Eukaryotic cells synthesize thousands of distinct forms from just the nine most-common monosaccharide subunits.^[3,4] Diversity is introduced to standard glycan classes (high-mannose, hybrid, and complex) through repeats, branching patterns, elaboration with sugars such as fucose or sialic acid, and sugar modifications.

As “form follows function,” a structural understanding of these complicated molecules is required to appreciate fully their role in biomolecular pathways. Unfortunately, the characterization methods currently available make structure-determination difficult. Whether as ligands or as conjugates, the electron density of glycans is often not resolved in crystal structures because of their inherent flexibility. Computational methods to assist in refining such crystal structures would be welcome, and protocols for studying glycan interactions *in silico* are needed.

However, computational modeling of carbohydrates has not proven straightforward,^[5,6] though there has been significant progress. For small systems, such as individual mono- or disaccharides, quantum mechanical (QM) methods have been used to model carbohydrate structures.^[7] For modeling larger systems of glycans, several computational options are currently available. Carbohydrate molecular dynamics (MD) forcefields include GLYCAM,^[8] CHARMM,^[9] OPLS-AA-SEI,^[10] GROMOS45A3/4,^[11] and MM4.^[12] Several software packages have been used to dock carbohydrates, among them AutoDock^[13] and AutoDock Vina,^[14] DOCK,^[15] FlexX,^[16] Glide,^[17] and Gold.^[18] The majority of docking applications reported thus far in the literature have involved the docking of

carbohydrate ligands and not the interactions of glycoproteins with other proteins or glycoproteins.

Currently, there are no computational methods specifically for designing glycoproteins for particular functions. One might desire a means both of computationally designing amino acid residues around a particular glycan and of designing a conjugated glycan—produced with glycoengineering techniques^[19]—for a particular protein system.

The Rosetta structure prediction and design suite^[20] is an ideal platform for addressing these challenges. Rosetta has solved the structures of proteins^[21] and RNA;^[22] been used to refine NMR,^[23] crystal,^[24] and cryo-electron microscopy^[25] structures; modeled antibody loops;^[26] and docked both protein–protein^[27] and protein–ligand^[28] complexes. Rosetta has successfully designed unique sequences to match a fixed peptide backbone;^[29,30] novel protein folds,^[31,32] including with functional sites;^[33] enzyme active sites;^[34,35] protein–protein interfaces;^[36] RNA sequences;^[37] and peptides to modify mineral growth.^[38,39] Rosetta has also been expanded to model non-canonical and non-peptide polymers.^[40]

How Rosetta Differs from Other Approaches

In contrast to quantum or molecular mechanics/dynamics approaches, Rosetta is “residue-centric”^[20] instead of “atom-centric”. That is, a residue is the primary unit for scoring and manipulation of a structure. Rosetta represents all atoms within the context of their residues instead of as individual units. This approach has several advantages. A residue can be classified with other molecular fragments that share certain chemical properties. From a computational point of view, this organization leads to a data structure that can store chemical and nomenclature information beyond simple atom coordinates and charges. Related residues can share data common to their “type”, which allows rapid packing—wherein residues with shared backbone structure have their side chains substituted with those of other rotamers—and design—wherein residues have their side chains swapped with those from related residues. Finally, this data organization permits quick insertion or deletion of chains of residues, such as loops, since the structure of a macromolecule can be treated as a tree of residue units.^[20]

The RosettaCarbohydrate Framework

In earlier work in collaboration with other Rosetta labs, we described how Rosetta’s residue-centric framework could be generalized and adapted to model alternative-backbone polymers.^[40] Much of the underlying code in the Rosetta codebase had originally operated on the assumption of the constant, repeating N–C α –C backbone of peptides, but creative use of particular features of Rosetta’s topology files and patching system now allow for modeling of virtually any polymer.^[40,41] This current work expands on this framework with specific consideration to the challenges involved in modeling oligo- and polysaccharides.

In this report, we describe our efforts to make Rosetta “carbohydrate-ready”, creating a tool to empower researchers solving problems in the growing fields of glycobiology and glycoengineering. We have established the RosettaCarbohydrate framework to provide alternative and complementary methods for general modeling and docking applications involving oligomeric and polymeric carbohydrate ligands and glycoconjugates. Here, we

discuss general problems in sampling, scoring, and nomenclature as related to carbohydrate modeling, and we outline our strategies for overcoming some of these difficulties. We highlight new features of ring sampling and virtual glycosylation and present data that benchmark protein–glyco-ligand docking and novel glycan “loop” modeling.

General Modeling Concerns

Any macromolecular structural modeling problem, from simple minimization to more complicated cases, such as loop refinement or docking, requires sufficient *sampling* of coordinate space and accurate *scoring* to approximate a system’s energy. A more often overlooked concern is the choice of how to represent the molecule to the modeling software, that is, how to communicate by means of proper *nomenclature* the precise complex to be modeled. In the following paragraphs, we discuss these three issues—sampling, scoring, and nomenclature—in regards to carbohydrates. In the methods section we describe in detail how we have addressed these matters in Rosetta.

Sampling

Carbohydrates are flexible molecules, because of their many degrees of freedom (DoFs), and this flexibility contributes to their difficulty to model. Rosetta algorithms primarily sample torsion angle DoFs instead of manipulating Cartesian coordinates directly. The number of torsional DoFs in a typical monosaccharide residue is much greater than the number found in an amino acid residue.^[42] Oligo- and polypeptides effectively have only two backbone torsion angles per residue, ϕ and ψ , since the third angle, ω , is fixed at one of two acceptable values (Figure 1a). Oligo- and polysaccharides also have main-chain torsions of ϕ , ψ , and sometimes ω . However, peptide chains are almost always linear, whereas saccharide chains are very commonly branched. Branching poses multiple sampling challenges. Sampling a torsion angle upstream of a branch point results in multiple downstream effects, instead of effects on a single polymer chain. This increases the chances of clashing. Moreover, it requires more thorough sampling before the branch point, as the positioning of the tips of branches depends heavily on the orientation of the stem. In addition, saccharide residues can adopt linear or multiple cyclic forms, and potential conformer shifts among ring forms provide additional backbone flexibility to the polysaccharide. This flexibility occurs through internal ring torsion angles labeled ν and defined by the four ring atoms about the bond (Figure 1b). Peptide residues, on the other hand, do not have aliphatic backbone rings, except proline, but its ring form is determined by its main-chain torsion angles.^[43] Finally, peptides only have one side chain per residue (designated with the label χ), in contrast to saccharide compounds, which have multiple side chains per residue (Figure 1). Moreover, only five of the standard amino acid residues (leucine, isoleucine, methionine, arginine, and lysine) have more than three rotatable side-chain torsions. All of the common, unmodified aldohexopyranose residues have four, when internally linked at non-exocyclic hydroxyl groups, and the common modified sugars *N*-acetylglucosamine and 5-acetylneuraminate (sialic acid) have five and nine, respectively, if internally linked. In peptides, since there is only one side chain, the torsional preferences are coupled; in saccharide residues, most of the side-chain motions are independent. To sample all of the side-chain conformations of a

terminal 5-acetylneuraminate, which is very common in human glycans, would require visiting n^{10} conformations!

Scoring

Rosetta uses a combination of physics-based and statistics-based scoring methods in its all-atom, implicit-solvent scoring function.^[44,45] Physics-based terms in the scoring function include van der Waals attractive and Pauli repulsive Lennard-Jones potentials, a Coulombic electrostatic potential, and the Lazaridis–Karplus implicit solvation potential. Statistics-based terms include hydrogen-bonding potentials, a peptide backbone-dependent rotamer probability potential,^[46] and Ramachandran backbone propensity potential.^[47]

Most of Rosetta's statistics-based terms do not apply to carbohydrate structures, and most of the atom types used in parameterization of the residues were derived empirically with peptides and peptide-like molecules and not sugars. Carbohydrates are known to have complicated electronic effects, such as the anomeric effect,^[42] and binding of proteins with glycans may involve such interactions as nonconventional C–H...O hydrogen bonds^[48] or π interactions.^[42]

Nomenclature

The third general modeling concern to overcome is nomenclature. While perhaps not a significant issue in the case of modeling typical biopolymers of other classes—DNA and RNA for example have only four nucleobase residues each and canonical peptides have 20—it is not a straightforward task to designate to modeling software which monosaccharide residues should be modeled. Whereas one can represent a full protein sequence with a series of one-letter or three-letter codes, a polysaccharide sequence is necessarily more complex and must include each base monosaccharide residue's anomeric stereochemistry, ring size, enantiomer, linkages, and any sugar modifications. For example, the term “GlcNAc”, while commonly used in literature to refer to α -D-2-deoxy-2-acetylaminoglucopyranose (IUPAC designation^[49] α -D-GlcpN²Ac), is in fact ambiguous and could refer to 36 distinct glucosamines—two for the specific anomer times two for the stereochemical designation times three for the ring form times three for the location of the *N*-acetyl group. This number increases to 108 if linkage is included as part of the designation (*e.g.*, \rightarrow 4)- α -D-GlcpN²Ac to indicate that a downstream residue is attached to this residue's O4) and grows even larger if the sugar residue is a branch point.

