Development of data repositories for glycan-related resources

March 2024

Yushi Takahashi

Contents

Abstract

Glycosylation is known as one of the major modifications to various biomolecules, including proteins and lipids. Unlike DNA or protein molecules, glycans are diverse and complex branching structures formed by bonds between monosaccharide molecules, called glycosidic bonds. As it is now known that glycan molecules bound to proteins and lipids contribute to the fine-tuning of interactions between their substrates and other biomolecules, advances in the field of glycoscience are becoming increasingly important in many research areas, including medicine, pharmacy, and materials engineering.

To understand the precise interactions between various biomolecules in organisms, including proteins, glycans, and lipids, the importance of large-scale multi-omics analysis is increasing. In order to accelerate and promote such large-scale analysis, it is essential to share the results of various experiments and analyses, and most recently, mass spectrometry and liquid chromatography technologies have increasingly produced large amounts of experimental data. For this purpose, many public data repositories have been developed in recent years in the life science field as an infrastructure for researchers to upload and publish the results of their experiments on the Internet when they write their papers. These data repositories include PRIDE (PRoteomics IDEntifications database) and PeptideAtlas for proteomics, GlyTouCan, an international glycan repository which can assign a unique identifier to each glycan structure, and GlycoPOST and UniCarb-DR for glycomics. These data repositories assign a unique identifier to each submission from researchers, allowing them to uniquely present their research data to readers by indicating the identifiers in their papers. The mission of these data repositories is to share and accumulate the extremely valuable experimental and analytical results submitted by researchers in accordance with the FAIR (Findability, Accessibility, Interoperability, and Re-usability) data principles.

In addition, in order to reanalyze these accumulated research data using computers, it is also necessary to consider various experiment condition information, such as the sample preparation methods, the types of reagents, solution concentrations, reaction times, and various settings of the analytical equipments. For reporting such qualitative and quantitative experiment results, several standard reporting guidelines have been proposed, including the "Minimum Information About a Proteomics

Experiment" (MIAPE) guidelines for proteomics, and the "Minimum Information Required for A Glycomics Experiment" (MIRAGE) guidelines for glycomics. These guidelines in each research field summarize the minimum experimental information that must be reported together when reporting sample preparation methods, mass spectrometry experiments, liquid chromatography experiments, and capillary electrophoresis experiments, for example.

In the field of glycoscience, several public data repositories have been developed and released under the GlyCosmos project, including GlyTouCan, GlycoPOST and UniCarb-DR. GlycoPOST allows researchers to submit their mass spectrometry-based experiment results in glycomics, including raw data from mass spectrometers, according to the MIRAGE guidelines. UniCarb-DR is a data repository for mass spectrometry experiment identification results annotated using GlycoWorkbench, an identification assistant software for glycomics mass spectrometry experiments, and according to the MIRAGE guidelines, each of the identification results are should be submitted and made available as well. In particular, GlycoPOST is expected to serve as a valuable foundation for the large-scale reanalysis of a number of raw data from mass spectrometers by accumulating them.

These data repositories have established a foundation for accumulating and sharing much of the data from glycoscience research. Unlike protein molecules, however, glycan molecules do not function alone in organisms. Rather, they serve as modulators of the functions of substrate molecules such as proteins and lipids. Therefore, in order to understand the functions of glycans in orgnisms, it is also necessary to accumulate information on the substrates to which they are attached. Despite this need, there has been no informatics infrastructure to collectively capture information on glycoconjugates such as glycopeptides, glycoproteins, glycolipids, and glycosides. As for glycan structures, GlyTouCan uniquely assigns an identifier to each glycan structure, allowing it to collect and store glycan structure information. In order to collect and accumulate diverse glycoconjugate information while strictly distinguishing the types of attached glycan structures and their binding sites, a system like GlyTouCan that can assign a unique identifier to each glycoconjugate entry was necessary.

Moreover, there were some issues among the existing data repositories for glycomics. First, although GlycoPOST and UniCarb-DR are both data repositories for submitting the results of mass spectrometry experiments in glycomics, they are separate data repositories that are independent of each other. Therefore, users need to submit data to these data repositories separately, even if they are the results of the same experiment. Also, while GlycoPOST allows users to freely set the publication date of their submitted data, and in addition, allows only reviewers to view their submitted data during the peer review period (embargo period) of the paper they have submitted for review,

UniCarb-DR is difficult to use for users who are in the process of writing a paper because submitted data are immediately made public. Whereas GlycoPOST can accept a variety of data including raw data obtained from experiments quickly, it does not have any data visualization or flexible search functions due to this variability in the type of the data it accepts. On the other hand, UniCarb-DR can visualize the results of mass spectrometry experiments annotated with GlycoWorkbench software, and users can search registered data with various conditions, including peak information from mass spectrometry experiments, species, and glycan structure composition information, etc. Therefore, it was important to strengthen the integration between these complementary data repositories to maximize user-friendliness.

Based on this background, in this study, I conducted the following research as part of the development of repositories for glycoscience-related data:

- Development of a novel data repository, GlyComb, to facilitate the accumulation of glycoconjugate information
- Improve user-friendliness by strengthening the integration between two existing data repositories, GlycoPOST and UniCarb-DR

Through these efforts, I attempted to make these data repositories the basis of a comprehensive system for future glycoscience research that can store information not only on glycan structures but also on various glycoconjugates such as glycopeptides and glycoproteins. In this doctoral dissertation, I report the results of these studies and provide a perspective on the possibilities of glycoinformatics in glycoscience research.

Keywords: Data repository, Data integration, Glycoscience, Glycoproteomics, Glycomics, Semantic web, Functional programming

Chapter 1

Introduction

1.1 Glycans and glycoconjugates

Glycans are the third major biological macromolecule after DNA and proteins, and they are biosynthesized by glycosyltransferases and glycosidases. While glycosyltransferases add the carbohydrate moiety of sugar-nucleotide molecules to their substrates as monosaccharides, glycosidases are involved in the biosynthesis of glycan molecules by removing monosaccharide residues from their substrates.

Each monosaccharide molecule that constitutes a glycan molecule has more than five carbon atoms in its carbon skeleton, and such monosaccharides usually form a cyclic structure in aqueous solution. Here, the monosaccharide forming the five-membered ring is called furanose, and the monosaccharide forming the six-membered ring is called pyranose. For example, the D-glucose molecule forms a cyclic structure by a hemiacetal bond between an aldehyde group containing a C-1 carbon atom and an OH group attached to a C-5 carbon atom in the molecule. When forming the ring structure, the C-1 carbon atom becomes a chiral carbon atom, which is called the anomeric carbon atom. There are two types of anomers: α-anomers and β-anomers. For example, two different disaccharide structures can be formed from two molecules of D-glucose: maltose when a glycosidic bond is formed between the C-1 and C-4 carbon atoms, and gentiobiose when a glycosidic bond is formed between the C-1 and C-6 carbon atoms. These glycosidic bonds are called α-1,4 and β-1,6 bonds, respectively. The anomeric information of the C-1 carbon atoms is included in the names of each bond. Figure 1.1 shows these two types of disaccharide molecules resulting from the varying glycosidic bonds between two D-glucose molecules.

Glycosylation is a biochemical reaction in which glycans are attached to substrates such as proteins and lipids. By attaching to these substrates, glycans can contribute to fine-tuning the interactions between their substrates and other biomolecules (Varki et al., 2022), It is known that glycosylated

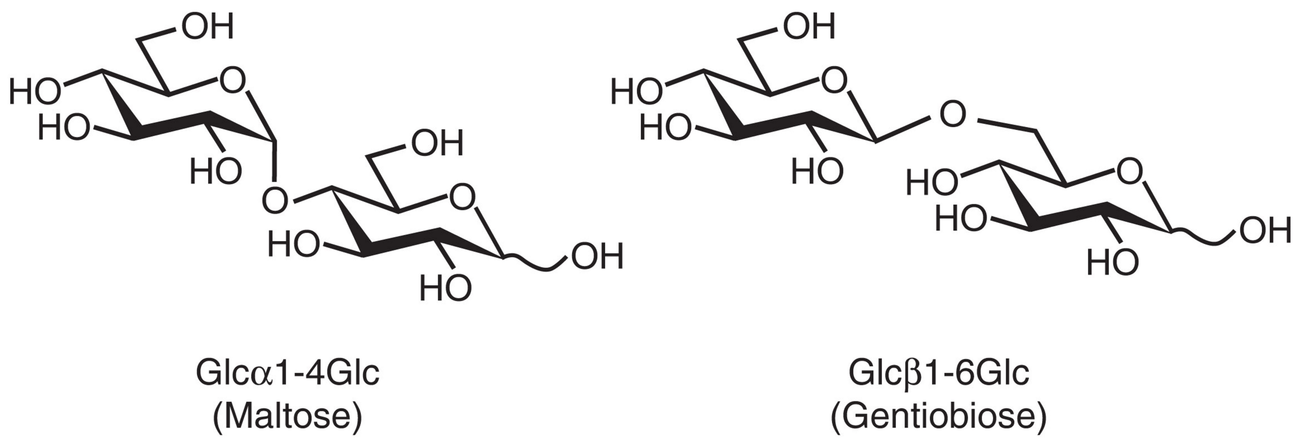


FIGURE 1.1: Two disaccharide molecules, maltose (Glc α1-4 Glc) and gentiobiose (Glc β1-6 Glc), are produced by a glycosidic bond between two D-glucose molecules. This figure was taken from chapter 2 of Essentials of Glycobiology 4th edition (Varki et al., 2022).

proteins are presented on the cell surface or secreted out of the cell to mediate various types of intercellular communication, cell-extracellular matrix interactions, and cell-molecule interactions (Day et al., 2015). Thereby, glycosylation to proteins and lipids is reported to be closely related to the mechanisms of various diseases such as influenza, cancer, and muscular dystrophy (Skehel and Wiley, 2000; Michele et al., 2002; Bao et al., 2009). Therefore, the analysis of post-translational protein modifications, including glycosylation, is essential to accurately understand the functions of each protein molecule and the interactions among them.

Biomolecules which include glycan structures that are covalently bound to various substrates such as proteins, peptides, lipids, and other compounds are called glycoconjugates; these can be broadly classified into glycopeptides, glycoproteins, peptidoglycans, glycolipids, glycosides, and so forth. Several major classes of glycosylation in glycoconjugates are well known. First, in amino acid sequences of peptides and proteins, the glycan structures attached to the nitrogen atom in the asparagine residues in the consensus sequence of Asn-X-Ser/Thr (X is any amino acid residue) are called *N*-linked glycan structures and are known to have a core structure consisting of five monosaccharides including two *N*-acetylglucosamine (GlcNAc) residues and three mannose (Man) residues. *N*-linked glycan structures are further classified into three types according to the type of remaining structure bound to the terminal mannose residues: high mannose type, hybrid type, and complex type. Another major class of glycan structures are those attached to oxygen atoms in serine or threonine residues in the amino acid sequence called *O*-linked glycan structures; there are eight known core structures from Core-1 to

Core-8. Figure 1.2 shows examples of glycan structures in these classes.

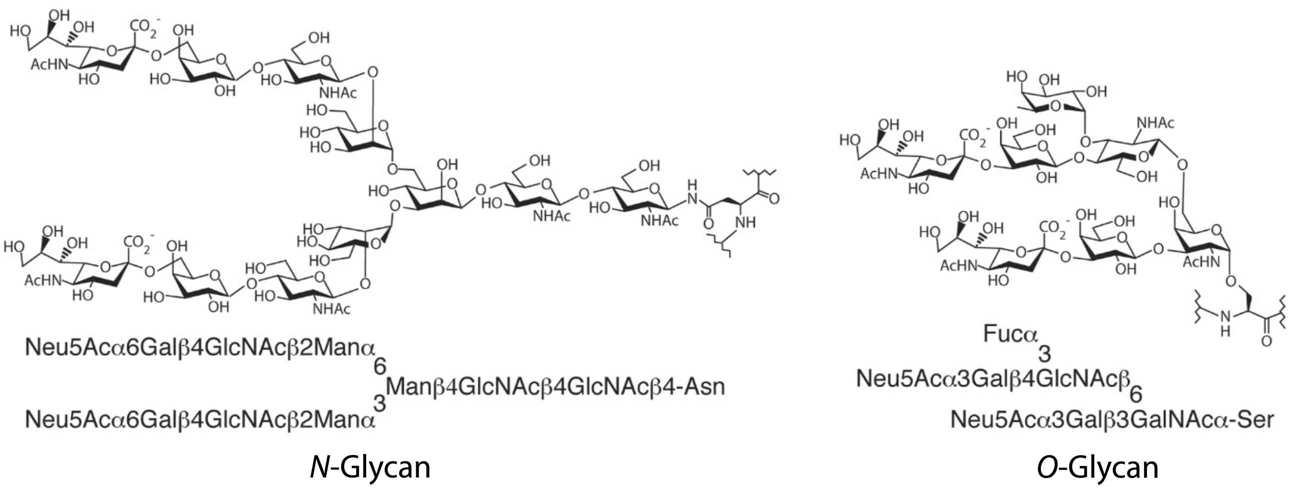

Neu5Acα6Galβ4GlcNAcβ2Manα 6
Manβ4GlcNAcβ4GlcNAcβ4-Asn
Neu5Acα6Galβ4GlcNAcβ2Manα 3
N-Glycan
Fucα 3
Neu5Acα3Galβ4GlcNAcβ 6
Neu5Acα3Galβ3GalNAcα-Ser
O-Glycan

Figure 1.2: Examples of *N*-linked glycan structures attached to asparagine residues and *O*-linked glycan structures attached to serine/threonine residues in amino acid sequences. This figure was taken from chapter 3 of Essentials of Glycobiology 4th edition (Varki et al., 2022).

