

CarbBuilder: An Adjustable Tool for Building 3D Molecular Structures of Carbohydrates for Molecular Simulation.

Michelle Kuttel
and Yue Mao

Department of Computer Science
University of Cape Town
Cape Town, South Africa 7701
Email: mkuttel@cs.uct.ac.za

Göran Widmalm
and Magnus Lundborg

Department of Organic Chemistry
Arrhenius Laboratory
Stockholm University
S-106 91 Stockholm
Sweden
Email: gw@organ.su.se

Abstract—We describe CarbBuilder, a software tool for building 3D structures of carbohydrates: the most structurally varied of all molecular classes. CarbBuilder was designed with the dual aims of portability and adaptability, using an iterative software development approach. CarbBuilder employs a simple, heuristic algorithm based upon experimental data to convert a primary structure description of a carbohydrate molecule into a three-dimensional structure file. This straightforward approach means that CarbBuilder can be easily adapted: users can add additional monosaccharide building blocks or alter the conformational defaults to suit specific requirements. The output carbohydrate structure can be used for subsequent molecular simulation investigations. CarbBuilder is freely available and portable: it is a text-based stand-alone program that can run on Windows, Linux and MacOS X systems without installation.

I. INTRODUCTION

Molecular modelling methods are increasingly employed to provide insight into important questions of chemical structure and interaction, both of which can play crucial roles in disease. One of the first obstacles to a successful molecular mechanics simulation is establishing and creating an initial structure file from which to begin a simulation. In the absence of an experimentally-determined structure for a molecule, it is often very difficult and/or time consuming to build a reasonable structure from which to start simulations. This is particularly true for carbohydrate simulations, because very few structures of these flexible molecules have been solved experimentally. Even where a structure is available, this must typically be edited so that atom names and residue types conform to those listed in the chosen empirical force field; often a tedious task.

There are currently very few products available for accessing or building three-dimensional structures of carbohydrates where experimental data does not exist. Two of the main packages, SweetDB and Glycam-web, are exclusively web-based and thus require a reliable internet connection and are difficult to integrate into other software.

The SweetDB [1] project, initiated by the glycosciences.de

group at the German Cancer Research Center (DKFZ), provides access to databases and bioinformatics tools to support glycobiology and glycomics research. The web service provides access to a database of existing experimentally-determined carbohydrate structures, as well as the facility to build structures using the SWEET software. SWEET and its successor SWEET-II were the first web-based molecular builders for three-dimensional structures of carbohydrates. The server migrated to the University of Giessen in March 2010, but SweetDB is no longer under active development.

Glycam Biomolecule Builder [2] is a free, online builder for carbohydrates and related molecules (e.g., glycoproteins) developed at the University of Georgia. It allows users to build linear or branched oligosaccharides. Automated builds can be initiated via URL and Glycam also provides files for AMBER or CHARMM simulations. However, Glycam supports input of only the most common monosaccharides. In addition, building of larger branched oligo- or polysaccharides is awkward with these web-based tools.

There have been a number of other software products for building carbohydrates structures reported in the literature, but most are not freely available, even for purchase, or else poorly supported. This is a very common problem with scientific software in general: useful algorithms are developed for use in-house by laboratories, but seldom made publicly available or incorporated into existing tools [3]. Examples of these are the POLYS [4] and Shape [5] algorithms. POLYS is reported as producing three-dimensional structures of polysaccharides and complex carbohydrates, employing for this purpose a database of monosaccharide structures and information on populations of independent neighboring glycosidic linkages in disaccharide fragments. The Shape software package uses a genetic algorithm conformation search for automated modelling. Another point illustrated by these two packages is that is very difficult for users to search for software named with commonly used words!

We felt that there was a clear need for a portable, standalone

tool for building carbohydrate structures. Together, we have a history in development of software to support carbohydrate research: CASPER – a web based program to determine the primary structure of oligo- and polysaccharides using NMR data [6], [7] – and the *Twister* and *PaperChain* carbohydrate visualization algorithms to support analysis of complex carbohydrate structures [8], [9]. We are currently collaborating on the development of a comprehensive software package that will provide the scientific community with tools to enable the construction and analysis of 3D carbohydrate structures to support and facilitate computation investigations into carbohydrate structure and dynamics. Here we describe development of the first component of this software package, CarbBuilder. Specifically, the role of the CarbBuilder module is to convert the primary structure description of carbohydrate into a “reasonable” 3D structure: a 3D structure sufficiently accurate to be used for subsequent successful molecular modelling investigations.

