Bioinformatics, 35(14), 2019, 2434–2440
doi: 10.1093/bioinformatics/bty990
Advance Access Publication Date: 7 December 2018
Original Paper

OXFORD

Structural bioinformatics

GlycanFormatConverter: a conversion tool for translating the complexities of glycans

Shinichiro Tsuchiya[1], Issaku Yamada[2] and Kiyoko F. Aoki-Kinoshita[1,3,*]

[1]Graduate School of Engineering, Soka University, Hachioji, Tokyo 192-8577, Japan, [2]The Noguchi Institute, Itabashi, Tokyo 173-0003, Japan and [3]Faculty of Science and Engineering, Soka University, Hachioji, Tokyo 192-8577, Japan

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on June 29, 2018; revised on October 1, 2018; editorial decision on November 24, 2018; accepted on December 6, 2018

Abstract

Motivation: Glycans are biomolecules that take an important role in the biological processes of living organisms. They form diverse, complicated structures such as branched and cyclic forms. Web3 Unique Representation of Carbohydrate Structures (WURCS) was proposed as a new linear notation for uniquely representing glycans during the GlyTouCan project. WURCS defines rules for complex glycan structures that other text formats did not support, and so it is possible to represent a wide variety glycans. However, WURCS uses a complicated nomenclature, so it is not human-readable. Therefore, we aimed to support the interpretation of WURCS by converting WURCS to the most basic and widely used format IUPAC.

Results: In this study, we developed GlycanFormatConverter and succeeded in converting WURCS to the three kinds of IUPAC formats (IUPAC-Extended, IUPAC-Condensed and IUPAC-Short). Furthermore, we have implemented functionality to import IUPAC-Extended, KEGG Chemical Function (KCF) and LinearCode formats and to export WURCS. We have thoroughly tested our GlycanFormatConverter and were able to show that it was possible to convert all the glycans registered in the GlyTouCan repository, with exceptions owing only to the limitations of the original format. The source code for this conversion tool has been released as an open source tool.

Availability and implementation: https://github.com/glycoinfo/GlycanFormatConverter.git

Contact: kkiyoko@soka.ac.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Glycans are biomolecules that consist of a small number (2–20) of monosaccharide residues connected by glycosidic linkages. The term polysaccharide is typically used to denote any linear or branched polymer consisting of monosaccharide residues, such as cellulose. Thus, the relationship of monosaccharides to oligosaccharides or polysaccharides is analogous to that of amino acids to proteins, or nucleotides to nucleic acids (Varki *et al.*, 2009). Diverse structures can be created by simply linking different monosaccharides through glycosidic linkages, to make oligosaccharides or polysaccharides (Fig. 1). The common classes of glycans are primarily defined according to the nature of the linkage to the aglycone (protein or lipid). A glycoprotein is a glycoconjugate in which a protein carries one or more glycans covalently attached to the polypeptide backbone, usually via *N*- or *O*-linkages in mammalian organisms. An *N*-glycan makes a glycosidic linkage with the side-chain nitrogen of an asparagine residue that is a part of a consensus peptide sequence NX(S/T). An *O*-glycan makes a glycosidic linkage with the terminal oxygen of a serine or threonine residue (Packer *et al.*, 2017).

Many bacterial glycoconjugates are associated with the cell envelope and cell surfaces, and some are essential for viability. Surface glycoconjugates drive a variety of important interactions with elements of the host innate and adaptive immune defenses. They show a wide diversity in structures and often feature sugars not found

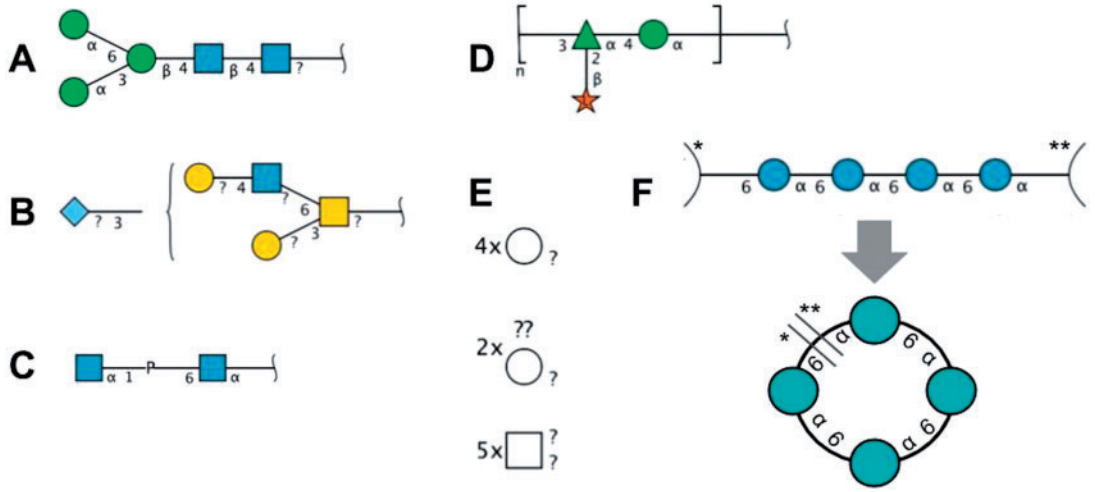

Fig. 1. Illustration of various diverse glycans. These glycans are represented based on the symbolic rules of Structure Nomenclature for Glycans (SNFG). (**A**) The basic *N*-Glycan core structure. (**B**) Glycan fragments. (**C**) Cross-linked substituent. (**D**) Repeating structure. (**E**) Monosaccharide compositions. (**F**) Cyclic structure