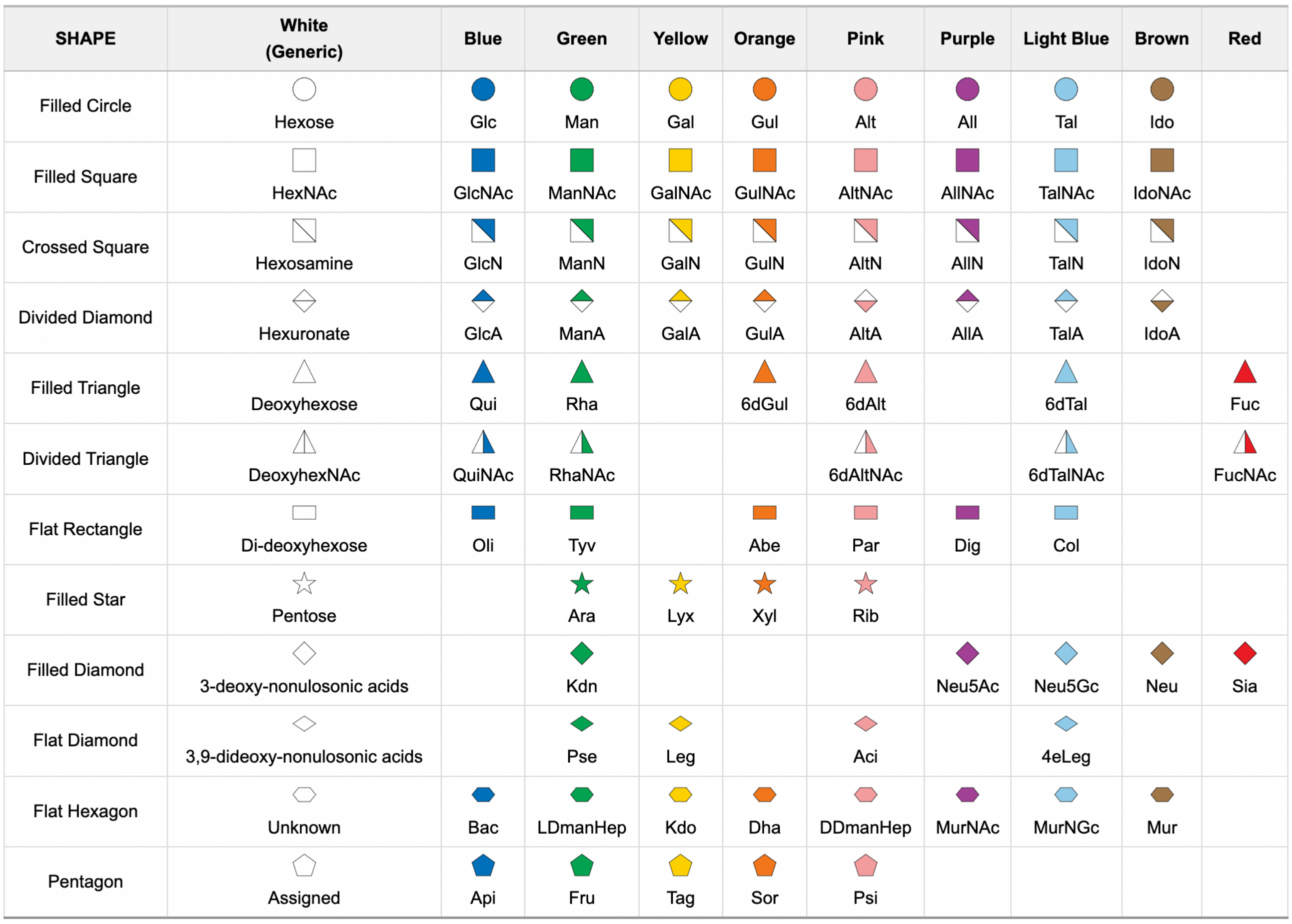
Recent advances in glycoproteomics have led to the mapping of *N*-glycosylation sites in biofluids and cellular materials obtained from patients. This implies that the occupancy of *N*-glycosylation sites in proteins and the signatures of glycopeptides could be used as biomarkers for the diagnosis of neurological diseases such as Alzheimer's disease and for the early detection of cancer (Chen et al., 2021; Zhang et al., 2020; Sinha et al., 2019).

1.2 Glycan structures

Although the chemical formulas shown in Figures 1.1 and 1.2 accurately represent the structures of glycan molecules, they have the disadvantage that it is difficult to understand what the structures are at a glance, especially for biologists. Therefore, in the field of glycoscience research, the Symbol Nomenclature For Glycans (SNFG) (Varki et al., 2015) notation is widely used to graphically represent glycan structures by symbolically drawing frequently occurring monosaccharides. In addition, in the field of glycoinformatics, where glycan-related information is analyzed using computers, various text notations of glycan structures have been proposed to handle glycan structures on computers. In this section, the standard representation formats for glycans are described.

1.2.1 Symbol Nomenclature For Glycans (SNFG) notation

In general, a monosaccharide molecule has many stereoisomers, making it difficult to distinguish a monosaccharide structure represented by a chemical formula from other monosaccharide structures. Therefore, to facilitate communication among scientists in the glycoscience community, the SNFG notation is used to represent the symbolic structure of glycans. It uses nine colors and 12 shapes to symbolically represent monosaccharide molecules commonly found in mammals and other organisms. Figure 1.3 shows the list of monosaccharide symbols defined in SNFG as of December 15, 2023. Also, Figure 1.4 shows the glycan structures shown in Figure 1.2, drawn using the symbols defined in the SNFG monosaccharide notation.



SHAPE	White (Generic)	Blue	Green	Yellow	Orange	Pink	Purple	Light Blue	Brown	Red
Filled Circle	Hexose	Glc	Man	Gal	Gul	Alt	All	Tal	Ido	
Filled Square	HexNAc	GlcNAc	ManNAc	GalNAc	GulNAc	AltNAc	AllNAc	TalNAc	IdoNAc	
Crossed Square	Hexosamine	GlcN	ManN	GalN	GulN	AltN	AllN	TalN	IdoN	
Divided Diamond	Hexuronate	GlcA	ManA	GalA	GulA	AltA	AllA	TalA	IdoA	
Filled Triangle	Deoxyhexose	Qui	Rha		6dGul	6dAlt		6dTal		Fuc
Divided Triangle	DeoxyhexNAc	QuiNAc	RhaNAc			6dAltNAc		6dTalNAc		FucNAc
Flat Rectangle	Di-deoxyhexose	Oli	Tyv		Abe	Par	Dig	Col		
Filled Star	Pentose		Ara	Lyx	Xyl	Rib				
Filled Diamond	3-deoxy-nonulosonic acids		Kdn				Neu5Ac	Neu5Gc	Neu	Sia
Flat Diamond	3,9-dideoxy-nonulosonic acids		Pse	Leg		Aci		4eLeg		
Flat Hexagon	Unknown	Bac	LDmanHep	Kdo	Dha	DDmanHep	MurNAc	MurNGc	Mur	
Pentagon	Assigned	Api	Fru	Tag	Sor	Psi				

FIGURE 1.3: Table of monosaccharide symbols defined in the SNFG notation as of December 15, 2023.

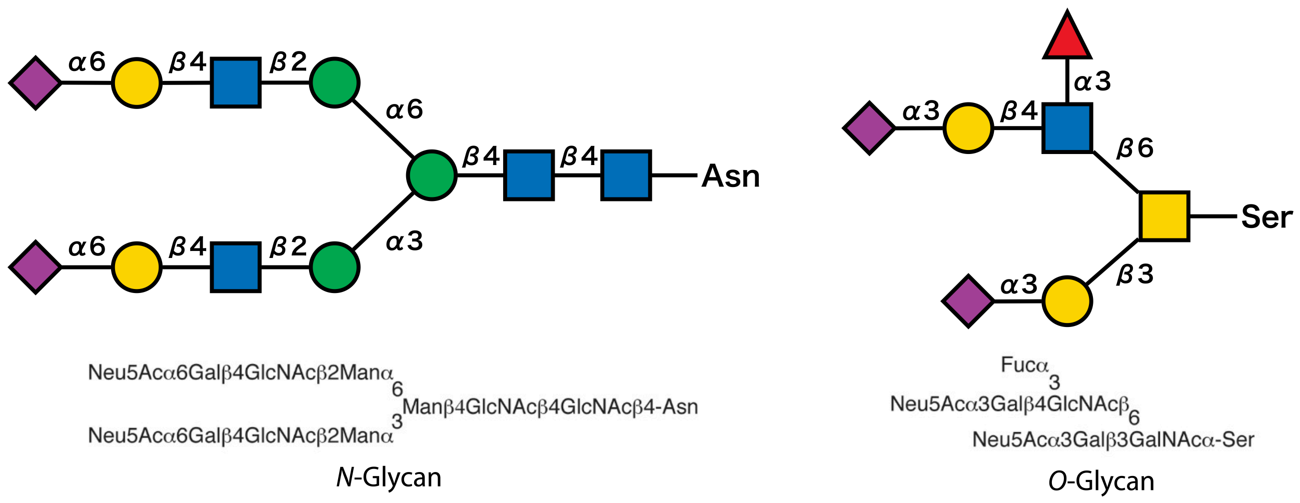


FIGURE 1.4: The glycan structures shown in Figure 1.2, drawn using the symbols defined in the SNFG notation.

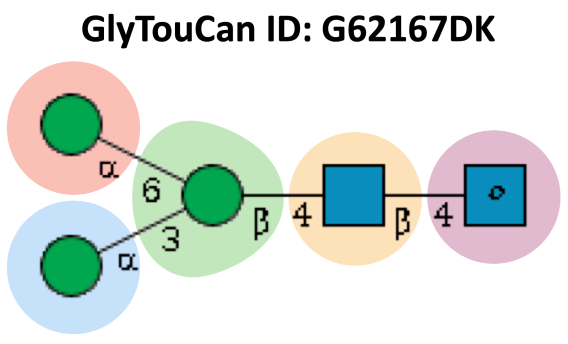
1.2.2 Glycan text notations

In the field of glycoinformatics, various glycan structure notations have been developed to handle glycan structures on computers, including the IUPAC-condensed format, GlycoMinds LinearCode format (Banin et al., 2002), KEGG Chemical Function (KCF) format (Kanehisa et al., 2004), GlycoCT format (Herget et al., 2008), and the Web3 Unique Representation of Carbohydrate Structures (WURCS) format (Matsubara et al., 2017). Here, I briefly describe two glycan structure notations that are relevant to this study: the GlycoWorkbench sequence (GWS) format (Ceroni et al., 2008; Damerell et al., 2012) and WURCS format. Using the example of the core structure of *N*-linked glycans shown in Figure 1.5, in which the aldehyde group of the reducing terminal GlcNAc is reduced to alditol after ring-opening, I compare the two notations of this structure in these two glycan structure notations.

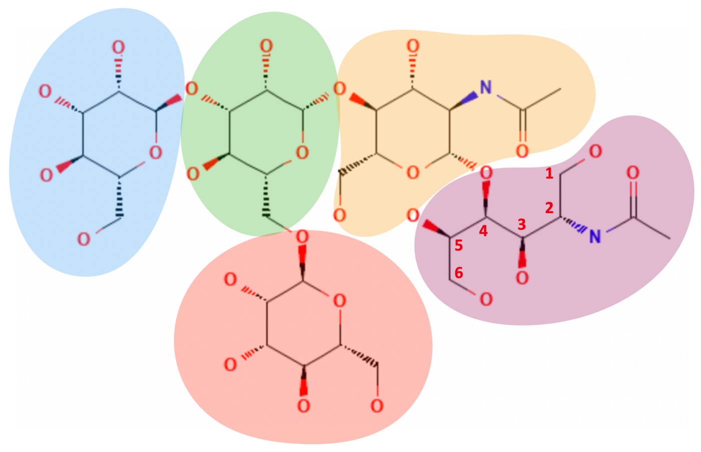
The GlycoWorkbench sequence (GWS) format is a notation format used to describe glycan structures in GlycoWorkbench (Ceroni et al., 2008), a software program that assists in the identification of the results of mass spectrometry experiments. Figure 1.5 (d) shows the core structure of the *N*-linked glycan shown in Figure 1.5 (a) in GWS format. In GWS format, each monosaccharide information is placed in order from the reducing end by separators "--". Each glycosidic linkage information is placed before the monosaccharide name of the non-reducing terminal side. For example, "4b1D-GlcNAc,p" indicates that the pyranose form of D-GlcNAc is attached to the reducing end monosaccharide by a β-1,4 bond.