II. BACKGROUND

A. Carbohydrates

Carbohydrates, or saccharides, are organic compounds comprising carbon, hydrogen and oxygen atoms. Monosaccharides, or simple sugars, are the most basic form of carbohydrate and are typically in a ring form. More complex oligosaccharides and polysaccharides are built from these simple sugar residues linked together by a glycosidic bond. The constituent rings (monosaccharides) and the linkage sites for a particular carbohydrate are specified in the primary structure. Polysaccharides are large molecules built from monosaccharide units. These molecules can be linear or branched and have a multitude of possible arrangements of the basic repeating unit. Indeed, the complexity of carbohydrate structures arises from a combination of the wide variety of possible ring residues and the vast number of possible linkages between these residues.

A common simplified view of a polysaccharide is a “chain”, comprising rigid rings connected by rotatable linkages, with possible multiple branches off the main chain. The rings (monosaccharides) come in a variety of different types and are differentiated according to their type (monosaccharides are typically given common names, such as glucose), the chemical configuration of the first carbon in the ring (either α or β configuration), the enantiomeric form of the ring (either D or L) and the number of carbon atoms in the ring (5 or 6). Linkages, too, have a number of different forms, depending on where the link attaches to a ring. For example, in glucose there are five possible attachment points for a glycosidic linkage (to the C1, C2, C3, C4 and C6 carbons). The linkage specification indicates where the anomeric carbon (C1) in one sugar is linked to the carbon in the next sugar. For example, a (1-3) linkage means that C1 of one monosaccharide is linked to the C3 of the next subunit. Fig. 1 shows an α -D-glucose ring linked to the second carbon in an α -L-rhamnose ring. However, it must be remembered that the “chain” view is a highly simplified model: the rings can in fact flip conformation (e.g. glucose can flip from a chair to a boat in response to

steric crowding or strain) and the glycosidic linkages are not completely flexible, with many linkages having a characteristic and quite restricted range of motion.

B. The CASPER notation for carbohydrate primary structure.

The canonical reference for carbohydrate primary structure representation is the International Union of Pure and Applied Chemists (IUPAC) recommendation [10]. In this system, monosaccharide residues are described with three letters, together with their anomeric descriptors. Most applications use the IUPAC conventions to represent the primary structure of a carbohydrate. For example, SWEET (discussed above) allows the user to input a structure in the IUPAC format, in a table to facilitate identification of branches in the structure.

CASPER is a web based program to determine the primary structure of oligo- and polysaccharides using NMR data [6], [7] and use of CarbBuilder with CASPER now also allows the user to retrieve 3D glycan structures suitable for molecular simulations. CASPER outputs the primary structure in a format similar to that specified by IUPAC. However, the CASPER format adds square brackets (nested if need be) to indicate branches from the main structure. If there are “open” glycosidic linkages at each end of the structure, then the structure is indicated to be the repeating unit of a polysaccharide.

C. Carbohydrate simulations

The conformation and dynamics of specific carbohydrates are most commonly simulated with Molecular Dynamics (MD) methods using empirical force fields [11]–[20]. There are currently a number of new Molecular Mechanics force fields available that have been specifically adapted for carbohydrates, including the GLYCAM series of force fields for AMBER [21], OPLS-AA [22], GROMOS [23], and, for CHARMM, CSFF [24] and a new, redesigned CHARMM force field for cyclic and acyclic carbohydrates under active development by MacKerrel and co-workers [25]–[28]. These specialized force fields predict conformational equilibria more consistently than general organic force fields and are reasonably consistent with each other [29], [30].

III. DESIGN

The aim of CarbBuilder is to convert the primary structure description of carbohydrate into a reasonable 3D structure, as illustrated in Fig. 1.

In our approach, we decided not to attempt to solve the scientific question of “What is the correct structure of this carbohydrate?”, but rather to provide a reasonable estimate of the structure, allowing the user to refine the model if they so choose. The focus is on an extensible software system, that the user can adjust to suit their requirements. Essentially, our approach separates building the structure from structural investigations (which are best performed using complex molecular modelling software), aiming to provide rapidly a model from which investigations can begin. However, in order for molecular simulations to be successful, it is necessary to begin

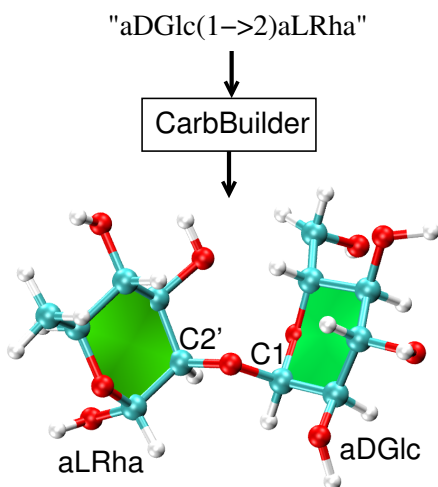


Fig. 1: The CarbBuilder module accepts a text description of the primary structure (residues and linkages) of a carbohydrate in the CASPER format and returns a three-dimensional structure in the PDB format. Here an α -D-glucose ring is linked via a glycosidic bond to the second carbon in a α -L-rhamnose ring. The rings of the α -D-Glc(1 \rightarrow 2) α -LRha disaccharide are depicted using the *PaperChain* visualization in the VMD package [9].