Table 1. Comparison with text formats

Format	Style	Repeating units	Cyclic units	Structural ambiguity	Support for non-monosaccharides	Rare monosaccharides	Human readability
IUPAC	linear text	✓					✓
KCF	connection table	✓	✓	✓	✓		
LinearCode	linear text	✓	✓	✓	✓		✓
GlycoCT{Condensed}	connection table	✓	✓	✓		✓	
WURCS	linear text	✓	✓	✓	✓	✓	

elsewhere in nature. Antigens of many pathogenic microorganisms have carbohydrate origin, and recognition of bacteria by the host immune system is determined by the structure of these compounds. The bacterial saccharide sequences have a greater diversity of monosaccharides, with certain monosaccharides being specific to certain groups of bacteria. Bacterial glycan structures form a very complicated structure that is unlike in any other organisms. Therefore, it is difficult to visualize and encode these structures.

Due to the development of glycan databases, several formats for representing glycan structures have been developed. Many of these formats represent glycans using a connection table, or adjacency matrix. The features of the glycan sequence formats utilized in this study are shown in Table 1. 'Style' indicates the format of each representation, which is either linear text or a connection table. Usually, a connection table format divides monosaccharides and linkages, whereas linear text is a single string representing the glycan structure. 'Repeating units' indicates whether it is possible to handle a repeating glycan structure (Fig. 1D). 'Cyclic units' indicates whether it is possible to handle cyclic glycan structures (Fig. 1F). 'Structural ambiguity' indicates whether it is possible to handle ambiguous structures such as glycan fragments (Fig. 1B). 'Support for non-monosaccharides' indicates whether it is possible to handle aglycones such as amino acids. Modifications such as N-acetylation of monosaccharides does not apply to this. 'Rare monosaccharides' indicates whether it is possible to accurately represent non-mammalian monosaccharides such as 'Lgro-L3, 9dmanNon-2-ulop5N7NFormyl-onic'. 'Human readability' indicates whether the text is in general human readable. This usually applies to linear text formats.

The International Union of Pure Applied Chemistry—International Union of Biochemistry and Molecular Biology (IUPAC—IUBMB) has specified the 'Nomenclature of Carbohydrates' to describe complex oligosaccharides based on a three-letter code to represent monosaccharides (McNaught, 1997; Sharon, 1986). Each monosaccharide code is preceded by the anomeric descriptor and the configuration symbol. The ring size is indicated by an italic *f* for furanose or *p* for pyranose. The carbon numbers that link the two monosaccharide units are given in parentheses between the symbols separated by an arrow.

The KEGG Chemical Function (KCF) format for representing glycan structures was originally used to represent chemical structures in Kyoto Encyclopedia of Genes and Genomes (KEGG) (Hashimoto *et al.*, 2006; Hattori *et al.*, 2003). It represents the first published sequence format for saccharides which uses a connection table approach. KCF uses the graph notation, where nodes are monosaccharides and edges are glycosidic linkages. The NODE section of the format describes the building blocks (monosaccharides) sequentially. The list is numbered and its members constitute the nodes of a graph. The EDGE section describes the linkages (glycosidic linkages) between the entries defined in the NODE section. The KCF description includes X, Y coordinates for each node, used for drawing purposes. An extension of the format allows repeating structures to be encoded, using an additional BRACKET section.

LinearCode® is a new syntax for representing glycoconjugates and their associated molecules in a simple linear fashion. This format is used to represent glycans in the Consortium Functional Glycomics (CFG) (Raman *et al.*, 2006). It uses a single letter code to represent each monosaccharide and includes a condensed

description of the connections between monosaccharides (Banin *et al.*, 2002). Modifications to the common structure are indicated by specific symbols. For example, D-Galp is the common form of galactose, thus its code A is used alone. However, if it is in a furanose form, it would be written as A'. Linkage information is represented using the symbols a and b for α and β, respectively. This is followed by the carbon number of the parent to which the residue is attached.