The WURCS format, on the other hand, is a glycan structure notation format that can linearly and

(a)



(b)



(c)

```
RES
1b:o-dglc-HEX-0:0|1:aldi
2s:n-acetyl
3b:b-dglc-HEX-1:5
4s:n-acetyl
5b:b-dman-HEX-1:5
6b:a-dman-HEX-1:5
7b:a-dman-HEX-1:5
LIN
1:1d(2+1)2n
2:1o(4+1)3d
3:3d(2+1)4n
4:3o(4+1)5d
5:5o(3+1)6d
6:5o(6+1)7d
```

(d)

```
redEnd--?b1D-GlcNAc,p--4b1D-GlcNAc,p--4b1D-Man,p(--3a1D-Man,p)
--6a1D-Man,p$MONO,Und,-H,0,redEnd
```

(e)

```
WURCS=2.0/4,5,4/[h2122h_2*NCC/3=O][a2122h-1b_1-5_2*NCC/3=O]
[a1122h-1b_1-5][a1122h-1a_1-5]/1-2-3-4-4/a4-b1_b4-c1_c3-d1_c6-e1
```

FIGURE 1.5: The core structure of *N*-linked glycans, in which the aldehyde group of the reducing terminal GlcNAc is reduced to alditol after the ring-opening. (a) A graphical representation of this glycan structure drawn using SNFG monosaccharide notation and the corresponding accession number, GlyTouCan ID (G62167DK), in GlyTouCan (Fujita et al., 2021), the international glycan structure repository. The five background colors respectively correspond to each component of the chemical formula shown in Figure 1.5 (b) and substring in the glycan text notations shown in Figures 1.5 (c) to (e). (b) Chemical formula of this glycan structure from PubChem Substance (Kim et al., 2016) (SID 252280436). The carbon atoms participating in the carbon skeleton of the reducing terminal GlcNAc are numbered from 1 to 6. (c) Same glycan structure described in GlycoCT format. In this notation, monosaccharide residues and linkage information are described in separate sections. (d) Same glycan structure described in GWS format. (e) Same glycan structure described in WURCS format. In this case, there are two α-anomeric mannoses in the structure, resulting in this monosaccharide appearing twice in the list of linkage information (red and blue).

uniquely describe glycan structures having ambiguous monosaccharide types and linkage information. Moreover, it can also describe glycan structures that contain "repeating" or "fragmented" substructures, as well as structures that contain only monosaccharide composition information. In particular, it is adopted as the standard glycan structure notation format in the international glycan structure repository, GlyTouCan. As described later, mass spectrometry experiments are a major structural identification method in current glycoscience research, and the glycan structures obtained as a result of these experiments often do not contain information on some or all of the bonds in the structure, or only contain information on the composition of monosaccharides. Thus, WURCS is capable of handling such structures to uniquely assign accession numbers to them. In addition, the WURCS format is highly compatible with Semantic Web technologies that enable integrated retrieval of information among multiple databases, as will be described later. Therefore, it is important to use the WURCS format for glycan structures as a means to retrieve various glycan-related resources on the Internet. The glycan structure shown in Figure 1.5 is described in WURCS format as follows. Figure 1.5 (e) shows the same *N*-linked glycan core structure shown in Figure 1.5 (a) in WURCS format. In this format, the numbers of residues and bonds in the structure, the unique monosaccharide structures, the sequence of monosaccharide residues, and the linkage information are separated by "/".

1.2.3 Subsumption relationships among glycan structures

As described in the previous section, the WURCS format can uniquely represent glycan structures that contain ambiguities. Taking full advantage of this property, GlyCosmos, one of the portal sites for glycoscience research, defines relationships between glycan structures with respect to structural ambiguity called "subsumption relationships" (Zhang and Edwards, 2021). This is a binary relationship between glycan structures in which a glycan structure that contains more ambiguity in its structure "subsumes" a more specific glycan structure. As of December 15, 2023, GlyCosmos classifies each registered glycan structure into the following five levels.

1. Fully-defined saccharide
2. Linkage-defined saccharide
3. Topology
4. Monosaccharide composition
5. Base composition

The "Fully-defined saccharide" level refers to glycan structures in which all glycosidic linkage information, including anomers and carbon numbers, has been identified. The "Linkage-defined saccharide" level classifies glycan structures that contain at least one identified glycosidic linkage information in the structure. The "Topology" level is for glycan structures in which the links between all the monosaccharides in the glycan are known, but none of the details of each linkage is known. In the "Monosaccharide composition" level, only the number of monosaccharides whose stereochemistry is defined, such as Gal and GlcNAc, is known, and no information on links is known. Finally, the "Base composition" level classifies glycan structures that contain only the number of base types, such as hexoses and deoxyhexoses, are known, but no information on links is known. Figure 1.6 shows examples of glycan structures corresponding to each of these levels taking the core structure of a *N*-linked glycan as an example.

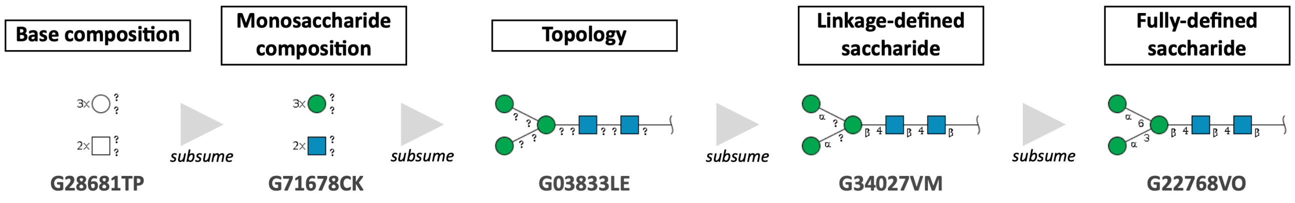


FIGURE 1.6: An example of subsumption relationships between glycan structures. In these five glycan structures, the structural information is more identified toward the right side, and the structural information is more ambiguous toward the left side.

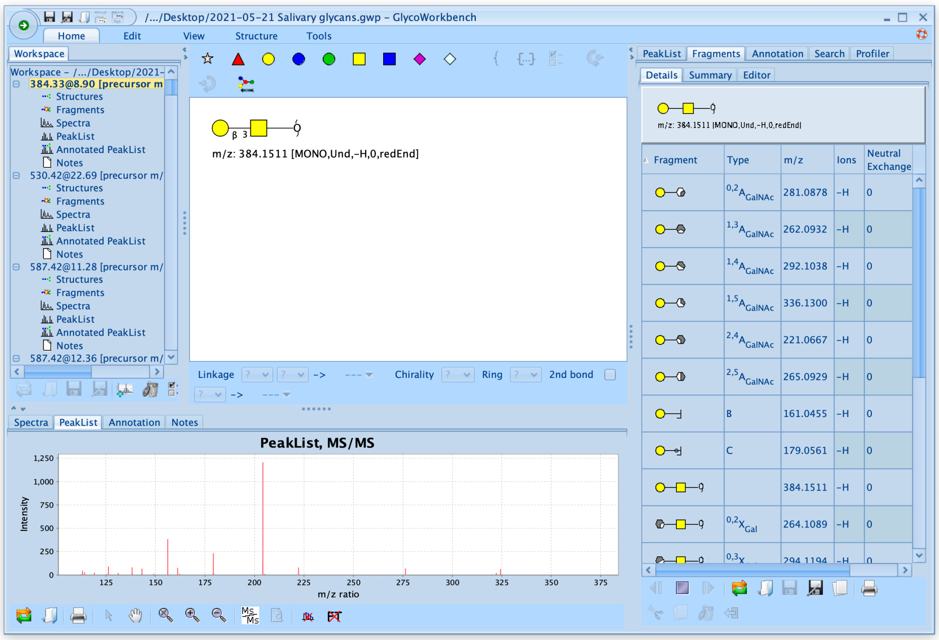
The subsumption relationships among glycan structures are useful for identifying existing glycan structures that are related to a monosaccharide composition, for example, which is obtained mainly from mass spectrometry experiments.

1.3 Mass spectrometry as a primary experimental method in glycoscience research

In glycomics, glycan structures obtained by removing attached glycans on glycopeptides and glycoproteins using glycosidases such as PNGase F are subject to analysis. Also, the intact glycoproteomics approach in glycoproteomics analyzes peptides and proteins that are still glycosylated to prevent loss of information on the amino acid residues where the glycan structures are bound. The main experimental analysis methods of them are tandem mass spectrometry experiments such as MS/MS or LC-MS/MS.

For example, in MS/MS experiments, when two mass spectrometers are coupled and a specific peak separated by the first mass spectrometer (MS1) is selectively introduced into the second mass spectrometer (MS2), the selected ions are further broken down into smaller fragments in a collision cell placed between MS1 and MS2. Then, more detailed structural information on the ion can be obtained by querying the database for a more detailed peak list obtained by MS2. This allows the identification of glycan structures in glycomics by MS/MS.

Researchers perform glycan structure identification manually by using assistant software such as GlycoWorkbench. Figure 1.7 shows the user interface of GlycoWorkbench. GlycoWorkbench outputs the results of glycan structure identification in GWP format. The results, as well as the raw data from mass spectrometer, will be submitted to data repositories for glycomics data, such as GlycoPOST and UniCarb-DR.



m/z: 384.1511 [MONO,Und,-H,0,redEnd]

Fragment	Type	m/z	Ions	Neutral Exchange
	$^{0,2}A_{GalNAc}$	281.0878	-H	0
	$^{1,3}A_{GalNAc}$	262.0932	-H	0
	$^{1,4}A_{GalNAc}$	292.1038	-H	0
	$^{1,5}A_{GalNAc}$	336.1300	-H	0
	$^{2,4}A_{GalNAc}$	221.0667	-H	0
	$^{2,5}A_{GalNAc}$	265.0929	-H	0
	B	161.0455	-H	0
	C	179.0561	-H	0
		384.1511	-H	0
	$^{0,2}X_{Gal}$	264.1089	-H	0
	$^{0,3}X$	294.1194	-H	0

Figure 1.7: User interface of GlycoWorkbench 2.1 Stable (Build 157) running on macOS Sonoma. Users can graphically draw glycan structures and view peak lists and fragment information. By pasting the results processed by the mass spectrometer manufacturer's proprietary software onto GlycoWorkbench, users will be able to annotate the peak list information.

On the other hand, LC-MS/MS is mainly used in glycoproteomics. Glycoproteins are cleaved into glycopeptides using trypsin, chymotrypsin, and other proteases, and then separated by liquid chromatography (LC) and sent to a mass spectrometer. They will be ionized and MS/MS experiments will be performed. For the identification of glycopeptides, analysis software such as PMI-Byonic (Bern, Kil, and Becker, 2012) and MaxQuant (Tyanova, Temu, and Cox, 2016) are used.

In the case of PMI-Byonic, researchers need to enter the raw data of the peak lists obtained from the experiments, as well as the original glycoprotein sequence information (FASTA format) and a list of proteases used. Then, the glycopeptides are identified by matching each peak list against the database. The peaks that match the structures in the database are called peptide-spectrum matches (PSM), and are output as identification results. Unfortunately, since there are many isomers in the monosaccharide molecules that constitute glycans, this method in most cases provides only compositional information about the glycan structures bound to the glycopeptides. The identification results obtained in this way are now submitted together with the raw data from the mass spectrometry experiments to the data repository for proteomics under the ProteomeXchange Consortium (Deutsch et al., 2020), which will be described later.

1.4 Standard reporting guidelines

Sharing the experiment and analysis results of mass spectrometry and other methods among researchers around the world is extremely important for the purpose of confirming the reproducibility of the experiment and for knowledge discovery through large-scale data analysis using computers. In order to validate the reproducibility of an experiment, it is necessary to report not only the experimental data itself, but also the conditions of the experiment (e.g., sample preparation, type of reagents used, reaction time, etc.) and the software tools used for the analysis. Many standard reporting guidelines have been proposed to share such information in the life science field in accordance with the FAIR (Findability, Accessibility, Interoperability, and Re-usability) data principles (Wilkinson et al., 2016). Major guidelines include MIAME (Minimum Information About a Microarray Experiment) (Brazma et al., 2001) for microarray experiments, MIAPE (The Minimum Information About a Proteomics Experiment) (Taylor et al., 2007) for proteomics experiments, and STRENDA (Standards for Reporting Enzymology Data) (Tipton et al., 2014) for enzyme information.