from a “reasonable” facsimile of the actual physical structure. What do we mean by a “reasonable model”? In the “chain” viewed of a carbohydrate (discussed above), the 3D structure (or conformation) of a polysaccharide is primarily determined by the relative rotation of successive glycosidic linkages in the chain. We determined that a reasonable model would arrange the residues correctly according to the current state of experimental knowledge of dihedral angle values, with no (physically impossible) self-intersections of the carbohydrate chain. This is a similar approach to that reported for the POLYS software [4]. However, we also decided to provide a set of possible suitable values for each monosaccharide and each dihedral angle and to make the preferred conformations user-adjustable. Self-intersections are solved with a heuristic approach, searching for an optimal conformation from a set of preferred angles. We also designed the software to be extensible, allowing users to adapt the system by adding new residues, for example of monosaccharide derivatives.

A. Support for Molecular Simulations

Although carbohydrate force fields produce similar results, they differ, sometimes dramatically, in their force expressions and parameter sets. We therefore needed to select an initial class of force fields that CarbBuilder will support. We chose the CHARMM carbohydrate force fields for reasons of familiarity and broad applicability. A CHARMM force field is described in two files: a topology file and a parameter file. The topology file defines the atom types used in the force field; the atom names, types, bonds, and partial charges of each residue type; and any patches necessary to link or otherwise

mutate these basic residues. The parameter file lists the specific numerical values for each atom type in the generic CHARMM potential function: specific spring constants and similar parameters for all of the bond, angle, dihedral, improper, and van der Waals terms in the CHARMM potential function. In order to simulate a molecular system with a force field, it is necessary to specify which residues (defined in the topology file) occur in the system, how they link together and all the bonds, angles and dihedral angles that make up this system. To do this can be a complex and error-prone task. It is much more convenient to replace the generic topology file with a specific PSF, or “protein structure file”. A PSF contains all of the molecule-specific information needed to apply a chosen force field to a specific molecular system, i.e. all the bonds, dihedrals etc. present in the system under study.

Conveniently, there is a software tool available, psfgen, that will generate PSF files, given a CHARMM topology file and a compatible PDB structure file [31]. This tool is currently distributed as a standalone program. We therefore decided to link CarbBuilder to psfgen to produce PSF files for later simulation.

IV. IMPLEMENTATION

CarbBuilder was written in the C# programming language. C# was chosen with the aim of ensuring both portability between different architectures and a future integration into a web service: C# with ASP is more compatible with web applications than Java using jsp. CarbBuilder was developed using the .Net 3.0 Framework provided by Microsoft. It can be run using terminal window on Windows (with .Net framework) or on Linux or MacOS X (with Mono) easily.

A. Software Engineering

Scientific software often is not developed with the same focus on the user as commercial products. As software is often developed on-the-fly by graduate students, there is a focus on getting the task done and a correct process of requirements gathering is typically omitted. The resulting software is often rather opaque to the user and, furthermore, typically poorly documented.

We followed a systematic, iterative approach to software development, in consultation with expert users. The iterative methodology is a cyclic software development activity, with several cycles of water fall model steps: communication and requirement gathering, planning and scheduling, analysis and design, implementation, testing and evaluation. The iterative approach has the advantages of accommodating to changes in the project specification. Each iteration produces a testable version of the final application, which may then be redesigned, modified and improved. For this project, we used four iterations to achieve the final product.

CarbBuilder consists of 4 components: a 2D format handler, a residue retrieval module, a dihedral angle rotation module and a collision detection module, as shown in Fig. 2.

1) *2D format handler*: We selected the CASPER format for input into CarbBuilder, both because of its simplicity and for the purposes of compatibility with this software. The format handler converts the primary structure description in CASPER format into a tree, or molecular graph, of residues. This allows the structure to be easily traversed. Residues are added in a stepwise fashion, from right to left in the text description. The first residue is the root of the tree, with subsequent residues children. Each residue can have only one parent, but multiple children. In the usual case, the current residue is the child of the previous residue. Branches in the structure add additional children onto the correct root node.

2) *Residue retrieval*: The Residue Retrieval module looks up the file containing the 3D structure for the current residue. Sometimes there are substituents attached to a residue, e.g. aDGlc3Ac has an acetyl substituent on position C3 of the α -D-glucose residue. In this case, CarbBuilder will first search for a residue named "aDGlc3Ac". If this is found, CarbBuilder will flag the C3 of the α -D-glucose as occupied, so that no other residue can be linked to C3. This functionality means that it is straightforward to add new monosaccharide derivatives. If the application cannot find a file for a given residue, it checks if the residue contains supported substituents (currently the chemical groups A, NAc, OMe, Ac, Me, SO4, and P), in which case they are added to the parent structure. Multiple substituents can be added to a residue in a nomenclature that is recognizable by chemists, e.g., "aDGlc3,6diAc4Me", which is an α -D-glucose with two acetyl substituents at positions 3 and 6 and one methyl at position 4. After the adding of acid and substituents, the node is treated as one residue. The application can process the rest of the sequence normally by linking to the current residue.