The data format of the Protein Data Bank (PDB) unfortunately assigns only a three-character column for specifying a residue type within a structure, which furthers this nomenclature problem. This limitation forces one to deviate from the three-letter abbreviations in wide use for the more common sugars. For example, since the PDB uses the code GLC to refer specifically to α -D-glucopyranose, any other glucoses or glucose derivatives must be assigned different—and usually unintuitive—three-letter codes.

Some groups have therefore devised a system of alternative three-letter codes to uniquely identify distinct saccharide residues. The GLYCAM format, for example, uses single characters to designate linkage, base monosaccharide, and anomer. Upper or lower case designate anomeric state^[50] Such codes suffice for commonly observed natural sugars but

are mathematically incapable of covering the full range of residues one might wish to model when rare natural sugars, synthetic sugars, and sugar modifications are included. Moreover, required use of case can be problematic, since some computational programs are not case-sensitive. Such a system also leads to conflicts with already existing three-letter codes within the PDB for non-canonical amino acids, solvent and other small molecules, ions, and other polymeric residue types.

The solution of Im research group was to avoid nomenclature altogether and to automatically prepare saccharide residues and linkages for direct input into CHARMM simulation software, such as the CHARMM-GUI,^[51] using atom coordinates, types, and bonding from .pdb file HETATM and CONECT records.^[52] Notably, the CHARMM format does not limit its residue names to three characters. While a much-needed tool for the field, the nomenclature problem still remained for Rosetta, which interfaces with .pdb files, not CHARMM ones.

RosettaCarbohydrate Development

ResidueType and CarbohydrateInfo

To integrate carbohydrate functionality into the Rosetta suite, we took advantage of its object-oriented code design.^[20] Rosetta stores data about a structural model primarily in a Pose object, whose primary data are stored in an array of Residue objects (Figure 2). These contain both conformational and chemical information. Properties specific to a particular monomeric chemical moiety are stored within a data structure called a ResidueType (Figure 2). Such properties include nomenclature, classifications, stereochemical information, and chemical functionality. Monosaccharide residues have a far greater diversity of properties than do amino acid residues, such as ring size, anomeric state, specific sugar modifications, and branch points; therefore, we expanded the ResidueType data structure by introducing into it two additional sub-structures—a ResidueProperties object to manage properties in general and a CarbohydrateInfo object specifically for saccharide ResidueTypes (Figure 2).

The CarbohydrateInfo object contains methods for deriving secondary properties from primary topology file data and for nomenclature output. The inclusion of the CarbohydrateInfo object within ResidueType means that a saccharide residue knows its chemical functionality and classification. This feature allows for glycoengineering design applications, where one monosaccharide subunit can be “swapped” with another of similar properties in a design protocol. Properties stored within CarbohydrateInfo include full and short-form IUPAC names, position of the anomeric carbon, number of carbons, stereochemistry (L or D), ring size, anomer (α or β), and a list of modifications. The RosettaCarbohydrate framework is not limited to aldohexopyranoses; CarbohydrateInfo can represent aldoses and ketoses, acyclic and cyclic sugars, sugars of sizes ranging from trioses to nonoses, and sugars with ring sizes ranging from oxiroses to septanoses (Figure 3).

We developed a singleton object called the CarbohydrateInfoManager (Figure 2) to handle data access to constant properties and data common to carbohydrates, such as nomenclature rules and common torsion angle definitions (*e.g.*, glycosidic ϕ , ψ , and ω definitions).

Branching

Unlike most amino acid and nucleic acid residue chains, oligo- and polysaccharide chains commonly branch. Rosetta's AtomTree and FoldTree^[20,27] had allowed only limited functionality for branching, such as for disulfide bonds and ubiquitination. We updated and expanded all Rosetta code for the handling of branching, including new methods of specifying torsional degrees of freedom across branch connections (through the MoveMap object) and correct handling of packing at branch-point residues. Moreover, we refactored Rosetta code that had assumed that a parent residue was always the previous residue in the sequence, which is only true for a linear polymer.

We further improved the input/output methods of Rosetta to properly read and interpret the LINK records from .pdb files from the Protein Data Bank, which are a standard record type for that file format. Rosetta now uses the information in the LINK records to build the FoldTree and AtomTree correctly for branched structures, including saccharides.

Input/Output and Nomenclature

For the input of .pdb files, we have made Rosetta compatible with both the current list of PDB three-letter codes and the three-letter code system used with GLYCAM-based utilities.^[50] In addition, we created a means to specify the exact residue desired by means of an expanded, backwards-compatible use of .pdb file HETNAM records, which allows for even the most exotic sugars to be designated with precision. We refactored Rosetta's input code so that it now reads HETNAM records and abstracts information about which ResidueTypes it should use to build the model. Rather than a single HETNAM record being used for every occurrence of a single three-letter code within the .pdb file, Rosetta now allows for the specification of the exact residue identity for each HETNAM record, so that the same three-letter code can be used for distinct but related saccharide residues. (This same solution can now also be applied for any noncanonical residue for which there may be three-letter code conflicts and is not limited in any way to saccharide residues.) Comparisons of simple oligosaccharide .pdb files in all three formats (standard PDB, GLYCAM, and Rosetta) are shown in Figure S1 and Figure S2.

In addition to .pdb-file input, we have provided a means of reading IUPAC-format polysaccharide sequences,^[49] including branched polysaccharides and sugar modifications (Figure 3), and of reading the .gws file format used by the GlycoWorkbench software.^[53] The latter provides an easy means of converting between standard Consortium for Functional Glycobiology (CFG) topology diagrams^[54] and Rosetta.

For file output, we designed Rosetta to output .pdb files, as well as .gws files of glycans.

Backbone Sampling

As discussed above, backbone torsion angles for carbohydrate polymers include ϕ , ψ , ω , and ν angles. ϕ , ψ , and ω always fall along the main chain of the Rosetta AtomTree. Sampling of ν angles (internal ring torsions) changes the backbone but involves atoms both on and off the main chain. Because of this, they require special sampling methods.

Main-Chain Sampling—We refactored the SmallMover and ShearMover main-chain sampling code to access the CarbohydrateInfo object for determining which main-chain torsion angles in Rosetta's FoldTree correspond to ϕ and ψ for a given residue (Figure 2). Thus, the Small- and ShearMover, which sample ϕ and ψ , can be used in Rosetta algorithms “out-of-the-box” with both saccharides and glycoconjugates.

Incorporating elements of our previously published Glycan Relax protocol,^[55] we further designed a LinkageConformerMover to select statistically favorable ϕ , ψ , and ω angles during sampling stages of Rosetta algorithms. This new method specific to carbohydrate residues selects from known, preferred sets of glycosidic torsion angles for pairs of saccharide residues. For example, ϕ and ψ torsions for β -D-Galp-(1 \rightarrow 3)-D-GlcpNAc linkages statistically show a strong preference (nearly 100%) for ϕ values of $-74^\circ \pm 10^\circ$ and ψ values of $-132^\circ \pm 18^\circ$.^[56] The LinkageConformerMover thus samples in this region of torsion space for any occurrence of β -D-Galp-(1 \rightarrow 3)-D-GlcpNAc in a structure. This sampling, when combined with the more general small and shear moves, brings about a more efficient determination of natural surface glycan structures.

The LinkageConformerMover code functions as follows. First, we constructed a mapping of linkages to LinkageConformerData structures, each of which represents a single glycosidic-bond conformation. Each LinkageConformerData object contains a conformation's reducing-end and non-reducing-end residues, population percentage, and mean and standard deviation values for ϕ , ψ , and any applicable ω torsions. We first used data from a series of papers by Petrescu et al., which include conformations of β -D-GlcpNAc-Asn for the modeling of N-linked glycans.^[56,57] The CarbohydrateInfoManager object described above loads and controls access to the mapping of LinkageConformerData (Figure 2)

When acting on a structure, the LinkageConformerMover selects a random carbohydrate residue and locates its parent residue in the direction of the reducing end. By means of the CarbohydrateInfoManager, the Mover searches for conformer data present for a linkage between the two residues. If conformer data are found, the Mover selects a conformer and sets each backbone torsion angle of the current residue to the mean of the value, plus or minus a uniform random value within a given number of standard deviations of the mean (Figure 2). The specific conformer selected can be chosen uniformly from among all conformers for the found linkage, or it can be selected based on population weight.

When the LinkageConformerMover does not find a residue pair within the map of LinkageConformerData, a conformer is selected from the minima of the appropriate CarboHydrate-Intrinsic (CHI) Energy Functions,^[58,59] which we describe below.

Ring-Conformer Sampling—As peptide chains and nucleic acids have a limited degree of ring-puckering flexibility, no general method had been developed for sampling ring conformations in Rosetta. Within the ResidueType class, we thus designed and implemented a RingConformerSet object (Figure 2), which stores the current and all possible ideal ring conformers for any given ring size.

Each RingConformer object in the Set represents a single ideal conformer and stores both specific IUPAC name (*e.g.*, 4C_1 for the most-stable glucopyranose ring form) and general name (*e.g.*, “chair”), degeneracy, Cremer–Pople (C–P) parameters,^[60] and ideal torsion (ν) and bond (τ) angles. Specific ring conformers can be accessed by IUPAC name or by C–P parameters. We included all idealized ring conformers for six-membered rings in the Rosetta database (Table S1).