For glycomics, a reporting guideline called MIRAGE (Minimum Information Required for A Glycomics Experiment) (York et al., 2014) has been proposed by the Beilstein-Institut as a reporting guideline for various glycan-related information, including the results of mass spectrometry experiments. As of December 15, 2023, the following eight reporting guidelines are available on the Internet (although some are still in manuscript in preparation).

- Sample Preparation (Struwe et al., 2016)
- Mass Spectrometric Analysis (Kolarich et al., 2013)
- Glycan Microarray Analysis (Liu et al., 2016)
- Liquid Chromatography Analysis (Campbell et al., 2019)
- NMR Glycan Recognition (Manuscript in preparation)
- NMR Glycan Structures (Manuscript in preparation)
- Capillary Electrophoresis Analysis (Lageveen-Kammeijer et al., 2022)
- Lectin Microarray Analysis (Manuscript in preparation)

These reporting guidelines specify the minimum experimental information to be reported when writing a publication based on the experimental results.

1.5 Data repositories in proteomics and glycomics

Each experiment and analysis result data with appropriate metadata such as experiment conditions and softwares used for the analysis in accordance with the standard reporting guidelines is uploaded to public data repositories on the Internet, where a unique identifier is assigned to the data. This allows researchers to be able to use the identifier to uniquely indicate the experiment results to readers around the world in their papers. In the field of glycoscience, there is also a data repository that assigns an identifier to the glycan structure itself, rather than to the experimental data. Here, I briefly introduce the data repositories and related tools currently used in proteomics and glycomics that are relevant to this study.

1.5.1 PRoteomics IDEntifications database (PRIDE) in the ProteomeXchange Consortium

The ProteomeXchange Consortium is an international pipeline for standardized data submission and sharing in proteomics and includes major data repositories in the field of proteomics. Currently, the consortium consists of six members: PRoteomics IDEntifications database (PRIDE) (Perez-Riverol et al., 2022), PeptideAtlas (Van Wijk et al., 2021), MassIVE (Choi et al., 2020), jPOST (Okuda et al., 2017), iProx (Ma et al., 2019), and Panorama Public (Sharma et al., 2018). Users can perform a cross search on ProteomeCentral (Jarnuczak and Vizcaíno, 2017) for information on all submissions to these data repositories.

PRIDE is one of the initial members of the consortium along with PeptideAtlas and is a data repository to which users can submit any type of proteomic dataset from a variety of experimental workflows. As of December 25, 2023, 25,237 submissions have been published in the PRIDE Archive. Identification results from mass spectrometry experiments obtained from glycoproteomics experiments are submitted in various formats such as MS-Excel files, CSV files, TSV files, etc., together with the raw data, to this data repository. Figure 1.8 shows the user interface of PRIDE displaying the submission assigned accession number PXD035445. Users can freely download the submitted experiment data from this screen.

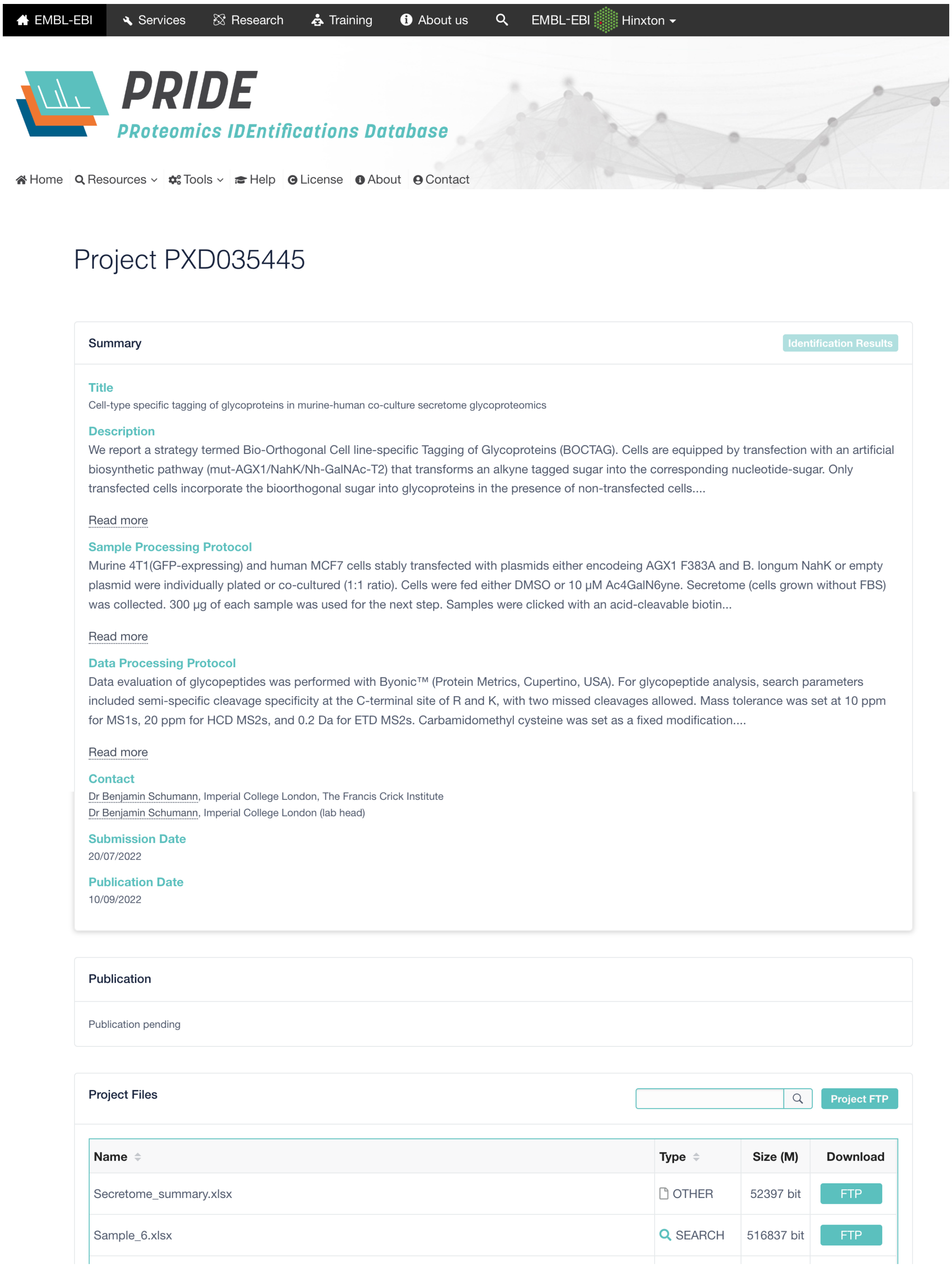
1.5.2 MS-Viewer in Protein Prospector

Protein Prospector (Darula et al., 2011) is a software and database package for mass spectrometry experiments in proteomics developed at the University of California, San Diego, and MS-Viewer (Baker and Chalkley, 2014) is the spectrum viewer included in it. Researchers can upload peak list files obtained from mass spectrometry experiments as well as results files containing annotations for the peak lists to MS-Viewer, allowing visualization of mass spectrometry results in web browsers. It supports various results file formats, such as Mascot (Perkins et al., 1999) CSV and MaxQuant, so files obtained from various types of mass spectrometry experiment analysis software can be visualized.

Since the results of mass spectrometry experiments uploaded to MS-Viewer are published on the Internet at the same time they are visualized, MS-Viewer also functions as a data repository for mass spectrometry result data. As of August 2, 2023, 124 submissions have been published in the MS-Viewer Repository Browser. Figure 1.9 shows the user interface of the MS-Viewer Repository Browser.

1.5.3 International glycan repository GlyTouCan

GlyTouCan is the international glycan repository that assigns a unique identifier, GlyTouCan ID, to each glycan structure. Experimental methods used to identify glycan structures, such as mass spectrometry experiments, do not always identify all the types of monosaccharides and linkage information between the monosaccharides in the glycan structures. Therefore, GlyTouCan must be able to deal with ambiguous glycan structures lacking some information. This functionality of GlyTouCan has been achieved based on the representation of each glycan structure as a string in the WURCS format.



Project PXD035445

Summary

Identification Results

Title
Cell-type specific tagging of glycoproteins in murine-human co-culture secretome glycoproteomics

Description
We report a strategy termed Bio-Orthogonal Cell line-specific Tagging of Glycoproteins (BOCTAG). Cells are equipped by transfection with an artificial biosynthetic pathway (mut-AGX1/NahK/Nh-GalNAc-T2) that transforms an alkyne tagged sugar into the corresponding nucleotide-sugar. Only transfected cells incorporate the bioorthogonal sugar into glycoproteins in the presence of non-transfected cells....

Read more

Sample Processing Protocol
Murine 4T1(GFP-expressing) and human MCF7 cells stably transfected with plasmids either encodeing AGX1 F383A and B. longum NahK or empty plasmid were individually plated or co-cultured (1:1 ratio). Cells were fed either DMSO or 10 µM Ac4GalN6yne. Secretome (cells grown without FBS) was collected. 300 µg of each sample was used for the next step. Samples were clicked with an acid-cleavable biotin...

Read more

Data Processing Protocol
Data evaluation of glycopeptides was performed with Byonic™ (Protein Metrics, Cupertino, USA). For glycopeptide analysis, search parameters included semi-specific cleavage specificity at the C-terminal site of R and K, with two missed cleavages allowed. Mass tolerance was set at 10 ppm for MS1s, 20 ppm for HCD MS2s, and 0.2 Da for ETD MS2s. Carbamidomethyl cysteine was set as a fixed modification....

Read more

Contact
Dr Benjamin Schumann, Imperial College London, The Francis Crick Institute
Dr Benjamin Schumann, Imperial College London (lab head)

Submission Date
20/07/2022

Publication Date
10/09/2022

Publication

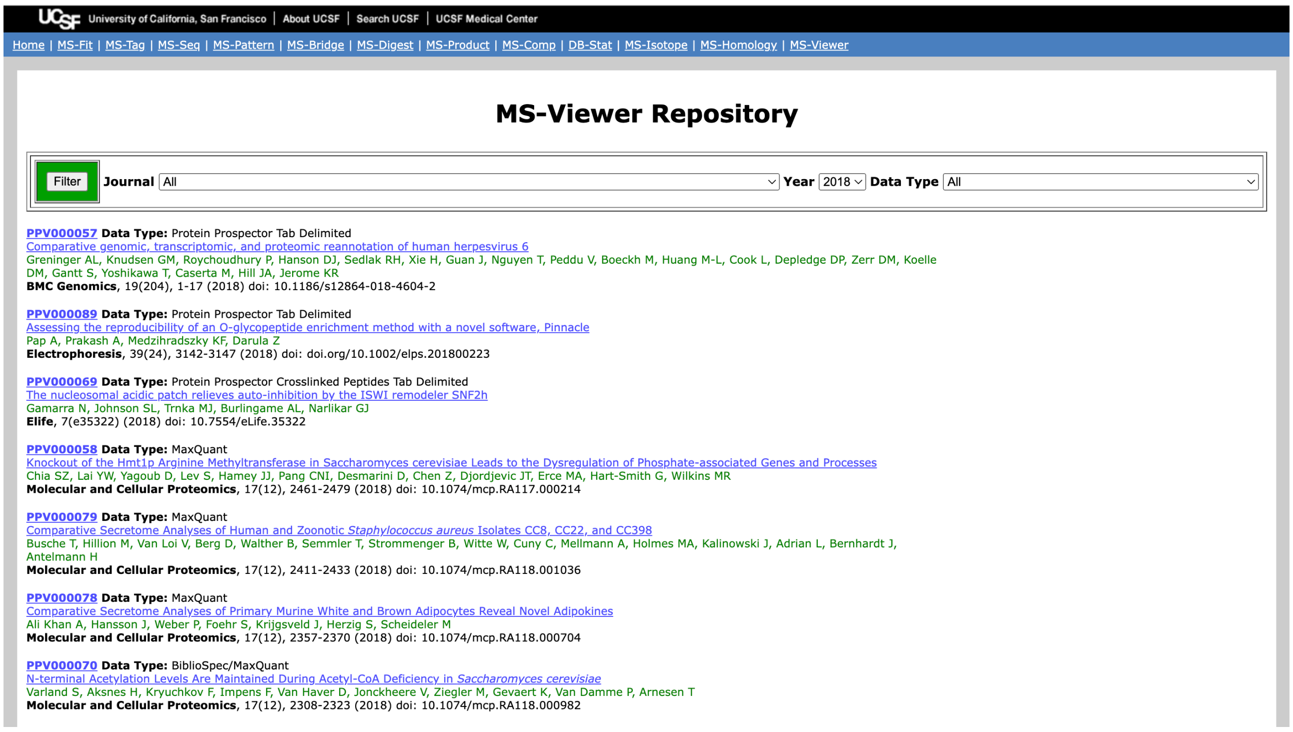
Publication pending

Project Files

Project FTP

Name	Type	Size (M)	Download
Secretome_summary.xlsx	OTHER	52397 bit	FTP
Sample_6.xlsx	SEARCH	516837 bit	FTP

Figure 1.8: User interface of PRIDE data repository as of October 6, 2023 displaying the submission assigned accession number PXD035445. Users can download the submitted files from this screen via the FTP server.