3) *Dihedral angle calculation*: In the dihedral angle rotation procedure, the linkages between each of the monosaccharide residues are rotated according to the specified default angles. The orientation of a specific glycosidic linkage is described by the values of the two (sometimes three) torsion angles making up the linkage, which are termed ϕ and ψ (ω for the third torsion). The dihedral angles are measured as follows:

$$\begin{aligned}\phi &= \text{H1-C1-Ox-Cx} \\ \psi &= \text{C1-Ox-Cx-Hx} \\ \omega &= \text{Ox-C6'-C5'-O5'}\end{aligned}$$

4) *Collision detection*: The structure is validated in the collision detection procedure as the application adds residues. In this recursive process, a check is made that, for all the atoms in a given residue, no atom from another residue lies within the distance of a covalent bond (determined using the Van der Waals radii of the specific atom pair). This step has to be carried out when every new residue is added to the structure. If self-intersections of the molecule are detected, CarbBuilder then selects the next pair of torsion angles from the angle file. A tree can be constructed of connected residues and dihedral angles. The tree is created on the fly, using

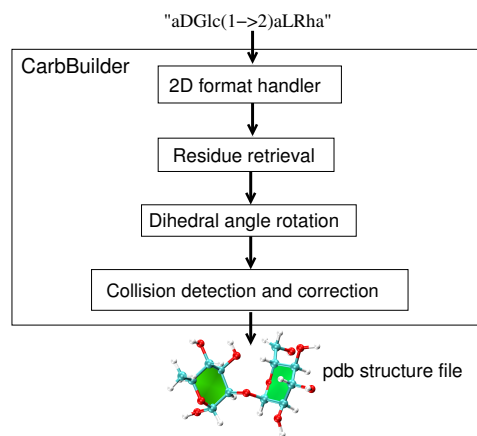


Fig. 2: CarbBuilder consists of 4 components: a 2D format handler, a carbohydrate retrieval module, an angle calculation module and a position allocation module.

depth first search as a collision happens. This means that paths are generated as they are required. The re-allocation step is carried out until no collision occurs or all dihedral possibilities are exhausted. The final values selected are listed in the output from CarbBuilder. If all dihedral possibilities are exhausted, the collision detection algorithm attempts to resolve the problem with the alternate residue structures listed in the *mapping.txt* file.

The final 3D structure is written out as a pdb file. However, in some cases, CarbBuilder may exhaust all possibilities and not be able to find an appropriate 3D structure without self-intersections. One example is the structure $\rightarrow 4[\text{bDGlcA}(1 \rightarrow 2)][\text{aLRha}(1 \rightarrow 3)]\text{aDMan}(1 \rightarrow 3)\text{aDMan6Ac}(1 \rightarrow 3)\text{bDGlcNAc}(1 \rightarrow$ where three substituents are attached to aDMan (aLRha, bDGlcA and bDGlcNAc) and they are crowded too close to together for a solution to be possible. In this case, CarbBuilder will return an error message and terminate.

After the PDB file is generated, the application calls the *ps-fgen* program to produce a PSF file for subsequent simulation studies.

B. Data files

CarbBuilder requires a number of data files, which are packaged with the software. For simplicity and ease of integration into other software packages, the CarbBuilder uses text files for linkage and default conformation data. Text files allow for easy editing, and are straight-forward to install and maintain for inexperienced users. The text files are as follows.

- 3D structure files in PDB format for each of the monosaccharide unit building blocks. The CASPER residue name is file name for default 3D position. We extracted default PDB files for most residues from the SWEETDB database. Note that, in order for CarbBuilder to successfully generate a psf file for simulation using a CHARMM force field, all residue names and atom types must conform to the CHARMM force field specification.
- *mapping.txt*: a file listing the names of the 3D structure files for each of the monosaccharide units recognized by

the system. Adding an entry to this file effectively adds a new monosaccharide residue. Each entry comprises a residue name (e.g. aDGlc), followed by a list of pdb structure file names, in descending order of preference. For example, the entry for α D-Glc (glucose) is as follows.

```
aDGlc a-D-Glcp.pdb glucose_boat.pdb
```

If it is not possible to build a valid structure with the first pdb file (the preferred low energy chair structure), the second one (a higher energy boat structure) will be chosen, and so forth. In this way, it is possible to include alternative structures for each monosaccharide, to allow CarBuilder to intelligently resolve collisions.