We also wrote a generalized method for sampling ring conformations of cyclic residues as part of backbone sampling, which we call the RingConformationMover (Figure 2). It selects idealized ring conformers from the RingConformerSet, such as the 4C_1 chair conformer found in most common glycans. The RingConformationMover selects residue-specific, energetically favored ring conformers for each residue when known. A list of these preferred ring conformers are stored in each residue’s topology/parameter file in the Rosetta database. The torsion angles and bond angles are set from the RingConformer data.

Additionally, we expanded Rosetta’s MoveMap object to communicate to other objects within Rosetta protocols whether internal ring torsion angles are permitted to move during sampling steps.

While designed with carbohydrates in mind, these new Rosetta objects will also be useful for sampling the conformations of non-canonical amino acid residues and other moieties containing rings.

Side-Chain Sampling

Side-chain moves change χ angles and any torsion angles found on sugar modifications. As mentioned above, Rosetta typically uses a rotamer-library approach to refine side-chain torsional space. For amino acid residues, it relies on statistical data of rotamer probabilities for each residue type. Because the majority of “side chains” in carbohydrates are simple hydroxyls, and structures with proton-resolution are so rare, such rotamer statistics are next to impossible to calculate and likely unneeded. Instead, we simply have exhaustively generated rotamers for every combination of staggered/non-eclipsed side-chain conformations per residue. We allow minimization to find the most favorable side-chain conformations from the staggered ideals given by packing.

Scoring Function

For peptides, Rosetta’s Pauli repulsive term is split into both inter-residue and intra-residue components and the weight for the intra-residue repulsion term is a small fraction of that for the inter-residue term to avoid double-counting with the term for rotamer potential. For carbohydrates and other non-peptide residues, which do not have a term for rotamer potential, we use a separate intra-residue repulsive term with the same weight as that for the inter-residue term, and we add an additional, carbohydrate-specific scoring method for glycosidic bonding.

We implemented the CHI Energy Function developed in the laboratory of Robert Woods. This function was determined from QM calculations involving various isomers of *O*-linked tetrahydropyran oligomers and confirmed by statistical data.^[58,59] This new scoring method

is analogous to the Ramachandran scoring method used within Rosetta for peptide bonds. For scoring glycosidic ϕ torsion angles, the function depends on the stereochemistry of the anomeric carbon, and for scoring the ψ angle, it depends on whether the connecting oxygen atom is in the axial or equatorial position. (Omega angles were added to the CHI scoring function^[59] during the writing of this manuscript and have not yet been implemented in Rosetta.) The anomeric stereochemistry is read from CarbohydrateInfo. In preliminary tests, we found that the inclusion of this term improved results in ligand docking (Figure 4).

We wrote a tutorial using Python that demonstrates the above features, which can be downloaded from http://graylab.jhu.edu/~labonte/shared/RosettaCarbohydrates/RosettaCarbohydrate_Tutorial-Demo.pdf.

Carbohydrate-Specific Applications

Having adapted the core Rosetta code to consider issues of carbohydrate sampling, scoring, and nomenclature, we began adding new methodologies. Here we describe “virtual glycosylation”, carbohydrate “loop” modeling, and glyco-ligand docking. Since most saccharide residues are known to have approximately fixed ring conformations,^[61] we did not sample ν angles in any of these three applications.

Virtual Glycosylation

Since many glycoprotein structures in the PDB lack their native glycans or have incomplete or unclear density in some of the carbohydrate residues, we added functionality to Rosetta to generate glycosylated starting structures from non- or partially glycosylated models. We generate the glycan from an IUPAC string or file and affix it to the peptide at the requested position in a starting conformation with torsion angles pulled from the linkage conformer database. Glycans can be appended to a structure in *N*- (Figure 5), *O*- (Figure S3), or *C*- linkages or attached to other monosaccharide residues. Such a glycosylated peptide can be used as a starting point for other modeling applications, or protocols can be written to sample the heterogeneous array of glycans a peptide might be able to exhibit. As a starting point, we have added a small file library of common glycan sequences to the Rosetta database for use with this function.

“Loop” Modeling of Carbohydrates

When conjugated glycans are not fully resolved in crystal structures, it is usually monosaccharide residues nearest the non-reducing end(s) that are unresolved. On occasion, however, the non-reducing-end residue(s) may interact tightly with another portion of the structure, and the internal residues of the glycan instead remain unresolved.^[62] This latter scenario can be treated as a loop-building problem, where the missing residues can be appended and then the gap closed to form a starting model for either MD or further Rosetta applications.

Rosetta regularly uses loop-closure algorithms borrowed from robotics in the modeling of dynamic loop regions of proteins. One of these algorithms, cyclic coordinate descent (CCD), minimizes the distance of the two atoms of the “cut” bond in the loop by changing one torsional degree of freedom at a time, one after the other in a cycle until the loop is

closed.^[63] We refactored and adapted the CCD code within Rosetta to allow the rapid closing of saccharide “loops”, including cases where residues of the loop are branch points. CCD can be combined with backbone sampling and sugar-aware scoring to model such cases where glycan density was undefined in a starting crystal. We then implemented a carbohydrate loop-modeling protocol with Python using the PyRosetta libraries. The protocol performs 25 Monte Carlo cycles^[6,64] of small moves, CCD loop closure, and minimization. (25 cycles provided score convergence in this test case.) Our script is included within the supplementary material.

As an example, we modeled the non-fucosylated glycan β -D-Glc p NAc-(1 \rightarrow 2)- α -D-Man p -(1 \rightarrow 3)-[β -D-Gal p -(1 \rightarrow 4)- β -D-Glc p NAc-(1 \rightarrow 2)- α -D-Man p -(1 \rightarrow 6)]- β -D-Man p -(1 \rightarrow 4)- β -D-Glc p NAc-(1 \rightarrow 4)- β -D-Glc p NAc- found on a subunit of the Fc region of IgG from PDB structure 3AY4 (Figure 6).^[65] First, from the native structure, we cut the bond between the third residue of the glycan (β -D-Man p) and the α -D-Man p off its O6 branch. We then extended the two fragments (from the reducing-end β -D-Glc p NAc and the non-reducing-end β -D-Gal p) by setting all glycosidic torsion angles to 180° to form an open loop. With this starting structure, we used our PyRosetta protocol described above to close the loop and sample alternate loop conformations. After generating 500 model structures, 40% showed closed glycan loops with heavy-atom RMSDs less than 0.25 Å from the crystal structure (Figure 6, Supplemental Video).

Docking of Carbohydrates

Docking is an important modeling application that attempts to capture the interactions among multiple molecules within a system; it is crucial for understanding biological processes. Standard sampling moves in Rosetta docking algorithms involve rigid-body translations and rotations. If performing flexible docking, backbone motions are sampled with side-chain packing during refinement steps. Rosetta has had success in solving protein–protein,^[27,66] protein–ligand,^[28] and protein–peptide^[67,68] docking problems, and our early attempts at docking carbohydrate ligands placed us seventh among 31 teams during CAPRI challenge round 27.^[41,69] The docking algorithm we describe here for carbohydrate ligands expands and generalizes the method we used in CAPRI round 27, which was modeled after the FlexPepDock algorithm.^[67,68]

Before docking, we prepack native structures to remove side-chain clashes and unusual conformations. Each distinct docking trajectory is then started from a pose with randomly oriented ligand in both rigid-body and glycosidic-bond torsion space.

Each trajectory’s pose undergoes 100 Monte-Carlo cycles in which the ligand’s rigid-body orientation is perturbed, slid into contact with the interface, and minimized into a local rigid-body orientation, followed by random, small backbone movements, packing (including both glycan and peptide interface side-chains), and another minimization step, before accepting or rejecting each pose with the Metropolis criterion (Figure S4). 100 cycles were selected, because convergence of the score was observed within this number during early testing of the protocol.

The talaris2014 scoring function was used for scoring,^[44] with terms adjusted with multipliers found favorable for docking.^[70] We ramp the attractive and repulsive scoring term weights down and up, respectively, to the standard values throughout the course of the 100 cycles. A site constraint with a flat harmonic function,

$$f(x) = \begin{cases} 0 \text{ REU} & \text{for } 5 \text{ \AA} \leq x \leq 10 \text{ \AA} \\ \left(\frac{x-7.5 \text{ \AA}}{1 \text{ \AA}}\right)^2 \text{ REU} & \text{for } x < 5 \text{ \AA} \text{ or } x > 10 \text{ \AA} \end{cases}$$

is used to ensure that carbon C1 of the first residue of the glycan ligand is held close to the protein surface (any atom of any residue of both heavy and light antibody chains). Without this constraint, glycans tend to escape into the solvent.

The C++ application for glycan docking is included with the current release version of Rosetta.

Docking Benchmark—To compare results to previous work, we tested our protocol using a set of antibody–glycoantigen pairs from Agostino *et al.*, who compared the capabilities of four docking algorithms.^[71] In that report, the eleven test cases were chosen for the small size of the glycan ligands (four or fewer monosaccharide residues), and all ligands were antigens in an antibody–carbohydrate complex with a solved structure in the PDB. For our test, we modeled nine of the eleven cases, (omitting the single glycolipid case—which was noted as problematic for all four algorithms tested in the Agostino report—and a case containing a deoxyrhamnose, for which we did not have parameters at the time of our test.) The nine glycoantigens docked included two Lewis series antigens, two high-mannose fragments, and three derivatives of α -D-Galp-(1→2)-[α -D-Abep-(1→3)]- α -D-Manp (Figure 7). Two of the glycoantigens were docked in two separate test cases each to different antibodies. The diverse set thus contained deoxy sugars, *N*-acetylhexosamines, methyl glycosides, and branched and linear molecules.