GlycoCT was developed as a part of the EuroCarbDB project (Herget *et al.*, 2008; Lieth *et al.*, 2010; Ranzinger *et al.*, 2008). This format is used in UniCarbKB (Campbell *et al.*, 2014) and ExPASy (Artimo *et al.*, 2012). GlycoCT uses a similar graph concept to the KCF format and consists of two varieties: a condensed format and an XML format. The monosaccharide namespace consists of five components and basically follows those defined by IUPAC: the base type, anomeric configuration, the monosaccharide name with configurational prefix, chain length indicator, ring forming positions and further modification designators. Trivial names such as fucose or rhamnose are not permitted in GlycoCT. Because GlycoCT is based on very complicated nomenclatures, it is difficult for humans to describe. Thus, EUROCarb has published MolecularFramework. MolecularFramework is an open source software that converts glycan texts such as KCF and IUPAC to GlycoCT.

Web3 Unique Representation of Carbohydrate Structures (WURCS) was proposed as a new linear notation for representing carbohydrates for the Semantic Web during the GlyTouCan project (Aoki-Kinoshita *et al.*, 2016). This format supports the representation of nonstandard monosaccharide units as a part of the glycan structure, as well as compositions, repeating units and ambiguous structures where linkages/linkage positions are undefined (Tanaka *et al.*, 2014). Moreover, WURCS was updated to additionally handle ambiguous monosaccharide structures (Matsubara *et al.*, 2017). WURCS represents monosaccharides by using a ResidueCode, which is similar to the Extended Stereocode used for representing monosaccharides in MonosaccharideDB (http://www.monosaccharidedb.org/).

The aim of GlyTouCan is to simplify the identification of glycan structures and to help link related research with the many life sciences databases available worldwide. A unique accession number is generated to every unique glycan structure, which can be used for reference in any glycan-related research or publications. Thereby, WURCS is designed to represent a wide variety of glycan structures in linear notation, but, it has poor human-readability. It was considered that there was a need to provide a human-readable format translated from WURCS on GlyTouCan. Since IUPAC is the most widely used human-readable format, there was a need for a converter for glycans to/from WURCS.

In this paper, we report the development of GlycanFormatConverter. This tool implements two types of conversion functions between IUPAC and WURCS. This conversion functionality can be executed with IUPAC-Extended, KCF, LinearCode® and WURCS. To verify the conversion accuracy, we did the following:

1. The glycans registered in KEGG and CFG were converted to WURCS.
2. All glycans registered in GlyTouCan were converted to IUPAC and reconverted to WURCS.
3. We compared whether the reconverted WURCS matches the WURCS registered in GlyTouCan.

As a result, we were able to show that the conversion between WURCS and IUPAC implemented in GlycanFormatConverter succeeded to encode glycan sequences with a wide variety of glycan structures and indicated high conversion accuracy. GlycanFormatConverter has been developed as an open source tool, and the source code has been released at https://github.com/glycoinfo/GlycanFormatConverter.git.

2 Materials and methods

2.1 Development environment and resources

The GlycanFormatConverter was developed utilizing Java version 7, and it has been tested on Mac (OS X version 10.12.6). We used the WURCSFramework library to implement WURCS input/output functionalities. WURCSFramework is an open source library developed in Java and generates three types of data structures from a WURCS string (https://github.com/glycoinfo/wurcsframework.git).

2.2 Development of the GlycanFormatConverter

First, we designed the data structure to handle glycan structures. We named this data structure *GlyContainer* and show the architecture in Figure 2. *GlyContainer* defines components of carbohydrates using basically two types of objects. *Node* is the most basic object, and it is a superclass of *Monosaccharide* and *Substituent*. On the other hand, *GlycanUndefinedUnit* is an object for handling ambiguous information for any building blocks.

Monosaccharides have the following members: *AnomericStateDescriptor*, *anomeric position*, *Stereo*, *SuperClass*, *Modifications*, *RingStart* and *RingEnd*. *AnomericStateDescriptor* enumerates the anomeric states as α, β or unknown. *SuperClass* enumerates the size of monosaccharide, which should be between a value between 3 and 10 indicating the number of carbons forming the monosaccharide backbone. *Stereo* stores the most basic monosaccharide three letter code (Supplementary Table S2.1). *GlycoModification* is an object for defining the molecular state of each backbone carbon, such as deoxy. This object contains a binding position and a *ModificationTemplate* listing molecular information about the modification.

Repeating units and substituent information such as N-acetyl groups are handled by *Substituent*. Substituent notation is stored in two types of dictionaries in *SubstituentTemplate* and *CrossLinkedTemplate*, and these dictionaries are integrated into *SubstituentInterface*. *SubstituentTemplate* enumerates single linkage substituents. This object can support 42 types of substituents (Supplementary Table S2.2). *CrossLinkedTemplate* enumerates multiple-linkage substituents. This object can support 15 types of substituents (Supplementary Table S2.3). *Edge* stores nodes and linkages for both donor and acceptor. *Linkage* has an acceptor side position, probability annotation and LinkageType. *LinkageType* enumerates the molecular state of each linkage.

2.3 Implementation of text conversion functionality

We implemented four types of sequence parsers including IUPAC-Extended, KCF, LinearCode® and WURCS in the GlycanFormatConverter. In each sequence parser, each format was broken down into node or edge, and these notations were utilized to generate the GlyContainer. This object was utilized to output WURCS, IUPAC-Short, IUPAC-Condensed and IUPAC-Extended.