- *dihedrals.txt*: a file which specifies the preferred values for particular glycosidic linkages between different residue pairs. For example, the entry for the α DGlc (1 \rightarrow 4) α DGlc linkage is currently listed as follows.

```
aDGlc 1 4 aDGlc,2,-25 -25,-35 170
```

This line specifies that there are 2 dihedral angles for this linkage (ϕ and ψ) and lists three alternate values that may be used if a collision occurs, in decreasing order of preference. These values were taken from a the energy minima identified by a prior detailed computational analysis of this linkage [32]. We have attempted to provide suitable values where there is experimental or computational information available on a particular glycosidic linkage. However, where data is not available, the default torsion values will be used (discussed below). In addition, alternative dihedral values may be specific on the command line and the *dihedrals.txt* data file is freely editable by the user, allowing the system to be customized or update when more data is available.

- *dihedral_defaults.txt*: a file listing the default torsion angle values for generalised glycosidic linkages. These angles are used when torsion angles for a particular linkage are not specified in *dihedrals.txt*. For example, the entry:

```
aD, 50 0 180
```

specifies that the default for all α -D residues (for example α -D-glucose) the torsion angles should be set as $\phi = -50, \psi = 0, \omega = 180$ (ω is only used if applicable - i.e. for a 3-bond glycosidic linkage). We set the default values according to those listed in a survey of carbohydrate structures in the Protein Data Bank [33]. This file can also be edited by the user.

- A CHARMM force field topology file is required to produce a PSF for subsequent simulation. The topology files defines the atom types used in the force field; the atom names, types, bonds, and partial charges of each residue type. (The corresponding CHARMM parameter file is required only for subsequent molecular simulations.)

C. Input and output

CarBuilder has a text-based user interface. Users input the primary sequence for a carbohydrate molecule in the CASPER format, along with a number of optional flags to CarBuilder. For example, optional arguments specify whether input is straight from the command line or read from a file, as well

as the number of repetitions to generate for a polysaccharide repeat unit (default 5).

CarBuilder produces two output files: a file containing the co-ordinates of the 3D structure in the Protein Data Bank (PDB) [2] file format and a PSF which lists all of the molecule-specific structure information needed for subsequent simulation with the CHARMM force field. The PDB format is the *de facto* standard for molecular structure data and can be viewed with any of the extant molecular visualization software packages, such as VMD [34]. The PDB file and the PSF in conjunction with the corresponding CHARMM parameter file permit simulation with either the CHARMM [35] or NAMD [36] molecular simulation packages.

The CHARMM force field specification supports only limited number of monosaccharide residues. In cases where a specific residue is present in the molecule but not in the CHARMM force field, only the PDB will be created and CarBuilder will report a warning.

D. Validation

The CarBuilder software is still under development and is essentially currently a beta version. However, it has been extensively tested to ensure that it operates correctly. We used six sets of primary structures for testing the correctness of PDB output, comprising linear, branched, repeated linear and branched, and complex structures. The correctness of the 3D structures was established with the molecular visualisation software, VMD: all linkages were correctly connected, dihedral angles were correctly rotated and that no self-intersections occurred. We also tested for failure, with both incorrect (impossible) structures and pathological cases that would always self-intersect. The usability of the system was confirmed by our small team of expert users, who suggested a number of improvements which were implemented. Finally, we confirmed that, where applicable, the output structure files could be used for successful minimization using NAMD and the CHARMM carbohydrate force field.

V. RESULTS: THE CARBUILDER PROGRAM

To demonstrate the utility of CarBuilder, we will step through an illustrative example, building the polysaccharide components of the starch compound. Fig. 3a shows a fragment (20 units) of the regular linear polysaccharide, amylose. Amylose is the chief component of starch and comprises α (1 \rightarrow 4)-linked glucose residues. This large oligosaccharide is built in CarBuilder with the simple input line:

```
"->4) aDGlc(1->" -r 20
```

where the arrows at the beginning and end of the line specify a polysaccharide and the “-r 20” flag indicates that 20 repeats of the structure are required. This amylose fragment had no collisions and was built successfully using the preferred glucose residue conformation (listed in *mapping.txt*) and dihedral angle values ($\phi, \psi = -25, -25$) for the α (1 \rightarrow 4)-linkage listed in *dihedrals.txt*.