As in the Agostino study, we performed local bound–unbound docking, where we started the simulation with the antibody coordinates from the PDB and the carbohydrate ligand coordinates from a randomized backbone conformation and a randomized rigid-body orientation in the vicinity of the binding site. Following Agostino *et al.*, we kept ring conformations locked in their starting orientations.

We generated 5,000 decoys for each protein–ligand pair. We compared all decoys resulting at the end of the protocol to the prepacked native structures and calculated root-mean-squared deviations (RMSDs).

Agostino *et al.* also performed “flexible receptor docking” in which the bound antibody structure was allowed to move during the simulation. We also performed such flexible receptor docking but found no qualitative difference from our “rigid receptor docking” results. We only describe the rigid-receptor results here.

Docking Results—We consider three criteria indicative of a successful docking calculation: A) that we have sampled structures close to the native complex, B) that our lowest-scoring decoys are closest to the native complex, and C) that the distribution of our decoys “funnels” toward only a single structure. We consider Rosetta to have been successful if all of the above criteria are true and partially successful if only two of the above criteria are true. For our benchmarking sample, Rosetta was fully successful in four of nine cases and partially successful in three. A summary of the results are shown in Table 1.

The four cases involving Gal-[Abe]-Man antigens showed similar results. In three of those four cases (1MFC was the exception), we observe a docking funnel in which the best ranked decoys by total Rosetta score are among the decoys with the lowest ligand root-mean-square deviation (RMSD). In the case of 1MFA, Rosetta’s lowest-scoring decoy was the best decoy by ligand RMSD (Table 1), but the docking funnel was not deep for total score *vs.* ligand RMSD in any of the cases (Figure S5). However, when we plot the interface score ($E_{\text{interface}} = E_{\text{total}} - E_{\text{split}}$, where E_{split} is the Rosetta score when the ligand is pulled away from the antibody) *vs.* ligand RMSD, we see steep funnels in all cases (Figure 8a, Figure S5). Four out of five lowest-interface-score structures for 1MFA recover better than 85% of native contacts, three out of five structures for 1MFC recover better than 75%, five of five for 1MFD recover better than 90%, and ten out of ten recover better than 80% for 1MFE. (We consider a receptor and ligand residue to be in contact if any heavy atom of one residue is within 5 Å of a heavy atom of the other residue.) These cases are complete successes.

For benchmark cases 1OP3, 1S3K, and 1ZLU (Figure S6), we see funnels both for total score *vs.* ligand RMSD and interface score *vs.* ligand RMSD, but they are not steep. On the contrary, some structures well beyond 10 Å RMSD drop lower in score than near-native decoys. We judge this a problem of scoring/discrimination. In the case of 1OP3, we sample very close to the native complex, with many decoys having 100% of native contacts received. Our second-best decoy has a ligand RMSD of 0.74 Å and recovers 100% of native contacts, yet our best decoy has an RMSD of 15.48 Å and recovers no contacts (Figure 8b,e). In contrast with our four fully successful cases, with 1OP3, Rosetta is better at predicting a correct structure using total score than it is by relying on interface score.

Case 1S3K shows, instead, a sampling failure. None of the 5,000 decoys generated have a ligand RMSD less than 2.5 Å. Even so, one structure with an RMSD of 2.65 Å recovers 88% of native contacts, but that is the only structure among the ten best decoys by interface score to recover more than 75%. While we do observe a steep funnel in the interface score *vs.* ligand RMSD plot (Figure S6), Rosetta did not sample well enough to consistently generate good models.

The same glycoantigen as in case 1OP3 was docked in complex 1ZLU, and the results were similar. Though the best model by total score was also the best model by ligand RMSD and half of the ten lowest-scoring decoys by interface score recovered 95%, we consider this only a partial success because the funnels in neither plot (Figure S6) were steep, leading to many false positives.

Thus, we consider our tests with the above three cases, 1OP3, 1S3K, and 1ZLU, to be partial successes. Taking the above seven success cases together, if we were to blindly take the top ten best interface-scoring decoys, we would have at least one extremely close model in all cases.

As an example of a failure case, as shown in Figure 8c and f, the decoy with the best interface does not superimpose with the native ligand at all. Even in the decoy with the lowest RMSD, the ring of the galactose residue of the Lewis^X ligand is flipped 180° relative to the native structure. Both sets of failure case docking funnels are shown in Figure S7. Failure cases indicate areas of future improvement to be made in Rosetta's sampling and scoring methods.

Summary & Outlook

We have created a new framework within the Rosetta modeling and design suite to permit structural modeling of saccharides and glycoconjugates. In doing so, we have also expanded Rosetta's handling of polymer branching and ring conformations in general. We have introduced functionality for virtual glycosylations of structures in Rosetta and demonstrated applications of glycan loop modeling and ligand docking. Our initial glyco-ligand benchmarking tests show great promise in recovering native-like docked structures for oligosaccharide ligands: we can recover multiple structures with high levels of native-like contacts in six of nine test cases and at least one native-like structure in an additional case.

Our RosettaCarbohydrate framework opens the door for many applications, two of which are particularly noteworthy. First is the crystal refinement of glycoproteins. Many large protein structures are covered in linked glycans, such as the HIV envelope protein. Historically, Rosetta has had great success at helping to refine the x-ray structures of large peptide complexes with initially poor resolutions.^[24] Unfortunately, since Rosetta could not properly account for carbohydrate residues, poor-resolution structures with glycan density could not be resolved using the same methods. Our modifications to Rosetta will now permit such untenable structures to be resolved.

Second, our framework allows for design applications. Through the manipulation of biosynthetic pathways and the provision of non-native or synthetic monosaccharides and precursors, glycoengineers can create unique glycosylation patterns.^[19] The natural diversity of glycans is already vast, but it multiplies when synthetic saccharides are added to the mix—and with them endless applications for research or medicinal purposes. Rosetta's design strategy capitalizes on its "residue-based" organization. During the packing stage of a design algorithm, when swapping out one residue's current rotamer for another, alternative residue types are sampled as well. It is a conceptually simple matter to extend Rosetta design to include non-natural monosaccharide residues. The ability to design alternative glycosylations in advance would save cost and speed the advance of glycoengineering research.

Code Availability

RosettaCarbohydrate is part of the Rosetta modeling suite (www.rosettacommons.org), which is freely available for academic and non-profit use. The supplementary material includes Python scripts and command-line syntax for using applications described in this paper. Component methods and objects are also available in the PyRosetta libraries (www.pyrosetta.org).^[72]

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors thank Dr. Brian Weitzner, Michael Pacella, Dr. Krishna Praneeth Kilambi, Nick Marze, Dr. Julia Koehler Leman, Sergey Lyskov, Matt Mulqueen, Morgan Nance, Sofia Bali, and Thuy-My Lee for helpful discussion and assistance. We are grateful to the members of the RosettaCommons, particularly Drs. Philip Bradley, Samuel DeLuca, Frank DiMaio, Kevin Drew, Andrew Leaver-Fay, Steven Lewis, Rocco Moretti, Matthew O'Meara, and Andrew Watkins. We also thank the members of the Rosetta Chemical XRW. Finally, we thank the Baltimore–Washington Glycobiology Interest Group, most notably Drs. Kevin Yarema, Michael Betenbaugh, Ronald Schnaar, Natasha Zachara, and Gerald Hart.

This work was supported by the National Institutes of Health grants 1F32-CA189246-01 (J.W.L.) and R01 GM-078221 (J.J.G.); the International AIDS Vaccine Initiative Neutralizing Antibody Consortium and Center (W.R.S.); CAVD funding for the IAVI NAC Center (W.R.S.); the Ragon Institute of MGH, MIT, and Harvard (W.R.S.); the National Institute of Allergy and Infectious Diseases grant CHAVI-ID 1UM1AI100663 (W.R.S.) and postdoctoral training grant T32AI007244 (J.A.-B.).