In the implementation of the WURCS parser and export functionalities, we utilized objects defined in the WURCSFramework library. WURCSFramework provides three types of objects including WURCSArray, WURCSSequence2 and WURCSGraph, which can all be constructed from a WURCS string. WURCSGraph supports

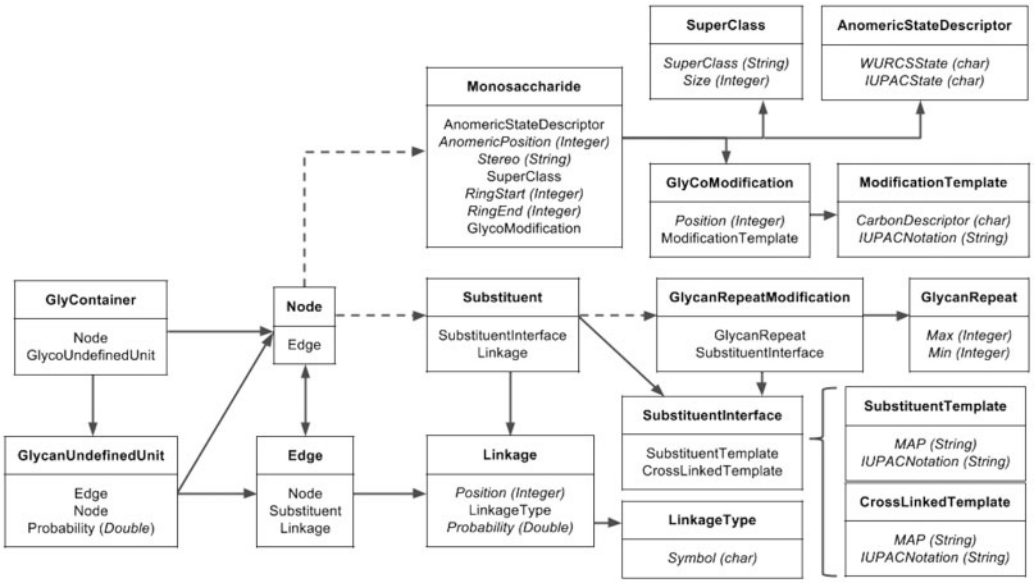

Fig. 2. The architecture of GlyContainer, the main object used in GlycanFormatConverter. Each box indicates a class in GlyContainer. The top field lists the name of the class. The bottom area lists the member(s) defined in the class. Italic members are primitive data types. Non-italic members are other classes. Arrows indicate a dependency between objects. Dotted arrows indicate an inheritance relationship. For example, Node is inherited by both Monosaccharide and Substituent

Table 2. Examples of our proposed extension of IUPAC sequences for glycans that cannot be represented by the original IUPAC recommendation format

Type of glycan	Example sequences
Compositions	{?-?-HexNAc-(? →}5,{?-?-Hexp??-(1 →}2,{?-?-Hexp-(1 →}4
Cyclic	6)-α-D-Glcp-(1 →6)-α-D-Glcp-(1 →6)-α-D-Glcp-(1 →6)-α-D-Glcp-(1 →
Cross-linked substituent	β-D-Glcp-(1-S →4)-β-D-Glcp(1 →
Fragments	?-D-Neup5Gc-(2 →3)=1$, 1$?-D-Galp-(1 →3)[1$?-D-Galp-(1 →4)-1$?-D-GlcpNAc-(1 →6)]-1$?-D-GalpNAc-(1 →
Multiple linkages	α-D-Neup5Ac-(2 →8: 1 →9)-α-D-Neup5Ac-(2 →
Probabilistic components	α-D-ManpNAc-(1 →4)[β-D-GlcpNAc-(1 →30%3)-[4)-β-D-ManpA2NAc-(1 →4)-β-D-GlcpNAc-(1 →6)]-α-D-GlcpNAc-(1 →]n

many detailed molecular states of monosaccharides, so it was most suitable to implement the IUPAC conversion functionality. Thus, we attempted to build an object converter between GlyContainer and WURCSGraph. Also, WURCS conversion functionality utilized methods in WURCSFramework.

2.4 Extension of the IUPAC nomenclature

In this study, we encoded IUPAC based on the 'Nomenclature of Carbohydrates'. However, IUPAC cannot represent all the glycan structures registered in GlyTouCan, and it was difficult to encode ambiguous glycan structures such as glycan fragments. Therefore, we needed to devise an extension to IUPAC to handle these unique glycan structures. For cyclic structures, glycan fragments and cross-linked substituents, the nomenclature utilized in CFG and the Complex Carbohydrate Structure database (CCSD) (Doubet *et al.*, 1989) was adapted to IUPAC. We show examples of expanded IUPAC notation in Table 2. Monosaccharide compositions (Fig. 1E) are represented by surrounding each building block by parentheses, followed by their cardinality. Cyclic structures (Fig. 1F) use parentheses to represent the cyclic position. Cross-linked substituents (Fig. 1C) are indicated by inserting modification information within the glycosidic linkage. Glycan fragments (Fig. 1B) are shown by assigning an ID to each fragment. Each ID is also assigned to the monosaccharide to which the fragment can bind. If more than one bond exists between two monosaccharides, the linkage information is separated by a colon. Probability annotation is indicated by inserting probability information within the glycosidic linkage. When the substituent is probabilistic, the probability is indicated in the linkage position of the substituent. More detailed rules for describing these glycan structures with IUPAC are listed in Supplementary Section S1.