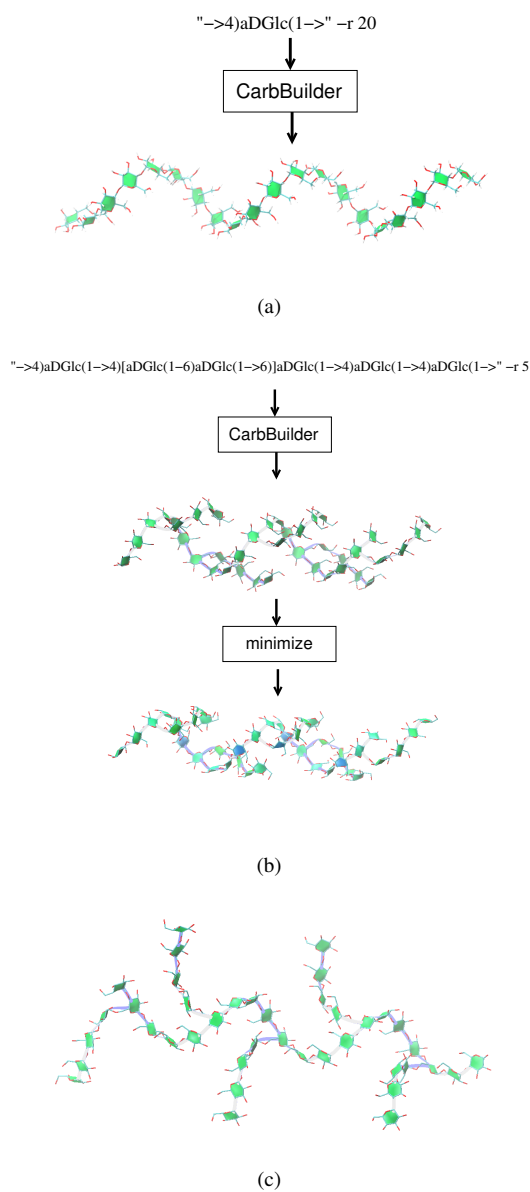


Fig. 3: Illustrative example: building starch oligosaccharides. Fig. 3a shows a 20 unit fragment of the $\alpha(1\rightarrow4)$ linked amylose polysaccharide. Fig. 3b shows a structure similar to amylopectin: comprising the same backbone as amylose, with the addition of a $(1\rightarrow6)$ branch every 4 residues. The effect of a molecular mechanics minimization procedure is also shown: the blue rings indicate that this is a strained structure. Fig. 3c shows the effect of adjusting dihedral angle values on the amylopectin structure. Structures are visualized using the VMD package, with rings highlighted using the *PaperChain* algorithm [9]. For clarity, the larger structures are shown without hydrogen atoms.

Amylopectin is also a component of starch. It is a similar molecule to amylose, but has a more complex structure, with $\alpha(1\rightarrow6)$ branches at irregular intervals along the backbone. In order to create a truly random structure in CarbBuilder, it

would be necessary to specify each residue in the structure on the command line. However, an approximation to the amylopectin molecule may be obtained by specifying a larger repeating unit, as illustrated in Fig. 3b. Here, five repeats of a eight residue unit are specified. The repeat unit comprises four $\alpha(1\rightarrow4)$ -linked glucose residues, with a $\alpha(1\rightarrow6)$ branch on the second residue comprising a further four $\alpha(1\rightarrow4)$ -linked residues – a fairly large and complex structure. As CarbBuilder assembled the structure, collisions occurred for residues close to the branch points. These were resolved in the heuristic algorithm by selecting the listed alternate ϕ, ψ values for the affected linkages: $\phi, \psi = -35, -170$. The $\alpha(1\rightarrow6)$ linkage had no values listed in *dihedrals.txt*, and so the default values listed in *dihedral_defaults.txt* were applied for the branch points: $\phi = -50, \psi = 0, \omega = 180$.

Glucose is one of the most common residues and listed in the CHARMM force field. Therefore, CarbBuilder, in conjunction with the *psfgen* program, is able to produce a PSF for both the amylose and amylopectin fragments. However, it is clear from minimization with the NAMD package that this structure is too crowded: the blue residues at the branch points in Fig. 3b indicate residue ring flips to a boat conformation – a sign of steric strain in carbohydrates [9]. Therefore, the user would likely choose to adjust the angle values listed in *dihedrals.txt*. An obvious approach is to add the missing values for the $\alpha(1\rightarrow6)$ linkage, as follows.

```
aDGlc 1 6 aDGlc,3,80 180 180,80 180 60
```

These more appropriate values were taken from a comprehensive study of the $\alpha(1\rightarrow6)$ linkage in isomaltose [17]. Fig. 3c shows the effect of this adjustment on the amylopectin structure, which is now more spread out. This is a rather nice illustration of why this linkage will result in the characteristic conformation of this very branched molecule. The resulting structure is not sterically strained at all: it minimizes with only a slight change in the original conformation and no ring flips occur.

VI. CONCLUSIONS AND FUTURE WORK

CarbBuilder takes a heuristic approach to building a carbohydrate structure, providing a reasonable structural estimate which the user can refine. In addition to building both simple and branched 3D structures (functionality currently supported by most current carbohydrate building tools), the CarbBuilder module also allows for input of repeated sequence and derived carbohydrate molecules, which no other building applications currently support. Although CarbBuilder is primarily focussed on expediting the simulation process, it can also be used simply to build a carbohydrate model in keeping with current understanding of glycosidic linkage properties. CarbBuilder is freely available and portable: it is a text-based stand-alone program that can run on Windows, Linux and MacOS X systems without installation. The system can be easily linked to other packages or to a web interface. We have demonstrated the simplicity and adaptability of this software, which we expect to be of great use to chemists studying carbohydrates.