University of California, San Francisco | About UCSF | Search UCSF | UCSF Medical Center
Home | MS-Fit | MS-Tag | MS-Seq | MS-Pattern | MS-Bridge | MS-Digest | MS-Product | MS-Comp | DB-Stat | MS-Isotope | MS-Homology | MS-Viewer
Filter

MS-Viewer Repository

Journal All **Year** 2018 **Data Type** All

PPV000057 **Data Type:** Protein Prospector Tab Delimited
Comparative genomic, transcriptomic, and proteomic reannotation of human herpesvirus 6
Greninger AL, Knudsen GM, Roychoudhury P, Hanson DJ, Sedlak RH, Xie H, Guan J, Nguyen T, Peddu V, Boeckh M, Huang M-L, Cook L, Depledge DP, Zerr DM, Koelle DM, Gantt S, Yoshikawa T, Caserta M, Hill JA, Jerome KR
BMC Genomics, 19(204), 1-17 (2018) doi: 10.1186/s12864-018-4604-2

PPV000089 **Data Type:** Protein Prospector Tab Delimited
Assessing the reproducibility of an O-glycopeptide enrichment method with a novel software, Pinnacle
Pap A, Prakash A, Medzihradszky KF, Darula Z
Electrophoresis, 39(24), 3142-3147 (2018) doi: doi.org/10.1002/elps.201800223

PPV000069 **Data Type:** Protein Prospector Crosslinked Peptides Tab Delimited
The nucleosomal acidic patch relieves auto-inhibition by the ISWI remodeler SNF2h
Gamarra N, Johnson SL, Trnka MJ, Burlingame AL, Narlikar GJ
Elife, 7(e35322) (2018) doi: 10.7554/eLife.35322

PPV000058 **Data Type:** MaxQuant
Knockout of the Hmt1p Arginine Methyltransferase in Saccharomyces cerevisiae Leads to the Dysregulation of Phosphate-associated Genes and Processes
Chia SZ, Lai YW, Yagoub D, Lev S, Hamey JJ, Pang CNI, Desmarini D, Chen Z, Djordjevic JT, Erce MA, Hart-Smith G, Wilkins MR
Molecular and Cellular Proteomics, 17(12), 2461-2479 (2018) doi: 10.1074/mcp.RA117.000214

PPV000079 **Data Type:** MaxQuant
Comparative Secretome Analyses of Human and Zoonotic *Staphylococcus aureus* Isolates CC8, CC22, and CC398
Busche T, Hillion M, Van Loi V, Berg D, Walther B, Semmler T, Strommenger B, Witte W, Cuny C, Mellmann A, Holmes MA, Kalinowski J, Adrian L, Bernhardt J, Antelmann H
Molecular and Cellular Proteomics, 17(12), 2411-2433 (2018) doi: 10.1074/mcp.RA118.001036

PPV000078 **Data Type:** MaxQuant
Comparative Secretome Analyses of Primary Murine White and Brown Adipocytes Reveal Novel Adipokines
Ali Khan A, Hansson J, Weber P, Foehr S, Krijgsveld J, Herzig S, Scheideler M
Molecular and Cellular Proteomics, 17(12), 2357-2370 (2018) doi: 10.1074/mcp.RA118.000704

PPV000070 **Data Type:** BiblioSpec/MaxQuant
N-terminal Acetylation Levels Are Maintained During Acetyl-CoA Deficiency in *Saccharomyces cerevisiae*
Varland S, Aksnes H, Kryuchkov F, Impens F, Van Haver D, Jonckheere V, Ziegler M, Gevaert K, Van Damme P, Arnesen T
Molecular and Cellular Proteomics, 17(12), 2308-2323 (2018) doi: 10.1074/mcp.RA118.000982

FIGURE 1.9: User interface of MS-Viewer Repository Browser as of October 4, 2023. Users can visualize and browse the archive of annotated mass spectrometry experiment results on web browsers.

The WURCS glycan structure notation can represent not only glycans containing ambiguous monosaccharides and glycosidic linkages, but also glycans containing fragmented substructures and structures containing only monosaccharide composition information. This means that all glycan structures that can be obtained by mass spectrometry experiments can be represented in WURCS format, which can thus be assigned a unique identifier, and thus GlyTouCan plays an important role in glycomics and glycoproteomics. Therefore, it has been adopted as the recommended repository for glycan data and is used to integrate glycan data in the GlySpace Alliance (Aoki-Kinoshita et al., 2020), which includes GlyCosmos, GlyGen (York et al., 2020), and Glycomics@ExPASy (Mariethoz et al., 2018).

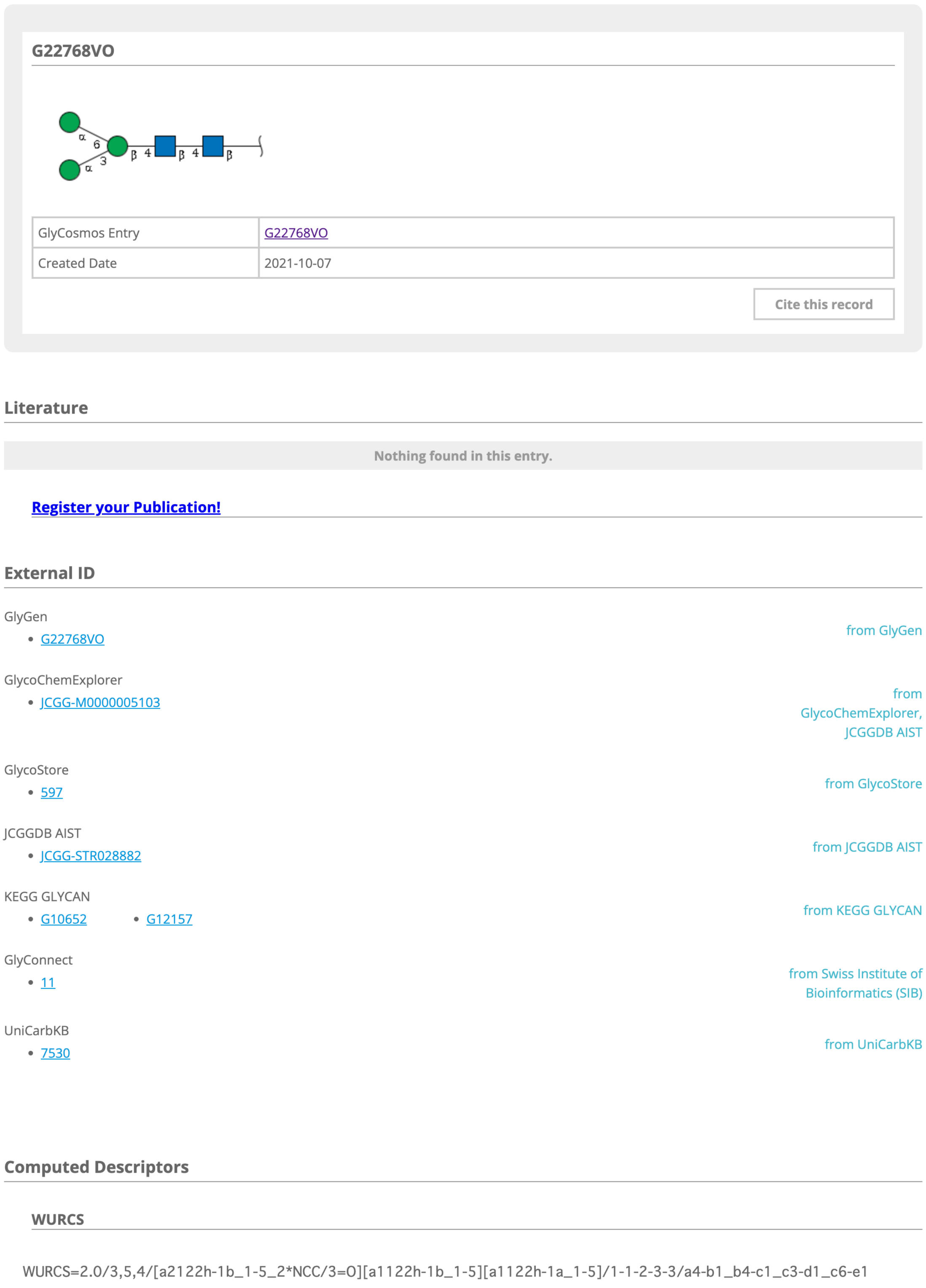
In addition, GlyTouCan is built on top of Semantic Web technologies. This allows users to write queries using a query language called SPARQL (Hogan, 2020) to perform cross-database queries against a wide variety of information stored in many databases on the Internet, and to obtain various information related to the target glycan structure all at once. As of December 25, 2023, 223,014 glycan structures are registered in GlyTouCan. Figure 1.10 shows the core structure of the *N*-linked glycan identified by GlyTouCan ID G22768VO on GlyTouCan.

1.5.4 Data repositories in glycomics

In glycomics, two data repositories, GlycoPOST (Watanabe et al., 2021) and UniCarb-DR (Rojas-Macias et al., 2019), are publicly available under the GlyCosmos (Yamada et al., 2020) project. Although both of these data repositories are based on the MIRAGE guidelines for mass spectrometry experimental data in glycomics, each has its own unique advantages. This section describes the characteristics of these data repositories.

GlycoPOST

GlycoPOST is a data repository where various types of experimental data, including raw data obtained from mass spectrometers, can be submitted following MIRAGE sample preparation guidelines and mass spectrometric analysis guidelines. GlycoPOST provides a unique MIRAGE-related information storing unit called “presets” to facilitate the registration of MIRAGE guideline information. Since submitters can reuse their own existing presets, they can greatly save time when submitting data obtained from experimental conditions that are similar to those of previously submitted data. GlycoPOST is designed for highly efficient data upload. The front-end application running on the web browser and the back-end application running on the server cooperate with each other to divide a large file into small data chunks of about 1 MB, which are sent in parallel at high speed. Each data



G22768VO

GlyCosmos Entry	G22768VO
Created Date	2021-10-07

Cite this record

Literature

Nothing found in this entry.

Register your Publication!

External ID

GlyGen
- G22768VO

from GlyGen

GlycoChemExplorer
- JCGG-M0000005103

from GlycoChemExplorer, JCGGDB AIST

GlycoStore
- 597

from GlycoStore

JCGGDB AIST
- JCGG-STR028882

from JCGGDB AIST

KEGG GLYCAN
- G10652
- G12157

from KEGG GLYCAN

GlyConnect
- 11

from Swiss Institute of Bioinformatics (SIB)

UniCarbKB
- 7530

from UniCarbKB

Computed Descriptors

WURCS

WURCS=2.0/3,5,4/[a2122h-1b_1-5_2*NCC/3=O][a1122h-1b_1-5][a1122h-1a_1-5]/1-1-2-3-3/a4-b1_b4-c1_c3-d1_c6-e1

FIGURE 1.10: Display screen of each glycan structure entry in GlyTouCan as of October 6, 2023. Here, the details of the core structure of *N*-linked glycans assigned the GlyTouCan ID G22768VO are displayed. In addition to the string representation of the glycan structure described in WURCS format, links to various related external resources are also available.

chunk sent to the server side is merged to restore the original file on the server. This enables the transmission of huge files in a very short period of time.

In addition, GlycoPOST allows submitters to freely set the publication date of their submitted data. Thus, during the peer review period (embargo period) of a paper, they can make their submitted data available only to reviewers, and after the paper is published, they can switch the submitted data to general public access. However, due to the nature of the various types of data it can accept, GlycoPOST does not implement any functionality to analyze or visualize the contents of individual submitted files.

As of December 25, 2023, 293 submissions have been registered, of which 218 are publicly available. Figure 1.11 shows the submission list screen of GlycoPOST.