2.5 Validation of conversion accuracy

We attempted to verify the accuracy of the conversion function between IUPAC and WURCS. First, 98 925 text strings of WURCS were collected from GlyTouCan, converted to IUPAC-Extended and the converted IUPAC was reconverted to WURCS. The reconverted WURCS was compared with the WURCS before conversion to see if they completely matched. To verify the conversion accuracy to/from KCF, we searched the glycan structures linked with KEGG GLYCAN in GlyTouCan, and as a result, the KCF formats of 10 370 glycan structures were obtained. We converted the KCF collected from KEGG into WURCS and compared it with the corresponding WURCS registered in GlyTouCan. To verify the conversion accuracy to/from LinearCode, 1217 glycan structures were collected from GlycomeAtlas (Konishi and Aoki-Kinoshita, 2012). In GlycomeAtlas, GlyTouCan accession numbers and their

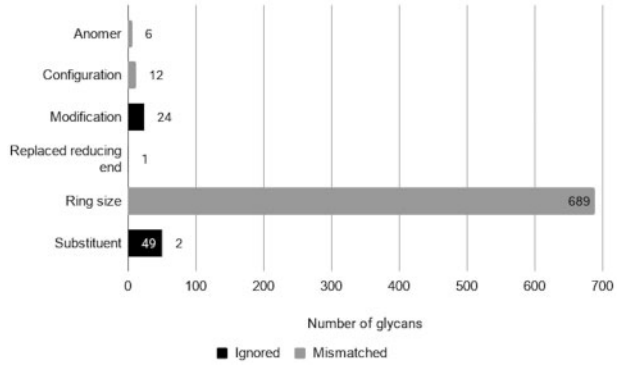

Fig. 3. The causes of mismatches between the 783 out of 10 370 WURCS converted from KCF. This validation found six types of problems from 783 glycans. 'Ignored' refers to a lack of representation for building blocks such as monosaccharides or substituents in the process of conversion. 'Mismatched' refers to a difference of notation between the original WURCS and the converted WURCS

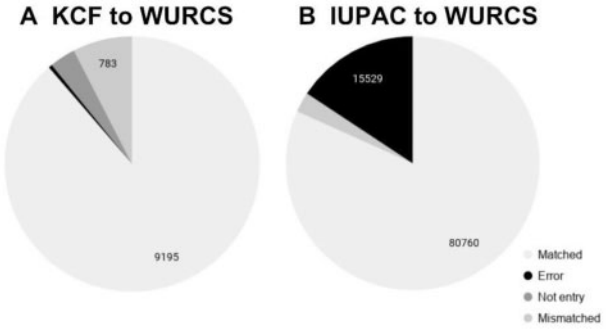

Fig. 4. Conversion result of GlycanFormatConverter. (**A**) 9195 out of 10 321 glycans were successfully converted between KCF and WURCS. Among the rest, 783 did not match, 343 could not be verified and 49 resulted in errors. (**B**) 83 300 out of 98 829 glycans were successfully converted between IUPAC to WURCS. Among the rest, 80 760 matched, 2540 did not match and 15 529 resulted in errors

LinearCode are assigned to each registered glycan structure. We converted the LinearCode to WURCS and compared it with the WURCS registered in GlyTouCan.

3 Results

3.1 Validation of the conversion accuracy of KCF to WURCS

We succeeded in converting 10 321 glycans in KCF format into WURCS (Fig. 4A). Of these, the majority, 9195, completely matched with the corresponding WURCS registered in GlyTouCan. On the other hand, 1126 WURCS sequences did not match. We found that among these, 343 actually did not have a link to GlyTouCan, so they could not be compared with the converted WURCS. For the remaining 783 that were linked to GlyTouCan, we analyzed the causes of this discrepancy (Fig. 3). Examples of the mismatched structures are listed in Supplementary Section S3.1.

There were basically six causes of mismatches, and each cause could be classified as 'Ignored' or 'Mismatched'. 'Ignored' indicates that some structural information of WURCS converted from KCF was lost, and these were broken down into ignored modifications, substituents and reducing end. 'Ignored modification' indicates that the WURCS had monosaccharides that had lost the specific state of molecules, such as deoxy. 'Ignored substituent' indicates that monosaccharides had lost substituents such as sulfate groups. 'Replaced reducing end' indicates that there were different numbers of monosaccharides between KEGG and GlyTouCan, usually due to different representations of the reducing end. 'Mismatched' indicates that the structural information of WURCS converted from KCF was different from the WURCS in GlyTouCan. These were broken down into mismatched anomer, configuration and ring size. 'Mismatched anomer' indicates that the anomeric state of one or more monosaccharides were different. 'Mismatched configuration' indicates that the stereoisomer of one of more monosaccharides were different. 'Mismatched ring size' indicates that the ring size of one or more monosaccharides were different.