Future extensions will add support for additional carbohydrate force fields, as well as a web-based graphical user

interface and more extensive user testing. We will also focus on refining the prediction algorithm, with a more extensive list of dihedral angle values extracted from the literature.

ACKNOWLEDGMENT

This work is based upon research supported in part by the South African National Research Foundation (NRF) as well as the Swedish Research Council.

REFERENCES

- [1] T. Lütke, A. Bohne-Lang, A. Loss, T. Goetz, and C. von der Lieth, "GLYCOSCIENCES.de: an internet portal to support glycomics and glycobiology research," *Glycobiology*, vol. 16, pp. 71R–81R, 2006.
- [2] (2005-2011, July). [Online]. Available: <http://www.glycam.com>
- [3] T. D. Goddard and T. E. Ferrin, "Visualization software for molecular assemblies," *Curr. Opin. Struc. Biol.*, vol. 17, pp. 587–595, 2007.
- [4] S. B. Engelsen, S. Cros, W. Mackie, and S. Pérez, "A molecular builder for carbohydrates: Application to polysaccharides and complex carbohydrates," *Biopolymers*, vol. 39, no. 3, pp. 417–433, 1996.
- [5] J. Rosen, L. Miguët, and S. Pérez, "Shape: automatic conformation prediction of carbohydrates using a genetic algorithm," *Journal of Cheminformatics 2009, 1:16*, vol. 1, no. 16, 2009.
- [6] M. Lundborg and G. Widmalm, "Structure analysis of glycans by nmr chemical shift prediction," *Anal. Chem.*, vol. 83, pp. 1514–1517, 2011.
- [7] P.-E. Jansson, R. Stenutz, and G. Widmalm, "Sequencing determination of oligosaccharides and regular polysaccharides using nmr spectroscopy and a novel web-based version of the computer program casper," *Carbohydr. Res.*, vol. 341, pp. 1003–1010, 2006.
- [8] M. Kuttel, J. Gain, A. Burger, and I. Eborn, "Techniques for visualization of carbohydrate molecules," *J. Mol. Graphics Modell.*, vol. 25, pp. 380–388, 2006.
- [9] S. Cross, M. M. Kuttel, J. E. Stone, and J. E. Gain, "Visualisation of cyclic and multi-branched molecules with vmd," *J. Mol. Graphics Modell.*, vol. 28, pp. 131–139, 2009.
- [10] A. D. McNaught, "Nomenclature of carbohydrates (recommendations 1996)," *Adv. Carbohydr. Chem. Biochem.*, vol. 52, pp. 43–177, 1997.
- [11] S. N. Ha, L. J. Madsen, and J. W. Brady, "Conformational analysis and molecular dynamics simulations of maltose," *Biopolymers*, vol. 27, pp. 1927–1952, 1988.
- [12] A. D. French, "Comparisons of rigid and relaxed conformational maps of maltose and cellobiose," *Carbohydr. Res.*, vol. 188, pp. 206–211, 1989.
- [13] M. K. Dowd, A. D. French, and P. J. Reilly, "Conformational analysis of the anomeric forms of sophorose, laminaribiose and cellobiose using MM3," *Carbohydr. Res.*, vol. 233, pp. 15–34, 1992.
- [14] B. Hardy and A. Sarko, "Conformational analysis and molecular dynamics simulation of cellobiose and larger celooligomers," *J. Comput. Chem.*, vol. 14, no. 7, pp. 831–847, 1993.
- [15] T. M. Glennon, Y. Zheng, S. M. le Grand, B. A. Shutzberg, and K. M. J. Merz, "A force field for monosaccharides and (1 → 4) linked polysaccharides," *J. Comput. Chem.*, vol. 15, no. 9, pp. 1019–1040, 1994.
- [16] K. J. Naidoo and M. Kuttel, "Water structure about the dimer and hexamer repeat units of amylose from molecular dynamics computer simulations," *J. Comput. Chem.*, vol. 22, no. 4, pp. 445–456, 2001.
- [17] R. B. Best, G. E. Jackson, and K. J. Naidoo, "Molecular dynamics and NMR study of the $\alpha(1\rightarrow4)$ and $\alpha(1\rightarrow6)$ glycosidic linkages: maltose and isomaltose," *J. Phys. Chem. B*, vol. 105, pp. 4742–4751, 2001.
- [18] K. Ueda, K. Iwama, and H. Nakayama, "Molecular dynamics simulation of neocarrabiose 2,4-bis(sulfate) and carrabiose 4,2-bis(sulfate) as building blocks of ι -carrageenan in water," *Bull. Chem. Soc. Jpn.*, vol. 74, no. 12, pp. 2269–2277, 2001.