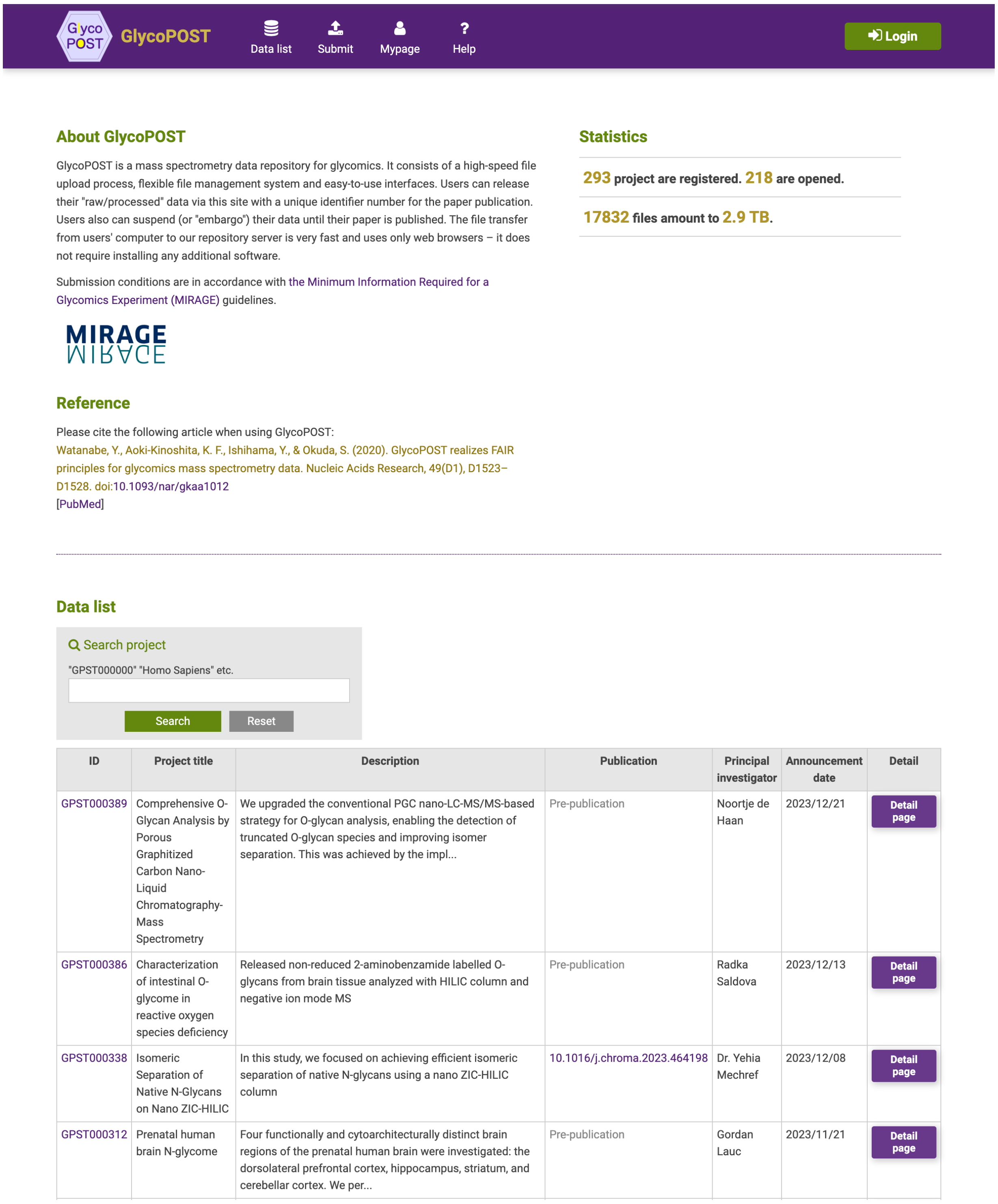
UniCarb-DR

UniCarb-DR, similar to GlycoPOST, is a data repository for submitting the results of mass spectrometry experiments following the sample preparation and mass spectrometric analysis guidelines of MIRAGE. However, while GlycoPOST can accept arbitrary data including raw data obtained from mass spectrometers, UniCarb-DR can only accept mass spectrometry experiment results annotated using GlycoWorkbench in GWP format. By virtue of the fact that it only accepts data in a certain format, UniCarb-DR provides rich visualization functionalities for the results of mass spectrometry experiments. It also provides flexible search functions for submitted mass spectrometry experiment results using metadata such as species information, monosaccharide composition information, etc. Unlike GlycoPOST, submitted data are published immediately, and submitters are not allowed to specify the date of publication of their data in advance.

As of December 25, 2023, 74 submissions have been registered, including 1,710 results of glycan structure identification. Figure 1.12 shows the identification result of a glycan structure contained in one submission registered in UniCarb-DR with spectral data.

1.6 Purpose

With the development of multiple data repositories such as GlyTouCan, GlycoPOST, and UniCarb-DR, the infrastructure for the accumulation and sharing of data in glycoscience research is now being established. However, unlike proteins, lipids, and other biomolecules, glycans do not function alone in organisms but rather serve as modulators of substrate functions by attaching to the substrates.



About GlycoPOST

GlycoPOST is a mass spectrometry data repository for glycomics. It consists of a high-speed file upload process, flexible file management system and easy-to-use interfaces. Users can release their "raw/processed" data via this site with a unique identifier number for the paper publication. Users also can suspend (or "embargo") their data until their paper is published. The file transfer from users' computer to our repository server is very fast and uses only web browsers – it does not require installing any additional software.

Submission conditions are in accordance with the Minimum Information Required for a Glycomics Experiment (MIRAGE) guidelines.

Statistics

293 project are registered. 218 are opened.

17832 files amount to 2.9 TB.

Reference

Please cite the following article when using GlycoPOST:
Watanabe, Y., Aoki-Kinoshita, K. F., Ishihama, Y., & Okuda, S. (2020). GlycoPOST realizes FAIR principles for glycomics mass spectrometry data. Nucleic Acids Research, 49(D1), D1523–D1528. doi:10.1093/nar/gkaa1012
[PubMed]

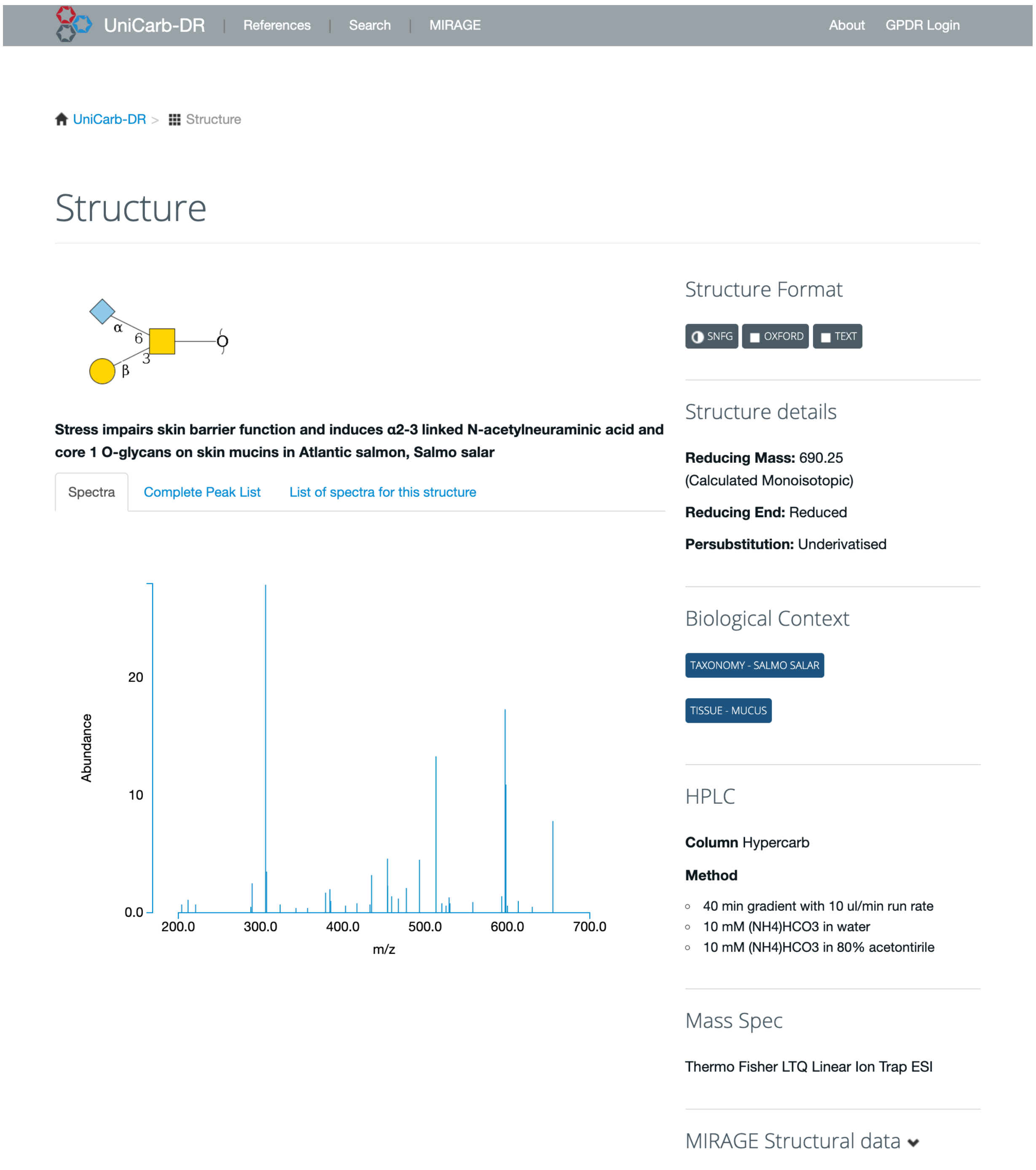
Data list

Search project
"GPST000000" "Homo Sapiens" etc.

Search Reset

ID	Project title	Description	Publication	Principal investigator	Announcement date	Detail
GPST000389	Comprehensive O-Glycan Analysis by Porous Graphitized Carbon Nano-Liquid Chromatography-Mass Spectrometry	We upgraded the conventional PGC nano-LC-MS/MS-based strategy for O-glycan analysis, enabling the detection of truncated O-glycan species and improving isomer separation. This was achieved by the impl...	Pre-publication	Noortje de Haan	2023/12/21	Detail page
GPST000386	Characterization of intestinal O-glycome in reactive oxygen species deficiency	Released non-reduced 2-aminobenzamide labelled O-glycans from brain tissue analyzed with HILIC column and negative ion mode MS	Pre-publication	Radka Saldova	2023/12/13	Detail page
GPST000338	Isomeric Separation of Native N-Glycans on Nano ZIC-HILIC	In this study, we focused on achieving efficient isomeric separation of native N-glycans using a nano ZIC-HILIC column	10.1016/j.chroma.2023.464198	Dr. Yehia Mechref	2023/12/08	Detail page
GPST000312	Prenatal human brain N-glycome	Four functionally and cytoarchitecturally distinct brain regions of the prenatal human brain were investigated: the dorsolateral prefrontal cortex, hippocampus, striatum, and cerebellar cortex. We per...	Pre-publication	Gordan Lauc	2023/11/21	Detail page

Figure 1.11: Submission list screen on GlycoPOST as of December 26, 2023 showing submissions that have been published after the embargo period. Submissions for which publication information was not entered at the time of registration are marked as "Pre-publication", and the submitter can set a PubMed ID or a DOI as publication information even after the submission has been published. Users can search for specific submissions by entering search words such as ID or species name.



Structure

Stress impairs skin barrier function and induces α2-3 linked N-acetylneuraminic acid and core 1 O-glycans on skin mucins in Atlantic salmon, Salmo salar

Structure Format

Structure details

Reducing Mass: 690.25 (Calculated Monoisotopic)

Reducing End: Reduced

Persubstitution: Underivatised

Biological Context

TAXONOMY - SALMO SALAR

TISSUE - MUCUS

HPLC

Column Hypercarb

Method

- 40 min gradient with 10 ul/min run rate
- 10 mM (NH4)HCO3 in water
- 10 mM (NH4)HCO3 in 80% acetontirile

Mass Spec

Thermo Fisher LTQ Linear Ion Trap ESI

MIRAGE Structural data

FIGURE 1.12: Display screen of a glycan structure identification result registered in UniCarb-DR as of October 4, 2023. The glycan structure identified as a result of the experiment is displayed as an image, as well as a chart for spectral data and metadata entered following the MIRAGE guidelines.

Therefore, to advance our understanding of the functions and roles of glycans, it is necessary to accumulate information not only on glycans but also on the substrates to which they attach, such as proteins and lipids.

For example, glycoproteomics is an important field for understanding how proteins are regulated by glycans. With current technologies, glycoproteins are identified using various mass spectrometry techniques combined with orthogonal methods, but in the end, the data usually consist of identified peptides and glycosylation sites annotated with one or more monosaccharide compositions (hereafter referred to as glycan structures). In order to integrate this data with current protein-centric and glycan-centric databases, it is crucial to have a standardized system to identify glycoconjugates, and in particular glycoproteins and glycopeptides. Such a public data repository for storing and sharing glycoconjugate data, including glycopeptides and glycoproteins, was lacking, thus hindering the efficient accumulation of glycoconjugate knowledge. It is true that the UniProtKB database (The UniProt Consortium, 2023), for example, contains glycosylation information for each protein entry. However, because UniProtKB assigns an identifier only at the protein level, while each glycosylation site may have a publication reference, it is currently not possible to search for a specific set of glycosylation patterns on a particular peptide sequence, which may have a particular relevance to a disease state. Glycopeptide and glycoprotein profiles identified in glycoproteomics experiments reported in the literature are often compiled into MS Excel or CSV files and either provided as Supplementary Data or uploaded together with raw data of liquid chromatography-tandem mass spectrometry (LC-MS/MS) experiments to a proteomics data repository participating in the ProteomeXchange Consortium such as PRIDE. However, since there is no standardized format for these files, each researcher submits their own data in their preferred format. This means that in order to collect and analyze experimental results containing specific glycopeptides or glycoproteins from such data repositories, these uniquely formatted files must be opened and processed individually, which is a time-consuming process.