Of the 10 370 KCF text, 49 could not be converted to WURCS. Among these, non-monosaccharides such as amino acids were represented in places other than the root. GlycanFormatConverter could not handle such non-monosaccharides if they were not substituents.

3.2 Validation of the conversion accuracy of IUPAC to WURCS

The results of validation of WURCS to IUPAC to WURCS conversion are shown in Figure 4B. We succeeded in reconverting 83 300 glycans to WURCS, among which 80 760 WURCS matched perfectly with the original WURCS from GlyTouCan. On the other hand, 15 524 WURCS strings failed in the conversion to IUPAC, and five types of IUPAC strings failed in the conversion back to WURCS. Examples of these mismatched glycans are shown in Supplementary Section S3.2. In this section we categorize the types of mismatches that occurred in the conversion of glycan structures and describe the details of their causes. We show details of the causes of the errors in the conversion between 'IUPAC to WURCS' and 'WURCS to IUPAC' in Table 3. For each cause of error we categorized them into either 'Conversion issue' or 'Format limitation'. Conversion issue essentially stemmed from an issue in the WURCSFramework library, whereas Format limitation indicates that information difficult to handle in the IUPAC format was included in the glycan structure. The majority of conversion errors in WURCS to IUPAC were due to unsupported substituents: 'Unsupported substituent', 'Unsupported substituent with double anchor', 'Unsupported substituent with multiple anchor' and 'Unsupported cross-linked substituent'. We identified 3478 types of substituents from these errors. Also, some errors occurred in the conversion from ResidueCode to its trivial monosaccharide name. 'Invalid superclass' occurred in reading 321 glycans containing monosaccharides composed of 10 or more backbone carbons such as a 2112x22122h. Conversely, 'Unsupported residue' occurred in reading 623 glycans containing monosaccharides composed of less than 10 backbone carbons such as AOCm. 'Opposite linkage' occurred in four glycans, where the relationship between donor and acceptor was reversed in WURCS. The reversal of the order of monosaccharides was often seen with phosphate cross-linked with gulose. This is a problem with the WURCSFramework library not able to handle this case, about which we have notified the developers. The following two types of glycan structures failed during the import into WURCSFramework. 'Unsupported substituent' occurred in glycans with carboxyl ethyl groups. More details about these errors and possible fixes are provided in the Supplementary Section S3.3.

In Figure 5, we show the causes of mismatches found in the 2540 glycan structures that did not match. 'Mismatched double cross-linkage position' indicates a difference in linkage position that occurred in cross-linked substituents containing a pyruvic acid.

Table 3. A list of conversion errors that arose for 'WURCS to IUPAC' and 'IUPAC to WURCS'

Conversion	Type of error	Error	No of Glycans
WURCS to IUPAC	Conversion issue	Error in WURCSFramework	2
		Invalid superclass	321
		Unsupported residue	623
		Opposite linkage	4
	Format limitation	Unsupported double cross-linkage substituent	489
		Unsupported substituent	13 952
		Unsupported substituent with double anchor	3
		Unsupported substituent with multiple anchors	132
IUPAC to WURCS	Conversion issue	Unsupported substituent	3
Total number of glycans utilized for conversion			98 829

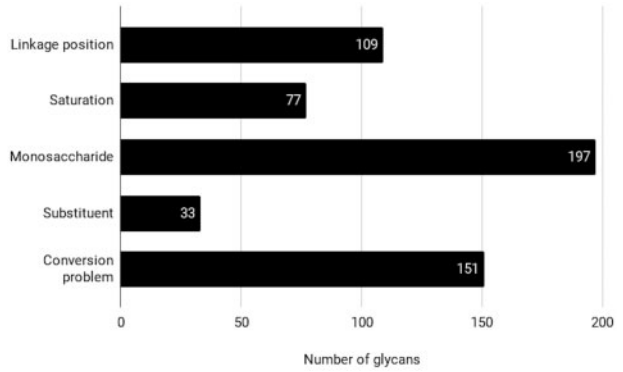

Fig. 5. The causes of mismatches between the 2540 out of 98 829 WURCS converted from IUPAC and the original WURCS from GlyTouCan. There were six types of mismatches: Mismatched linkage position, Mismatched saturation, Mismatched monosaccharide, Mismatched substituent and some other Conversion problem. There were no mismatches due to missing structure information as seen in the conversion from KCF to WURCS

'Mismatched saturation' indicates that the saturation of some monosaccharide was different. For example, the monosaccharide represented as AEe22h was reconverted to AFf22h because IUPAC does not make such distinctions. 'Mismatched monosaccharide' indicates that the composition of the ResidueCode was different in some monosaccharide. As an example, consider the ambiguous monosaccharide represented as a26h-1b_1-4. This monosaccharide is translated to erythrose (Ery) in IUPAC, which was reconverted to the less ambiguous form a22h-1b_1-4 in WURCS, resulting in the mismatch. This mismatched also occurred in many glycans containing methylated monosaccharides. 'Mismatched substituent' indicates that the notation of the substituent was different in the reconverted WURCS. 'Conversion problem' indicates that two or more of the above issues arose during the conversion process.