References

1. Committee on Assessing the Importance and Impact of Glycomics and Glycosciences, Board on Chemical Sciences and Technology, Board on Life Sciences, Division on Earth and Life Studies, National Research Council. Transforming Glycoscience: A Roadmap for the Future. National Academies Press; Washington, DC: 2012.
2. Varki, A.; Cummings, RD.; Esko, JD.; Freeze, HH.; Stanley, P.; Bertozzi, CR.; Hart, GW.; Etzler, ME. Essentials of Glycobiology. Cold Spring Harbor Laboratory Press; Cold Spring Harbor, NY: 2008.
3. Werz DB, Ranzinger R, Herget S, Adibekian A, von der Lieth CW, Seeberger PH. ACS Chem Biol. 2007; 2:685–691. [PubMed: 18041818]
4. Krambeck FJ, Betenbaugh MJ. Biotechnol Bioeng. 2005; 92:711–728. [PubMed: 16247773]
5. Fadda E, Woods RJ. Drug Discov Today. 2010; 15:596–609. [PubMed: 20594934]
6. Dowd MK, Kiely DE, Zhang J. Carbohydr Res. 2011; 346:1140–1148. [PubMed: 21536262]
7. Csonka GI, Elias K, Csizmadia IG. Chem Phys Lett. 1996; 257:49–60.
8. Kirschner KN, Yongye AB, Tschampel SM, Gonzalez-Outeirino J, Daniels CR, Foley BL, Woods RJ. J Comput Chem. 2008; 29:622–655. [PubMed: 17849372]
9. Mallajosyula SS, Guvench O, Hatcher E, Mackerell AD Jr. J Chem Theory Comput. 2012; 8:759–776. [PubMed: 22685386]
10. Kony D, Damm W, Stoll S, Van Gunsteren WF. J Comput Chem. 2002; 23:1416–1429. [PubMed: 12370944]
11. Lins RD, Hunenberger PH. J Comput Chem. 2005; 26:1400–1412. [PubMed: 16035088]
12. Allinger NL, Chen KH, Lii JH, Durkin KA. J Comput Chem. 2003; 24:1447–1472. [PubMed: 12868110]
13. Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, Olson AJ. J Comput Chem. 2009; 30:2785–2791. [PubMed: 19399780]

14. Trott O, Olson AJ. *J Comput Chem.* 2010; 31:455–461. [PubMed: 19499576]
15. Lang PT, Brozell SR, Mukherjee S, Pettersen EF, Meng EC, Thomas V, Rizzo RC, Case DA, James TL, Kuntz ID. *RNA.* 2009; 15:1219–1230. [PubMed: 19369428]
16. Claussen H, Gastreich M, Apelt V, Greene J, Hindle SA, Lemmen C. *Curr Drug Discov Technol.* 2004; 1:49–60. [PubMed: 16472219]
17. Friesner RA, Murphy RB, Repasky MP, Frye LL, Greenwood JR, Halgren TA, Sanschagrin PC, Mainz DT. *J Med Chem.* 2006; 49:6177–6196. [PubMed: 17034125]
18. Hartshorn MJ, Verdonk ML, Chessari G, Brewerton SC, Mooij WT, Mortenson PN, Murray CW. *J Med Chem.* 2007; 50:726–741. [PubMed: 17300160]
19. Du J, Meledeo MA, Wang Z, Khanna HS, Paruchuri VD, Yarema KJ. *Glycobiology.* 2009; 19:1382–1401. [PubMed: 19675091]
20. 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. *Methods Enzymol.* 2011; 487:545–574. [PubMed: 21187238]
21. Raman S, Vernon R, Thompson J, Tyka M, Sadreyev R, Pei J, Kim D, Kellogg E, DiMaio F, Lange O, Kinch L, Sheffler W, Kim BH, Das R, Grishin NV, Baker D. *Proteins.* 2009; 77(Suppl 9):89–99. [PubMed: 19701941]
22. Das R, Baker D. *Proc Natl Acad Sci USA.* 2007; 104:14664–14669. [PubMed: 17726102]
23. Ramelot TA, Raman S, Kuzin AP, Xiao R, Ma LC, Acton TB, Hunt JF, Montelione GT, Baker D, Kennedy MA. *Proteins.* 2009; 75:147–167. [PubMed: 18816799]
24. DiMaio F, Echols N, Headd JJ, Terwilliger TC, Adams PD, Baker D. *Nat Methods.* 2013; 10:1102–1104. [PubMed: 24076763]
25. DiMaio F, Song Y, Li X, Brunner MJ, Xu C, Conticello V, Egelman E, Marlovits TC, Cheng Y, Baker D. *Nat Methods.* 2015; 12:361–365. [PubMed: 25707030]
26. Mandell DJ, Coutsias EA, Kortemme T. *Nat Methods.* 2009; 6:551–552. [PubMed: 19644455]
27. Wang C, Bradley P, Baker D. *J Mol Biol.* 2007; 373:503–519. [PubMed: 17825317]
28. Davis IW, Baker D. *J Mol Biol.* 2009; 385:381–392. [PubMed: 19041878]
29. Kuhlman B, Baker D. *Proc Natl Acad Sci USA.* 2000; 97:10383–10388. [PubMed: 10984534]
30. Kortemme T, Joachimiak LA, Bullock AN, Schuler AD, Stoddard BL, Baker D. *Nat Struct Mol Biol.* 2004; 11:371–379. [PubMed: 15034550]
31. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, Baker D. *Science.* 2003; 302:1364–1368. [PubMed: 14631033]
32. Koga N, Tatsumi-Koga R, Liu GH, Xiao R, Acton TB, Montelione GT, Baker D. *Nature.* 2012; 491:222–227. [PubMed: 23135467]
33. Correia BE, Bates JT, Loomis RJ, Baneyx G, Carrico C, Jardine JG, Rupert P, Correnti C, Kalyuzhnyi O, Vittal V, Connell MJ, Stevens E, Schroeter A, Chen M, MacPherson S, Serra AM, Adachi Y, Holmes MA, Li YX, Kleivit RE, Graham BS, Wyatt RT, Baker D, Strong RK, Crowe JE, Johnson PR, Schief WR. *Nature.* 2014; 507:201–206. [PubMed: 24499818]
34. Jiang L, Althoff EA, Clemente FR, Doyle L, Rothlisberger D, Zanghellini A, Gallaher JL, Betker JL, Tanaka F, Barbas CF 3rd, Hilvert D, Houk KN, Stoddard BL, Baker D. *Science.* 2008; 319:1387–1391. [PubMed: 18323453]
35. Rothlisberger D, Khersonsky O, Wollacott AM, Jiang L, DeChancie J, Betker J, Gallaher JL, Althoff EA, Zanghellini A, Dym O, Albeck S, Houk KN, Tawfik DS, Baker D. *Nature.* 2008; 453:190–195. [PubMed: 18354394]
36. Fleishman SJ, Whitehead TA, Ekiert DC, Dreyfus C, Corn JE, Strauch EM, Wilson IA, Baker D. *Science.* 2011; 332:816–821. [PubMed: 21566186]
37. Das R, Karanicolas J, Baker D. *Nat Methods.* 2010; 7:291–294. [PubMed: 20190761]
38. Masica DL, Schrier SB, Specht EA, Gray JJ. *J Am Chem Soc.* 2010; 132:12252–12262. [PubMed: 20712308]
39. Schrier SB, Sayeg MK, Gray JJ. *Langmuir.* 2011; 27:11520–11527. [PubMed: 21797243]

40. Drew K, Renfrew PD, Craven TW, Butterfoss GL, Chou F, Lyskov S, Bullock BN, Watkins A, Labonte JW, Pacella M, Kilambi KP, Leaver-Fay A, Kuhlman B, Gray JJ, Bradley P, Kirshenbaum K, Arora PS, Das R, Bonneau R. *PLoS ONE*. 2013; 8
41. Kilambi KP, Pacella MS, Xu J, Labonte JW, Porter JR, Muthu P, Drew K, Kuroda D, Schueler-Furman O, Bonneau R, Gray JJ. *Proteins*. 2013; 81:2201–2209. [PubMed: 24123494]
42. Perez S, Tvaroska I. *Adv Carbohydr Chem Biochem*. 2014; 71:9–136. [PubMed: 25480504]
43. Ho BK, Coutsiyas EA, Seok C, Dill KA. *Protein science : a publication of the Protein Society*. 2005; 14:1011–1018. [PubMed: 15772308]
44. O'Meara MJ, Leaver-Fay A, Tyka MD, Stein A, Houlihan K, DiMaio F, Bradley P, Kortemme T, Baker D, Snoeyink J, Kuhlman B. *J Chem Theory Comput*. 2015; 11:609–622. [PubMed: 25866491]
45. 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. *Methods Enzymol*. 2013; 523:109–143. [PubMed: 23422428]
46. Shapovalov MV, Dunbrack RL Jr. *Structure*. 2011; 19:844–858. [PubMed: 21645855]
47. Ting D, Wang G, Shapovalov M, Mitra R, Jordan MI, Dunbrack RL Jr. *PLoS Comput Biol*. 2010; 6:e1000763. [PubMed: 20442867]
48. Zierke M, Smieško M, Rabbani S, Aeschbacher T, Cutting B, Allain FHT, Schubert M, Ernst B. *J Am Chem Soc*. 2013; 135:13464–13472. [PubMed: 24001318]
49. McNaught AD. *Adv Carbohydr Chem Biochem*. 1997; 52:43–177. [PubMed: 9218333]
50. DeMarco ML, Woods RJ. *Glycobiology*. 2008; 18:426–440. [PubMed: 18390826]
51. Lee J, Cheng X, Swails JM, Yeom MS, Eastman PK, Lemkul JA, Wei S, Buckner J, Jeong JC, Qi Y, Jo S, Pande VS, Case DA, Brooks CL 3rd, MacKerell AD Jr, Klauda JB, Im W. *J Chem Theory Comput*. 2016; 12:405–413. [PubMed: 26631602]
52. Jo S, Song KC, Desaire H, MacKerell AD Jr, Im W. *J Comput Chem*. 2011; 32:3135–3141. [PubMed: 21815173]
53. Ceroni A, Maass K, Geyer H, Geyer R, Dell A, Haslam SM. *J Proteome Res*. 2008; 7:1650–1659. [PubMed: 18311910]
54. Varki A, Cummings RD, Aebi M, Packer NH, Seeberger PH, Esko JD, Stanley P, Hart G, Darvill A, Kinoshita T, Prestegard JJ, Schnaar RL, Freeze HH, Marth JD, Bertozzi CR, Etzler ME, Frank M, Vliegenthart JF, Lutteke T, Perez S, Bolton E, Rudd P, Paulson J, Kanehisa M, Toukach P, Aoki-Kinoshita KF, Dell A, Narimatsu H, York W, Taniguchi N, Kornfeld S. *Glycobiology*. 2015; 25:1323–1324. [PubMed: 26543186]
55. Pancera M, Majeed S, Ban YEA, Chen L, Huang CC, Kong L, Kwon YD, Stuckey J, Zhou TQ, Robinson JE, Schief WR, Sodroski J, Wyatt R, Kwong PD. *Proc Natl Acad Sci USA*. 2010; 107:1166–1171. [PubMed: 20080564]
56. Petrescu AJ, Petrescu SM, Dwek RA, Wormald MR. *Glycobiology*. 1999; 9:343–352. [PubMed: 10089208]
57. Petrescu AJ, Milac AL, Petrescu SM, Dwek RA, Wormald MR. *Glycobiology*. 2004; 14:103–114. [PubMed: 14514716]
58. Nivedha AK, Makeneni S, Foley BL, Tessier MB, Woods RJ. *J Comput Chem*. 2014; 35:526–539. [PubMed: 24375430]
59. Nivedha AK, Thieker DF, Makeneni S, Hu H, Woods RJ. *J Chem Theory Comput*. 2016; 12:892–901. [PubMed: 26744922]
60. Cremer D, Pople JA. *J Am Chem Soc*. 1975; 97:1354–1358.
61. Bock K, Duus JO, Refn S. *Carbohydr Res*. 1994; 253:51–67. [PubMed: 8156558]
62. Fleming CD, Bencharit S, Edwards CC, Hyatt JL, Tsurkan L, Bai F, Fraga C, Morton CL, Howard-Williams EL, Potter PM, Redinbo MR. *J Mol Biol*. 2005; 352:165–177. [PubMed: 16081098]
63. Canutescu AA, Dunbrack RL Jr. *Protein Sci*. 2003; 12:963–972. [PubMed: 12717019]
64. Li Z, Scheraga HA. *Proc Natl Acad Sci USA*. 1987; 84:6611–6615. [PubMed: 3477791]
65. Mizushima T, Yagi H, Takemoto E, Shibata-Koyama M, Isoda Y, Iida S, Masuda K, Satoh M, Kato K. *Genes Cells*. 2011; 16:1071–1080. [PubMed: 22023369]

66. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D. *J Mol Biol.* 2003; 331:281–299. [PubMed: 12875852]
67. Raveh B, London N, Zimmerman L, Schueler-Furman O. *PLoS ONE.* 2011; 6:e18934. [PubMed: 21572516]
68. Raveh B, London N, Schueler-Furman O. *Proteins.* 2010; 78:2029–2040. [PubMed: 20455260]
69. Janin J. *Proteins.* 2013; 81:2075–2081. [PubMed: 23900782]
70. Chaudhury S, Gray JJ. *J Mol Biol.* 2008; 381:1068–1087. [PubMed: 18640688]
71. Agostino M, Jene C, Boyle T, Ramsland PA, Yuriev E. *J Chem Inf Model.* 2009; 49:2749–2760. [PubMed: 19994843]
72. Chaudhury S, Lyskov S, Gray JJ. *Bioinformatics.* 2010; 26:689–691. [PubMed: 20061306]

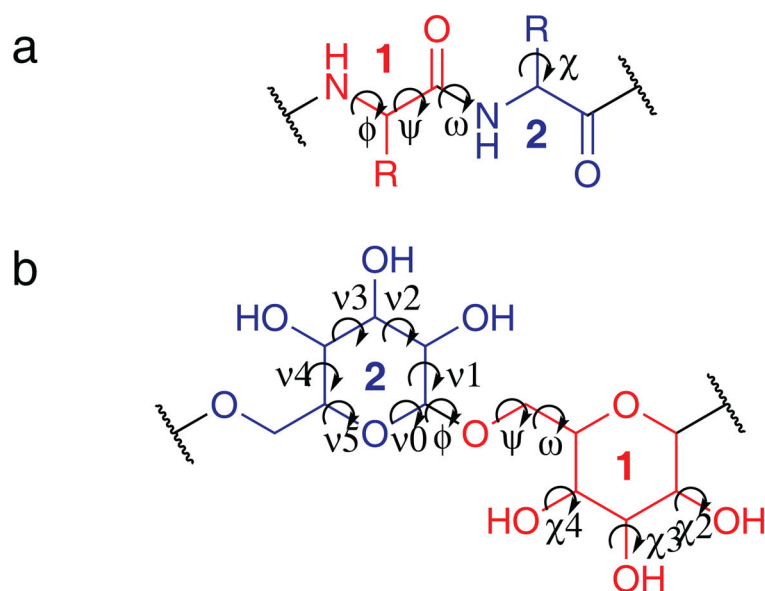


Figure 1.
A comparison of the degrees of freedom (DoFs) found in polypeptide (a) and polysaccharide (b) chains. The first and second residue are labeled and colored red and blue respectively. Torsion angles are indicated by arrows and labeled.

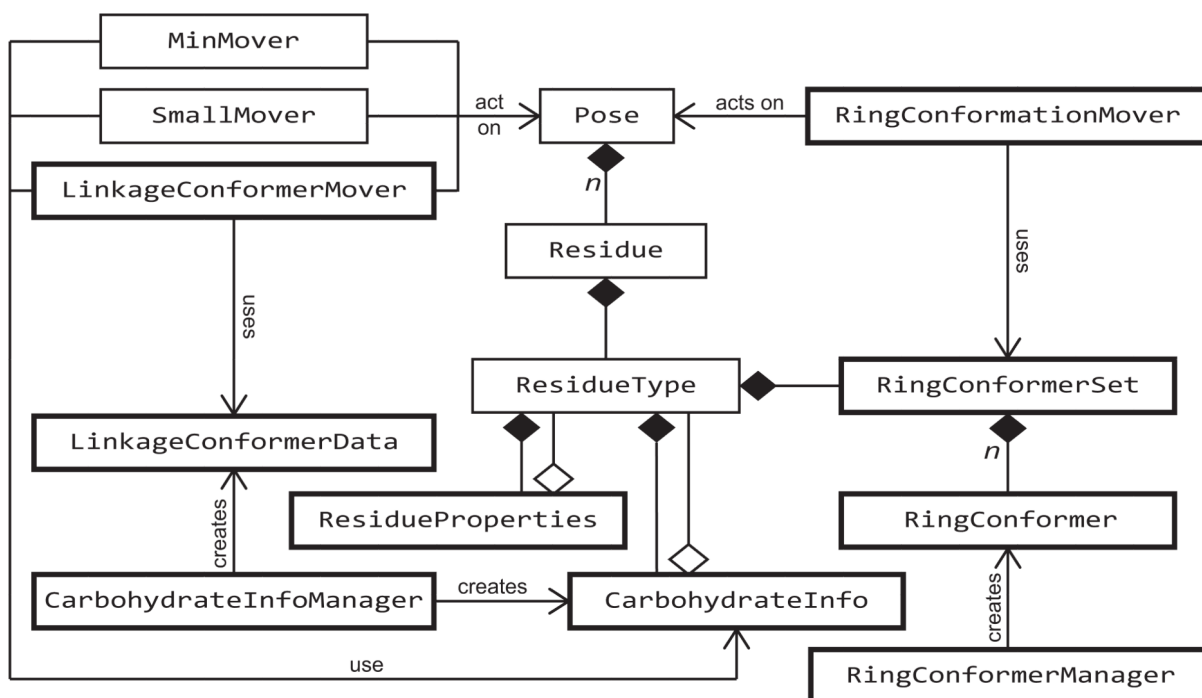


Figure 2.

A unified modeling language (UML) diagram of the RosettaCarbohydrate framework. New data objects introduced in this paper are shown with bold rectangles. In UML, solid diamonds (◆) indicate ownership of data, (*e.g.*, Pose owns n Residues); open diamonds (◇) indicate access of data; and arrows describe functional relationships.