To overcome these issues, a new data repository for depositing glycoconjugate data to accelerate glycoscience research was needed, and thus we developed GlyComb, a new data repository for such data. The data deposited in GlyComb will enable the unique identification of glycopeptides and glycoproteins contained in glycoproteomics experiment results submitted to GlycoPOST, and to link the glycan data with GlyTouCan identifiers. Currently, GlyComb can assign a unique identifier to a set of glycosylation information associated with a specific peptide sequence or UniProt ID, which can be registered and published as a glycopeptide or glycoprotein entry, respectively. GlyComb is now publicly available on the Internet at `https://glycomb.glycosmos.org`, and users can access this new web resource using a web browser. We have aimed to make GlyComb user-friendly, providing tools that make it easy to upload glycopeptide and glycoprotein information to register.

In addition, although both GlycoPOST and UniCarb-DR are MIRAGE-compiliant data repositories for mass spectrometry experimental data in glycomics, they have different characteristics. While GlycoPOST is more oriented toward hosting large raw data from analyzers, UniCarb-DR focuses on collecting and visualizing the results of annotated mass spectrometry experiments. Due to the fact that each has been developed and maintained separately, these are completely separate data repositories, and there were no cross-links at all between the submissions in each repository. However, from the data submitter's point of view, it is obviously inconvenient to have to separately submit raw data from mass spectrometry experiments and their identification result to separate data repositories. In addition, while GlycoPOST allows the submitters to freely set the publication date of the submitted data, UniCarb-DR immediately announces the submitted data to the public. This makes it difficult for users to use UniCarb-DR because they have to submit and publish their identification result data to be referenced in the paper under review to UniCarb-DR in advance. Previously, several attempts have been made to promote collaboration between the two data repositories to some extent, such as the introduction of a common login system between them. However, for example, in proteomics, ProteomeXchange provides extensive cross-linking among different data repositories, allowing users to cross-search between multiple data repositories from the same ProteomeCentral website, and the data repositories are being linked so that they operate as an integrated whole system. In order to take advantage of these two complementary glycomics data repositories more than ever before, more enhanced integration between them was needed. Therefore, we investigated the possibilities of further collaboration between GlycoPOST and UniCarb-DR in order to improve user convenience, and implemented some new features. As a result, a unified data submission flow to these two data repositories was implemented to achieve cross-referencing between the respective data of GlycoPOST and UniCarb-DR. In addition, by obtaining GlyTouCan IDs from the glycan structures described in the GWS format, the linkage between these two data repositories and GlyTouCan was strengthened.

Such data repositories are inherently expected to continue to run over the long term. Therefore, when new analysis software becomes mainstream or new versions of existing software are released in the future, it would be desirable to be able to submit the products of such software to these data repositories and extract data from them without any problems. To meet this requirement, these data repositories must be able to safely add or extend functionality without destroying existing functionalities. However, this is a very difficult requirement because it is impossible to expect all possible future changes in advance when developing a data repository. In fact, the Kinoshita lab took over the development of UniCarb-DR, which was originally developed at Gothenburg University in Sweden, however, the programs at that time were not designed for specification changes such as URL changes or login system changes, and various functions such as search functions and dataset submission functions failed to work. It was also difficult to identify the locations of programs where such problems occurred and the causes of them.

Thus, to ensure the long-term stable running of these data repositories and to enable them to meet the requirements of adding and extending functionality with relative ease, in this study I adopted several statically typed programming languages with expressive type systems (Pierce, 2002) as the technology for developing these data repositories, and applied techniques and idioms for safe software development with functional programming (Hudak, 1989) that have been used and studied in computer science for many years. This enables us to develop data repositories more reliably, and also to support the future development of new functions such as the linkage of these data repositories without destroying the existing functions.

Chapter 2

Methods

2.1 Infrastructure for developing data repositories

In this section, I describe the fundamental technologies used in this study to develop the data repositories.

2.1.1 Relational databases and querying with SQL

Database management systems (DBMSs) are used to store data in a wide variety of systems around the world, from in-house user management systems to large e-commerce sites. While many different types of DBMS software have been developed, including document stores such as MongoDB (Bradshaw, Brazil, and Chodorow, 2019) and key-value stores including Redis (Carlson, 2013) and Memcached (Fitzpatrick, 2004), the most commonly used type of DBMS as of 2023 is RDBMS (relational database management system). Figure 2.1 shows (a) the breakdown of categories of different 417 DBMS softwares by their model and (b) the popularity of each category listed in the DB-Engines website, a knowledge base of relational and NoSQL database management systems (`https://db-engines.com/en/ranking_categories`). From Figure 2.1 (b), it is evident that RDBMSs are the most popular of the many categories of database management softwares.

A relational database is a database based on a relational model of data, invented by Edgar F. Codd (Codd, 1970) in the 1970s. In the relational model, data are organized into one or more tables. Each column in a table represents an attribute of the data to be stored. On the other hand, each row is called a record or a tuple, and represents one data entity which is an N-tuple having N attributes. Entities stored in each table refer to each other using their own identifiers as foreign keys, thereby eliminating duplication of information among entities and enabling efficient data storage. Figure 2.2 shows an example of the tables in the relational model.

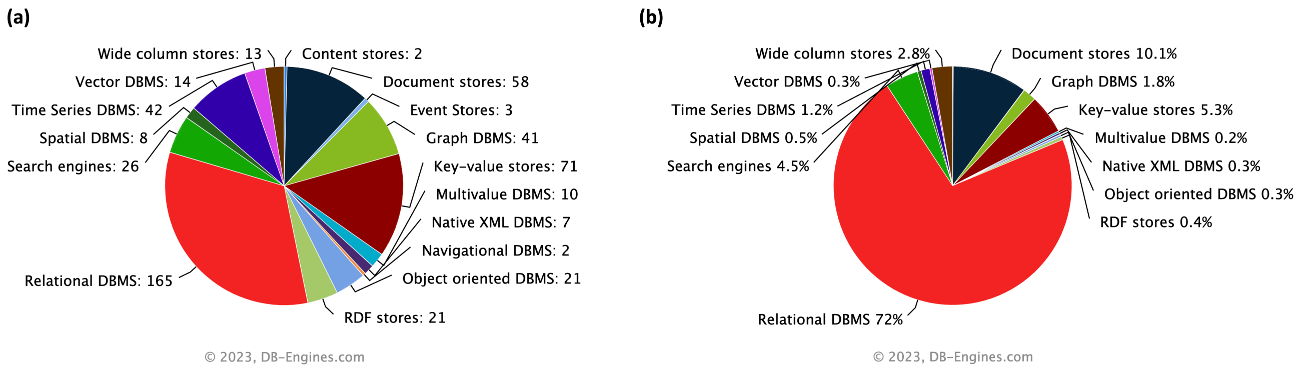
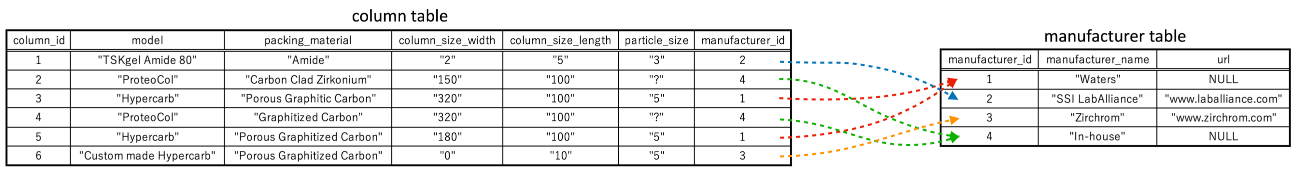


FIGURE 2.1: Breakdown of database management systems (DBMSs) by database model as of December 2023. (a) A breakdown of the 417 different database management systems published on DB-Engines, categorized by database model (e.g. RDBMS, key-value stores, etc.). Note that some systems belong to more than one category. (b) A breakdown of the popularity of each database model, calculated based on DB-Engines' own ranking scores. The sum of all scores is normalized to 100%. These scores are calculated considering the number of websites that mention the system, trending information from search engines such as Google Trends, and the frequency of technical discussions about the systems on sites such as Stack Overflow.


column table

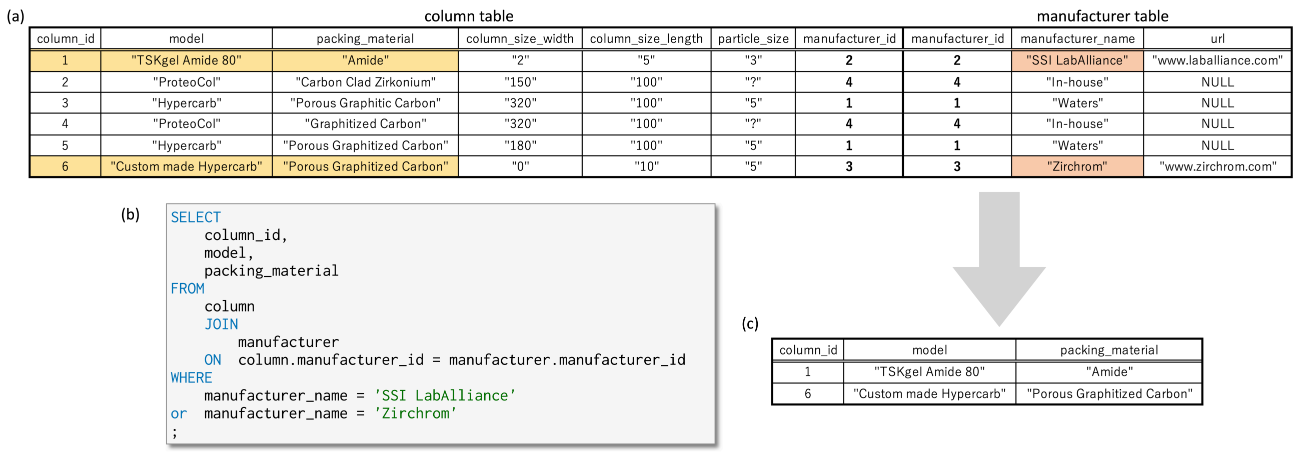
column_id	model	packing_material	column_size_width	column_size_length	particle_size	manufacturer_id
1	"TSKgel Amide 80"	"Amide"	"2"	"5"	"3"	2
2	"ProteoCol"	"Carbon Clad Zirkonium"	"150"	"100"	"?"	4
3	"Hypercarb"	"Porous Graphitic Carbon"	"320"	"100"	"5"	1
4	"ProteoCol"	"Graphitized Carbon"	"320"	"100"	"?"	4
5	"Hypercarb"	"Porous Graphitized Carbon"	"180"	"100"	"5"	1
6	"Custom made Hypercarb"	"Porous Graphitized Carbon"	"0"	"10"	"5"	3

manufacturer table

manufacturer_id	manufacturer_name	url
1	"Waters"	NULL
2	"SSI LabAlliance"	"www.laballiance.com"
3	"Zirchrom"	"www.zirchrom.com"
4	"In-house"	NULL

FIGURE 2.2: Example of the tables in the relational model. Here, the column table, which stores detailed information on columns used in HPLC experiments, and the manufacturer table, which stores the manufacturer's information on the columns, are shown. These are real examples of tables used in UniCarb-DR as of January 19, 2024. By referencing the manufacturer table from the column table using the manufacturer_id attribute, data can be stored without duplicate information in the database.

A query language called SQL is used to perform queries such as retrieving, inserting, deleting, and joining data in a relational database. Using SQL, it is possible to combine multiple tables in a database and retrieve entities that match specified conditions in a concise way. Figure 2.3 shows an example of SQL query to retrieve information from the tables shown in Figure 2.2.


(a)

column table							manufacturer table		
column_id	model	packing_material	column_size_width	column_size_length	particle_size	manufacturer_id	manufacturer_id	manufacturer_name	url
1	"TSKgel Amide 80"	"Amide"	"2"	"5"	"3"	**2**	**2**	"SSI LabAlliance"	"www.laballiance.com"
2	"ProteoCol"	"Carbon Clad Zirkonium"	"150"	"100"	"?"	**4**	**4**	"In-house"	NULL
3	"Hypercarb"	"Porous Graphitic Carbon"	"320"	"100"	"5"	**1**	**1**	"Waters"	NULL
4	"ProteoCol"	"Graphitized Carbon"	"320"	"100"	"?"	**4**	**4**	"In-house"	NULL
5	"Hypercarb"	"Porous Graphitized Carbon"	"180"	"100"	"5"	**1**	**1**	"Waters"	NULL
6	"Custom made Hypercarb"	"Porous Graphitized Carbon"	"0"	"10"	"5"	**3**	**3**	"Zirchrom"	"www.zirchrom.com"

(b)

```
SELECT
    column_id,
    model,
    packing_material
FROM
    column
    JOIN
        manufacturer
    ON  column.manufacturer_id = manufacturer.manufacturer_id
WHERE
    manufacturer_name = 'SSI LabAlliance'
or  manufacturer_name = 'Zirchrom'
;
```

(c)

column_id	model	packing_material
1	"TSKgel Amide 80"	"Amide"
6	"Custom made Hypercarb"	"Porous Graphitized Carbon"

FIGURE 2.3: Example of SQL query to retrieve information. (a) This table is constructed by joining the two tables shown in Figure 2.2, using the manufacturer_id of each table as a key. (b) This is a SQL query to retrieve the column_id, model, and packing_material attributes of the record from the derived table shown in (a) whose manufacturer's name is "SSI LabAlliance" or "Zirchrom". The data retrieved by this query corresponds to the cells highlighted in yellow in (a). (c) This is the result table containing the data retrieved by the query shown in (b).