In addition, out of the 2540 glycan structures that were mismatched, the majority, 1973, were originally monosaccharide compositions with linkage information, which contain glycosidic linkages on the acceptor side of each monosaccharide. Compared to monosaccharide compositions without linkage information, this form of composition indicates whether or not the glycan has the possibility of being cyclic or contains multiple glycosidic linkages between the same two monosaccharides. It also enables the accurate calculation of its mass. Unfortunately, IUPAC is unable to represent such monosaccharide compositions, as it automatically assumes that the glycan is a 'tree' structure. Therefore, these glycans will never match.

3.3 Validation of the conversion accuracy of LinearCode® to WURCS

All the LinearCode® strings obtained from GlycomeAtlas were converted to WURCS. However, most of glycans registered in GlycomeAtlas were not assigned accession numbers in GlyTouCan. Accession numbers were assigned to 259 out of the total 1217 glycans, and as a result, the WURCS sequences obtained from these glycans all completely matched with WURCS in GlyTouCan.

4 Discussion

In this study, we developed GlycanFormatConverter which can conversely convert between WURCS and IUPAC. The functionality to output IUPAC from WURCS contributes to providing human-readable text formats of glycans in GlyTouCan. We showed that this tool can also be executed using IUPAC-Extended, KCF and LinearCode®. In other words, we made it possible to utilize GlyTouCan not only with GlycoCT and WURCS, but also with other major glycan representation formats.

In the verification process of conversion via KCF and IUPAC, there were some glycans that could not be converted accurately. In analyzing the KCF data, we thought that there was a problem with the data that GlyTouCan referenced from other databases, as GlyTouCan integrates glycan information of various databases and in the integration process converts every data to WURCS format. For example, the glycan information from KEGG is represented in KCF, which is converted first to GlycoCT through the MolecularFramework library. In analyzing the functionality of importing KCF and outputting GlycoCT using MolecularFramework, structural information of such monosaccharides as arabinose and ribose were not completely reflected in GlycoCT. Furthermore, aglycone information such as amino acids are omitted, so if an amino acid is found between two monosaccharides in the middle of a glycan, it would be removed in GlycoCT, resulting in an incomplete and different glycan structure. Because GlyTouCan has used GlycomeDB as its original source for glycan structures, GlyTouCan also contains the incomplete version of such glycans. Due to our validation procedure, we were able to gleen out such inconsistencies within the GlyTouCan data, which has been reported.

The issue above is caused by the fact that modifications in GlycoCT are based on a library of known modifications. However, a glycan may consist of two glycan parts that are bridged by a peptide sequence. This would require GlycoCT to store a library of all possible peptide sequences, which is not realistic. Therefore, such glycans are currently beyond the scope of GlycoCT.

Regarding IUPAC, we found that GlycanFormatConverter has many unsupported monosaccharides and substituents. GlycanFormatConverter handles 49 kinds of substituent residues in

a dictionary format. However, as a result of investigating all the glycans registered in GlyTouCan, 3478 substituents were unsupported. We assumed that unsupported substituents will increase with the registration of new glycan structures to GlyTouCan. As a countermeasure against this, it is necessary to implement a function to convert the substituents without using a dictionary. WURCSFramework has some initial implementation of such functionality, which will be investigated in future work. WURCS format is capable of describing all structural information of glycans. Examples of structural information include molecular states between monosaccharides or between monosaccharide and substituent groups. In contrast, the IUPAC nomenclature was not designed to handle such detailed structural information. Therefore, many errors arose from the inability of IUPAC to capture the details in WURCS, which were subsequently lost or resulted in errors when trying to reconvert them back to WURCS.

The majority of monosaccharides that did not match in the conversion from WURCS to IUPAC was due to ambiguous representations. It is difficult to perfectly represent the ambiguity of monosaccharides with IUPAC's simple three letter code. Because IUPAC does not provide rules for representing ambiguous forms of monosaccharides, IUPAC strings end up with the default representation of monosaccharides, which are translated to less ambiguous forms in WURCS. Additionally, IUPAC is unable to handle ambiguity in glycan structures as a whole. Thus, we proposed additions to the IUPAC nomenclature for handling such structures (Supplementary Material S1). Before approaching the IUPAC commission regarding the consideration of adopting our recommendations, we will first approach the glycoinformatics community, such as GLIC (GlycoInformatics Consortium; http://glic.glycoinfo.org), to try to get agreement from them. Such an agreement will make it easier to propose a well-defined and well-supported recommendation for IUPAC.