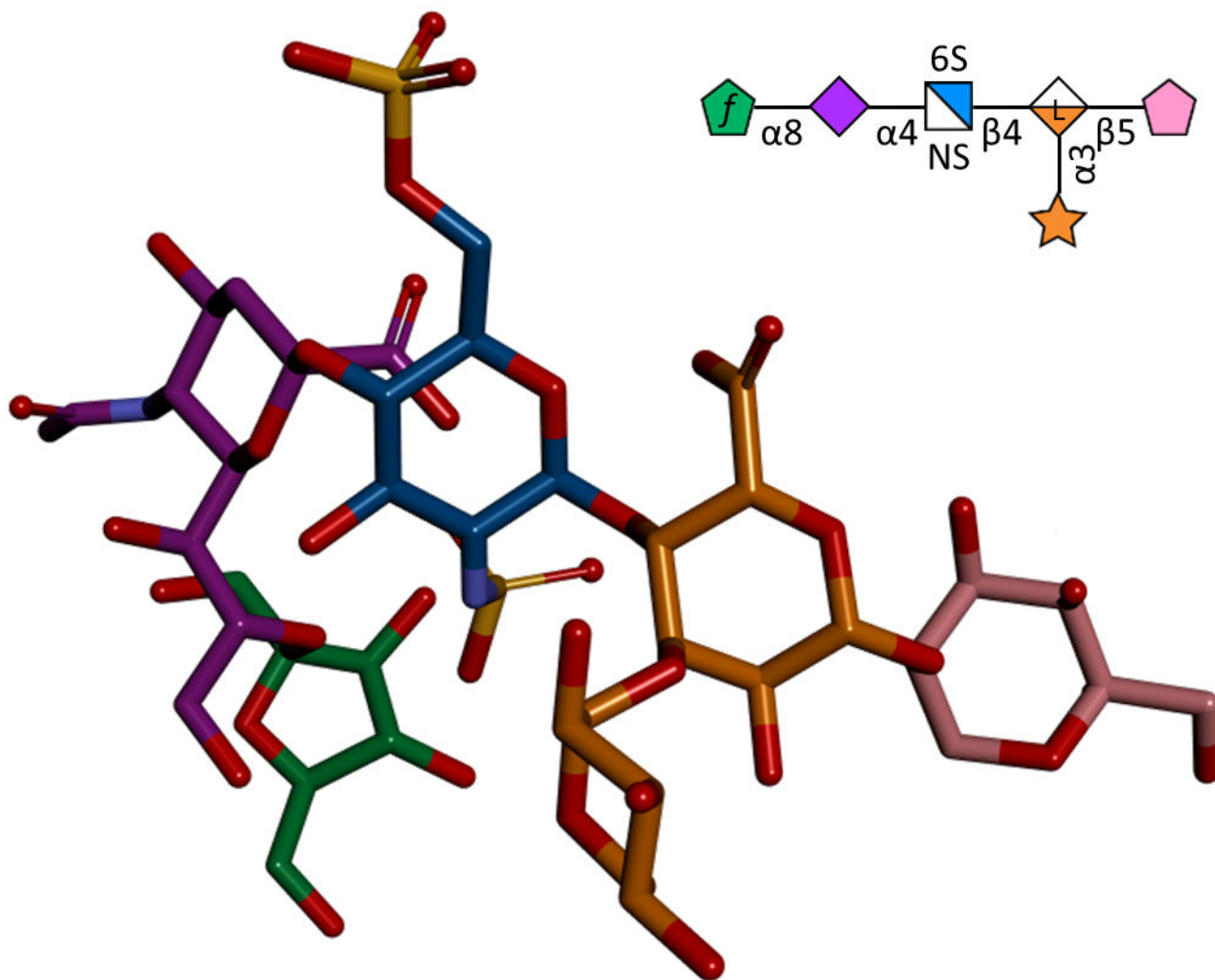


Figure 3.

A branched, completely unnatural sugar generated with the PyRosetta command `pose_from_saccharide_sequence(whacky_sugar, 'b-D-Fruf-(2->8)-a-Neup5Ac-(2->4)-b-D-GlcpNS6S-(1->4)-[a-D-Xylp-(1->3)]-b-L-GulpA-(1->5)-b-D-Psip'`), to demonstrate Rosetta's ability to generate and handle ketoses, uronic acids, L-sugars, sulfated sugars, sialic acids, furanoses, and pentoses. The structure is not refined.

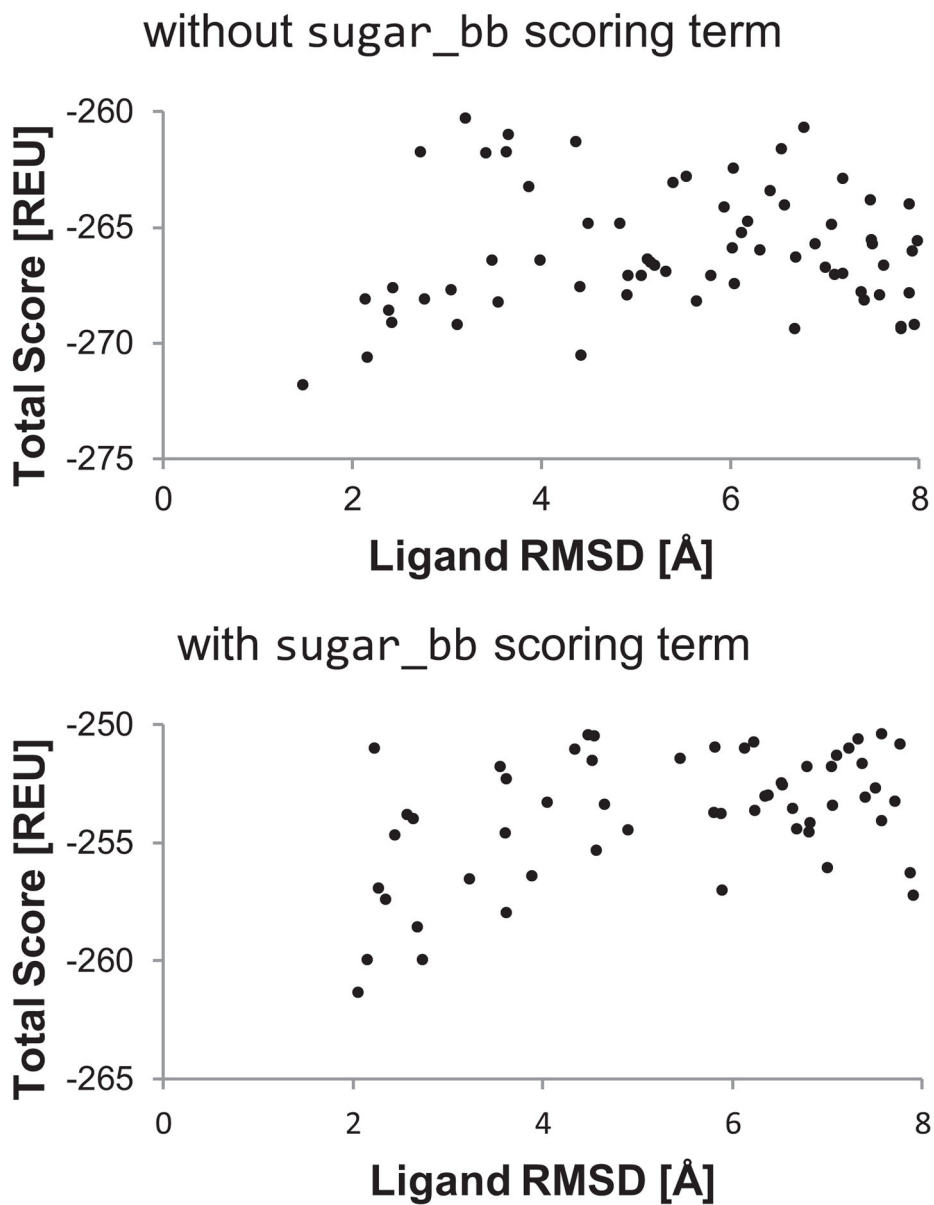


Figure 4. Funnel plots comparing flexible bound–unbound glyco-ligand docking of maltose-binding protein (MBP) with maltotetraose (G4). The docking funnel is steeper when the sugar_{bb} scoring term is used. 50 decoys were generated per plot.

```

from rosetta import init, pose_from_sequence, \
    MinMover, MoveMap
from rosetta.core.pose.carbohydrates import \
    glycosylate_pose_by_file

init('-include_sugars -write_pdb_link_records')

peptide = pose_from_sequence('AANAS')
peptide.set_phi(1, 105)
peptide.set_psi(1, 130)
...

glycosylate_pose_by_file(peptide, 3, 'man3')

mm = MoveMap()
mm.set_chi(True)
minimizer = MinMover()
minimizer.movemap(mm)
minimizer.apply(peptide)

peptide.dump_pdb('N-link_example.pdb')

```

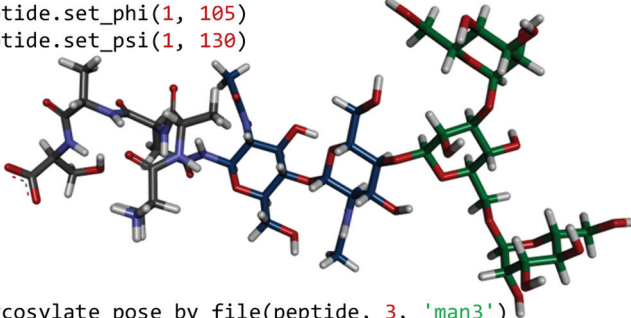
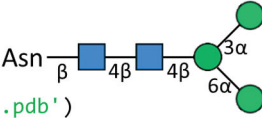



Figure 5.

A PyRosetta script, demonstrating the `glycosylate_pose_by_file()` function, and its output structure (overlay, CFG colors). A small peptide is first created from sequence, and its main-chain torsion angles are set to typical β -turn values (some lines omitted). After glycosylating, the structure is minimized to remove clashes. The `man3.iupac` file contains the sequence: a-D-Manp-(1->3)-[a-D-Manp-(1->6)]-b-D-Manp-(1->4)-b-D-GlcpNAc-(1->4)-b-D-GlcpNAc-.