- [19] F. Corzana, M. S. Motawia, C. H. D. Penhoat, S. Pérez, S. M. Tschampel, R. J. Woods, and S. B. Engelsen, "A hydration study of (1→4) and (1→6) linked alpha-glucans by comparative 10 ns molecular dynamics simulations and 500-MHz NMR," *J. Comput. Chem.*, vol. 25, no. 4, pp. 573–586, 2004.
- [20] G. A. Venter, R. P. Matthews, and K. J. Naidoo, "Conformational flexibility of sulphur linked saccharides a possible key to their glycosidase inhibitor activity," *Molecular Simulation*, vol. 34, no. 4, pp. 391–401, 2008.
- [21] K. Kirschner, A. Yongye, S. Tschampel, J. Gonzalez-Outeirino, C. Daniels, B. Foley, and R. Woods, "Glycam06: A generalizable biomolecular force field. carbohydrates," *J. Comput. Chem.*, vol. 29, pp. 622–655, 2008.
- [22] D. Kony, W. Damm, S. Stoll, and W. F. van Gunsteren, "An improved OPLS-AA force field for carbohydrates," *J. Comput. Chem.*, vol. 23, no. 15, pp. 1416–1429, 2002.
- [23] R. D. Lins and P. H. Hünenberger, "A new GROMOS force field for hexopyranose-based carbohydrates," *J. Comput. Chem.*, vol. 26, no. 13, pp. 1400–1412, 2005.
- [24] M. Kuttel, J. W. Brady, and K. J. Naidoo, "Carbohydrate solution simulations: Producing a force field with experimentally consistent hydroxyl rotational frequencies," *J. Comput. Chem.*, vol. 23, no. 13, pp. 1236–1243, 2002.
- [25] O. Guvench, S. N. Greene, G. Kamath, J. W. Brady, R. M. Venable, R. W. Pastor, and A. D. MacKerell, "Additive empirical force field for hexopyranose monosaccharides," *J. Comput. Chem.*, vol. 29, no. 15, pp. 2543–2564, 2008.
- [26] E. Hatcher, O. Guvench, and A. D. MacKerell, Jr., "CHARMM additive all-atom force field for acyclic polyalcohols, acyclic carbohydrates, and inositol," *Journal of Chemical Theory and Computation*, vol. 5, no. 5, pp. 1315–1327, 2009.
- [27] E. Hatcher, O. Guvench, and A. D. MacKerell, "CHARMM additive all-atom force field for aldopentofuranoses, methyl-aldopentofuranosides, and fructofuranose," *J. Phys. Chem. B.*, vol. 113, no. 37, pp. 12466–12476, 2009.
- [28] O. Guvench, E. Hatcher, R. M. Venable, R. W. Pastor, and J. Alexander D. MacKerell, "CHARMM additive all-atom force field for glycosidic linkages between hexopyranoses," *Journal of Chemical Theory and Computation*, vol. 5, no. 9, pp. 2353–2370, 2009.
- [29] S. M. Tschampel, K. N. Kirschner, and R. J. Woods, "Incorporation of carbohydrates into macromolecular force fields: Development and validation," in *NMR spectroscopy and computer modeling of carbohydrates: recent advances*, ser. ACS Symposium Series, J. F. G. Vliegthart and R. J. Woods, Eds., vol. 930, 2006, pp. 235–257.
- [30] C. A. Stortz, G. P. Johnson, A. D. French, and G. I. Csonka, "Comparison of different force fields for the study of disaccharides," *Carbohydr. Res.*, vol. 344, pp. 2217–2228, 2009.
- [31] (2011, July). [Online]. Available: <http://www.ks.uiuc.edu/Research/vmd/plugins/psfgen/>
- [32] M. Kuttel and K. J. Naidoo, "Glycosidic linkage rotations determine amylose stretching mechanism," *J. Am. Chem. Soc.*, vol. 127, pp. 12–13, 2005.
- [33] M. F. Thomas Lütke and C.-W. von der Lieth, "Carbohydrate structure suite (CSS): analysis of carbohydrate 3D structures derived from the PDB," *Nucleic Acids Res.*, vol. 33, pp. D242–D246, 2005.
- [34] W. Humphrey, A. Dalke, and K. Schulten, "VMD – Visual Molecular Dynamics," *J. Molec. Graphics*, vol. 14, pp. 33–38, 1996.
- [35] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, "CHARMM: A program for macromolecular energy, minimization and dynamics calculations," *J. Comput. Chem.*, vol. 4, no. 2, pp. 187–217, 1983.
- [36] L. Kalé, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten, "NAM2: Greater scalability for parallel molecular dynamics," *J. Comp. Phys.*, vol. 151, pp. 283–312, 1999.