In the future, we will improve the import process so that GlycanFormatConverter can handle a larger variety of glycan text formats. In particular, since GlycoCT is utilized by most researchers in the field of glycoinformatics, conversion from/to GlycoCT is indispensable. At present, there currently exists a translator for generating WURCS from GlycoCT (https://github.com/glycoinfo/glycocttowurcs.git) however it would be most useful to incorporate this functionality into GlycanFormatConverter. Moreover, in the current version of GlycoCT (2.0), rules for representing all glycan structures are insufficient. Therefore, we need to discuss with users/developers of GlycoCT to define all necessary rules in order to completely implement GlycoCT conversion functionality.

5 Conclusion

The GlycanFormatConverter was able to encode WURCS from IUPAC-Extended, KCF and LinearCode® for the great majority of glycans registered in GlyTouCan. Those that could not be matched were mainly caused by limitation of the intermediate format. Thus this tool provides bioinformaticians with a new tool to aid in obtaining WURCS sequences. Furthermore, it allows users to register structures in GlyTouCan using representations other than the currently supported WURCS and GlycoCT. By making it possible to convert WURCS to IUPAC-Extended, IUPAC-Condensed and IUPAC-Short formats, it has become possible to represent glycans in a human-readable format. This tool has been released under a GNU GPL license, and it can be downloaded from https://github.com/glycoinfo/GlycanFormatConverter.git.

Acknowledgements

The authors would like to thank M. Matsubara, N. Miura and Y. Takahashi for meaningful discussions.

Funding

This work was supported by the Database Integration and Coordination Program sponsored by the National Bioscience Database Center (NBDC) of the Japan Science and Technology Agency (JST).

Conflict of Interest: none declared.

References

Aoki-Kinoshita,K.F. (2010) *Glycome Informatics: Methods and Applications*. CRC Press, Tylor & Francis Group.

Aoki-Kinoshita,K.F. *et al.* (2016) GlyTouCan 1.0—the international glycan structure repository. *Nucleic Acids Res.*, **44**, D1237–D1242.

Artimo,P. *et al.* (2012) ExPASy: SIB bioinformatics resource portal. *Nucleic Acids Res.*, **40**, W597–W603.

Banin,E. *et al.* (2002) A novel linear code nomenclature for complex carbohydrates. *Trends Glycosci. Glycotechnol.*, **14**, 127–137.

Campbell,M.P. *et al.* (2014) UniCarbKB: building a knowledge platform for glycoproteomics. *Nucleic Acids Res.*, **42**, D215–D221.

Doubet,S. *et al.* (1989) The complex carbohydrate structure database. *Trends Biochem. Sci.*, **14**, 475–477.

Hashimoto,K. *et al.* (2006) KEGG as a glycome informatics resource. *Glycobiolotgy*, **16**, 63R–70R.

Hattori,M. *et al.* (2003) Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *J. Am. Chem. Soc.*, **125**, 11853–11865.

Herget,S. *et al.* (2008) GlycoCT—a unifying sequence format for carbohydrates. *Carbohydrate Res.*, **343**, 2162–2171.

Konishi,Y. and Aoki-Kinoshita,K.F. (2012) The GlycomeAtlas tool for visualizing and querying glycome data. *Bioinformatics*, **28**, 2849–2850.

Lieth,C.-W.V.D. *et al.* (2010) EUROCarbDB: an open-access platform for glycoinformatics. *Glycobiology*, **21**, 493–502.

Matsubara,M. *et al.* (2017) WURCS 2.0 update to encapsulate ambiguous carbohydrate structures. *J. Chem.Inf. Model.*, **57**, 632–637.

McNaught,A. (1997) Nomenclature of carbohydrates. *Carbohydrate Res.*, **297**, 1–92.

Packer,N.H. *et al.* (2017) *Oligosaccharides and Polysaccharides. Essentials of Glycobiology*, 3rd edn, Cold Spring Harbor Laboratory Press, NY.

Raman,R. *et al.* (2006) Advancing glycomics: implementation strategies at the Consortium for Functional Glycomics. *Glycobiology*, **16**, 82R–90R.

Ranzinger,R. *et al.* (2008) GlycomeDB—integration of open-access carbohydrate structure databases. *BMC Bioinformatics*, **9**, 384.

Sharon,N. (1986) IUPAC-IUB Joint Commission on Biochemical Nomenclature (JCBN). Nomenclature of glycoproteins, glycopeptides and peptidoglycans. *Glycoconjugate J.*, **3**, 123–133.

Tanaka,K. *et al.* (2014) WURCS: the Web3 unique representation of carbohydrate structures. *J. Chem. Inf. Model.*, **54**, 1558–1566.

Varki,A. *et al.* (2009) *Structural Basis of Glycan Diversity. Essentials of Glycobiology*, 2nd edn, Cold Spring Harbor Laboratory Press, NY.