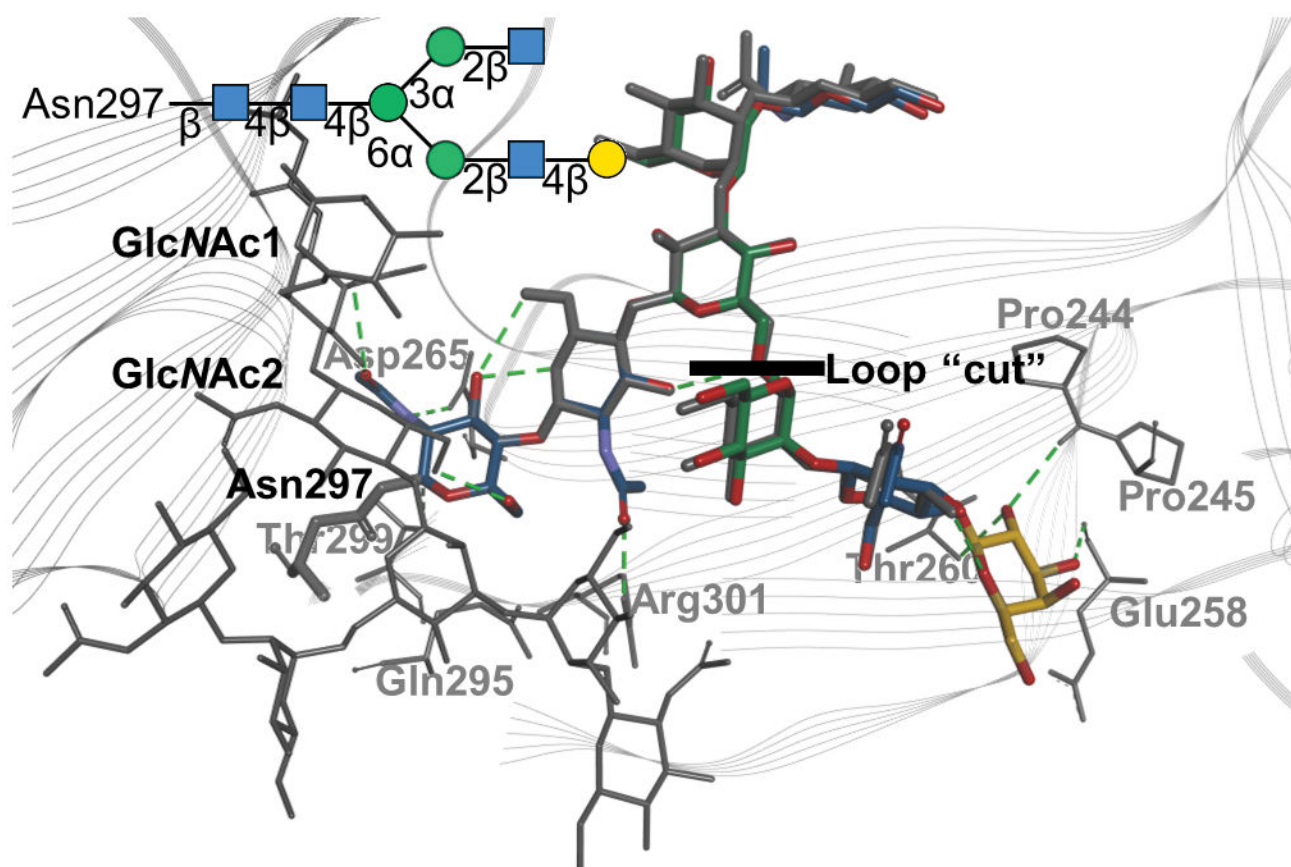


Figure 6.

A superimposition of the lowest-scoring Rosetta loop model (CFG colors) and the native structure from IgG Fc-Fc γ RIII (gray, PDB ID 3AY4). Saccharide residues are shown with thicker lines. Hydrogen bonds are shown in green dashed lines. The initial GlcNAc and terminal Gal residues were fixed as the start and end of a “loop” remodeled using the cyclic coordinate descent (CCD) algorithm with a “cut” in the middle of the loop.

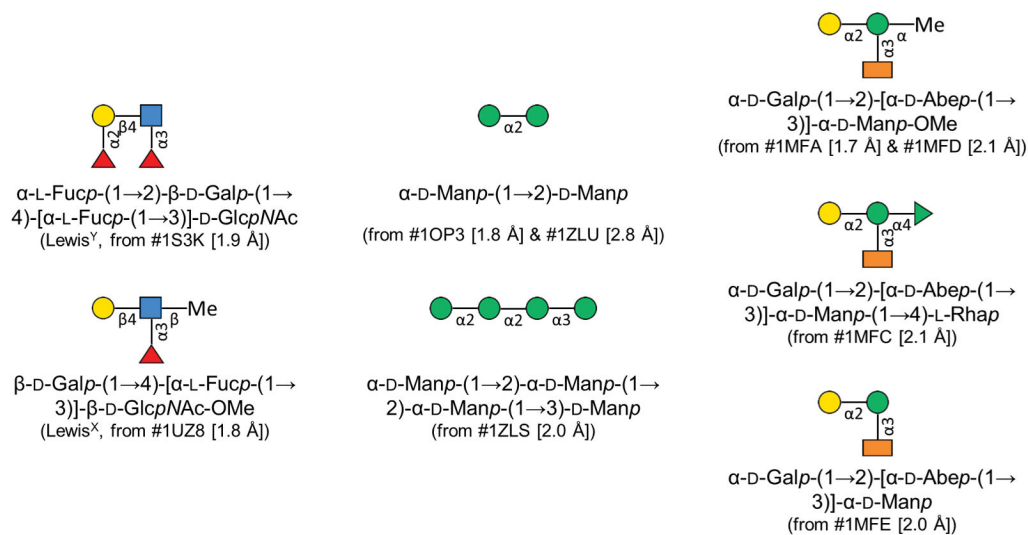


Figure 7.
 The seven glyco-ligands selected for the glycan docking benchmark. Symbols follow standard CFG notation.

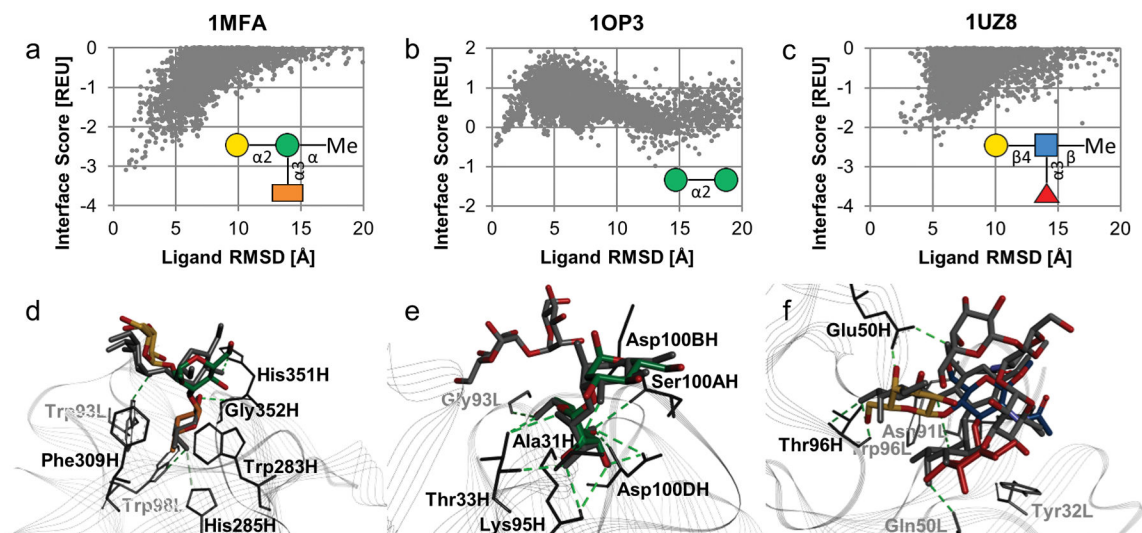


Figure 8.

A comparison of ligand docking results. (a)–(c): Plots of interface score in Rosetta energy units vs. ligand RMSD for 5,000 structures created from independent starting configurations. (d)–(f): Rosetta decoys superimposed with the relaxed native structures. Hydrogen bonds are shown with green dashed lines. (d): The decoy with best interface score in CFG colors. (e): The decoy with the second-best interface score in CFG colors and the one with best interface score colored by element. (f): The decoy with the lowest RMSD in CFG colors and the one with best interface score colored by element.

Table 1

Flexible bound–unbound antibody–glyco-ligand docking results.

PDB ID	RMSD of best decoy (Å)		Ranking of Decoy Closest to Native			$N_{\text{top-10}}$ decoys with $F_{\text{mat}} > 75\%$	
	overall	total score	interface score	total score	interface score	total score	interface score
IMEA	1.02	1.02	1.02	1 st	1 st	4	10
IMFC	2.31	7.43	2.71	>20 th	17 th	0	4
IMFD	0.59	1.02	1.44	3 rd	6 th	6	10
IMFE	0.54	1.15	0.54	4 th	1 st	8	10
<i>Complete Successes</i>							
IOP3	0.33	15.48	13.06	10 th	19 th	4	0
IS3K	2.65	10.26	4.86	>20 th	10 th	0	1
IZLU	0.27	0.76	0.55	>20 th	>20 th	9	5
<i>Failures</i>							
IUZ8	2.46	7.13	5.70	>20 th	>20 th	0	0
IZLS	1.61	5.59	9.32	>20 th	>20 th	0	0