MySQL(DuBois, 2013), MariaDB (Kenler and Razzoli, 2015), and PostgreSQL (Douglas and Douglas, 2003) are the most popular open-source RDBMSs that can be used free of charge. Each of the data repositories developed in this study also uses a relational DBMS for storing user information and submission data. Concretely, MariaDB is used for GlyComb, MySQL for GlycoPOST, and PostgreSQL for UniCarb-DR. While RDBMSs are generally high performance, it has been difficult to integrate data from many databases and data repositories developed using RDBMSs in life science research. To overcome this problem, many efforts have been made in recent years to link and integrate databases and data repositories using Semantic Web technologies, as described later. In this study, GlyComb registers data once registered in a relational database to a triplestore, a database that is compatible with Semantic Web technologies, after performing data validation during batch processing. This allows GlyComb to be linked to a various data resources, including GlyTouCan and others.

2.1.2 Three-tier client-server architecture

The data repositories developed in this study are all in the form of web applications that can be accessed and used using web browsers. Therefore, the structure of these data repositories all follow the three-tier client-server architecture. In this architecture, the entire web application is divided into three tiers: presentation tier, logic tier, and data tier. The roles of each tier are as follows:

- **Presentation tier**: This tier contains the user interface to interact with the user. The program in this tier runs on the web browser of the user's computer.
- **Logic tier**: This tier contains the application logic that runs on the server. It communicates with the database in the data tier in response to user requests sent from the presentation tier and returns responses to the presentation tier.
- **Data tier**: This tier contains the relational database managed by the RDBMS. It extracts, inserts, updates, and deletes data in the database according to SQL queries issued by the logic tier.

By dividing the application into multiple tiers in this way, it is easier to maintain and modify the entire application. Figure 2.4 shows a conceptual diagram of this architecture as a whole and the typical process at each tier when updating the screen at the presentation tier.

2.1.3 Containerized virtualization technology

One of the challenges in software development is that it is difficult to ensure the reproducibility of the environment in which the software works. Software often runs well on the computer used in its development, however, when it is attempted to run in a production environment on a server, it fails to run properly due to some errors. This can be caused by various reasons, such as the hardware or operating system of the computer used during the development being different from that used in the production environment. As a means of minimizing such problems, a technology called containerized virtualization, as typified by Docker (Merkel, 2014), has been gaining attention in recent years. Containerized virtualization technology allows developed software to run in a lightweight virtual environment called a "container". Within a container, software developers can explicitly specify the dependent libraries and their versions, which greatly improves the reproducibility of the software's working environment. Furthermore, once created, container environments can be shared and reused as "container images" across multiple computers, making it easy to reproduce the exact same environments on production

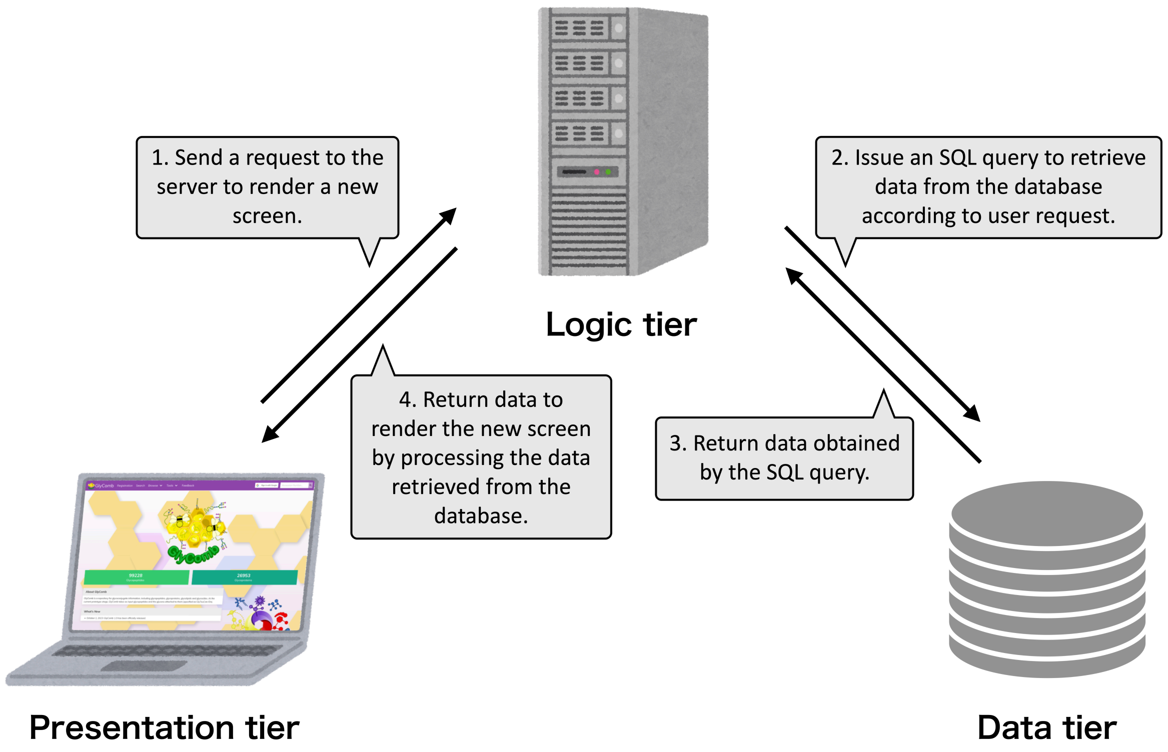

1. Send a request to the server to render a new screen.
2. Issue an SQL query to retrieve data from the database according to user request.
Logic tier
4. Return data to render the new screen by processing the data retrieved from the database.
3. Return data obtained by the SQL query.
Presentation tier
Data tier

FIGURE 2.4: Conceptual diagram of a three-tier client-server architecture and a typical process flow during a screen update of the user interface. First, a screen update request is sent from the presentation tier to the logic tier in response to the user's operation on the web browser (1). Then, the application in the logic tier running on the server issues an SQL query to obtain the data needed to render the new screen from the database in the data tier (2). The database returns the data retrieved according to that SQL query to the logic tier (3), which transforms that data and returns new screen data for the presentation tier (4).

servers. Each of the data repositories developed in this study is built as multiple container-based applications using Docker and docker-compose, a tool for managing multiple containers together. This allows these data repositories to run stably on a wide range of computers capable of running Docker, making it easier to keep them running over the long term. Figure 2.5 shows a conceptual diagram illustrating an example of a system based on multiple containers.

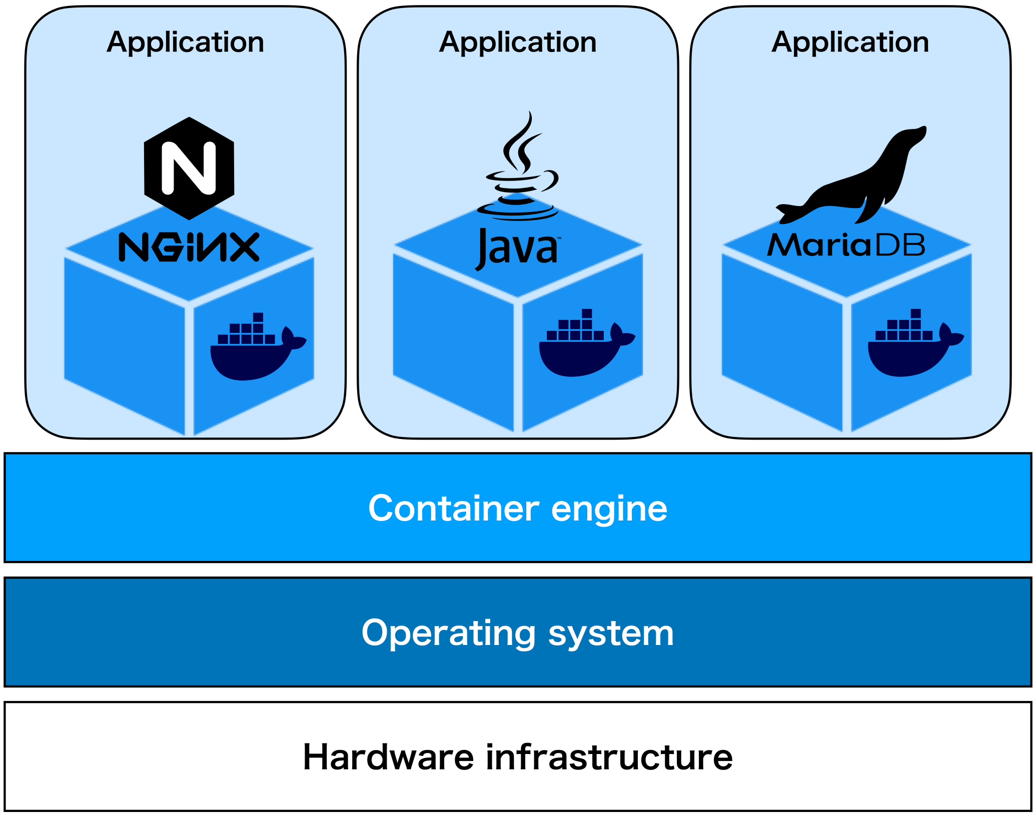


FIGURE 2.5: Example of a system based on multiple containers. By using containerized virtualization technology as typified by Docker, for example, Nginx (Reese, 2008), a typical web server used to serve static files, Java (Bloch, 2008), a programming language very popular for developing server-side applications running on the logic tier, and MariaDB, an RDBMS running on the data tier, can run in independent and separate containers. Since each container is executed by a container engine running on the host computer's operating system, rather than directly on the operating system, the influence of the operating system can be minimized, ensuring highly reproducible environments on a variety of computers. For example, GlyComb, the data repository developed in this study, is also constructed by collaborating Nginx, Java, and MariaDB running in independent containers, as shown in this figure.

2.1.4 Semantic Web technologies

The Semantic Web, proposed by Tim Berners-Lee, is an extension of the existing World Wide Web to make all data on the Internet machine-readable (Berners-Lee, Hendler, and Lassila, 2023). The conventional World Wide Web is oriented toward human readability, assuming that people read documents directly on the Internet, and thus forms a “web of documents” in which documents are interconnected using hyperlinks among documents. In contrast, the Semantic Web aims to create a “web of data” in which all data on the Internet are interlinked.

In order to establish the Semantic Web, it is necessary to have a framework for representing all kinds of data on the Internet and for defining semantics and concepts for them. By defining semantics for the data, computers will be able to infer information from links among the data according to certain rules. The Semantic Web adopts Resource Description Framework (RDF) (Manola, Miller, and McBride, 2004) as a technology to provide this framework.

RDF represents all information in the “subject-predicate-object” form, also known as a triple. The subject, predicate, and object are all described using uniform resource identifiers (URIs). However, only object may be a literal value representing a concrete string, number, or other value. If a triple is considered as a graph structure, the subject and object correspond to two nodes respectively, and the predicate corresponds to an edge. Thus, by accumulating multiple triples, they can be combined to obtain a single directed graph structure. Such a graph structure is called a knowledge graph. Figure 2.6 shows (a) an example triple and (b) a part of the graph structure about glycan structures using RDF that is actually used in GlyTouCan.

There are several formats for describing triple-based information in RDF. Among them, the Turtle format (Beckett et al., 2014) is the most commonly used in the field of bioinformatics because of its concise syntax and convenience. Figure 2.7 describes triples in the graph structure shown in Figure 2.6 (b) in Turtle format. The `prefix` declarations that appear at the beginning of the file define a set of local prefixes to describe long URI in a shortened form. Using these local prefixes, for example, the URI `http://purl.jp/bio/12/glyco/glycan#Saccharide` can be shortened to `glycan:Saccharide`, resulting in a better readability of a set of triple definitions.

The vocabulary used to assign meanings to entities in RDF is called an ontology. Various ontologies have already been developed to express various meanings in the Semantic Web. In the field of glycoinformatics, GlycoRDF (Ranzinger et al., 2015) for describing glycan-related information and

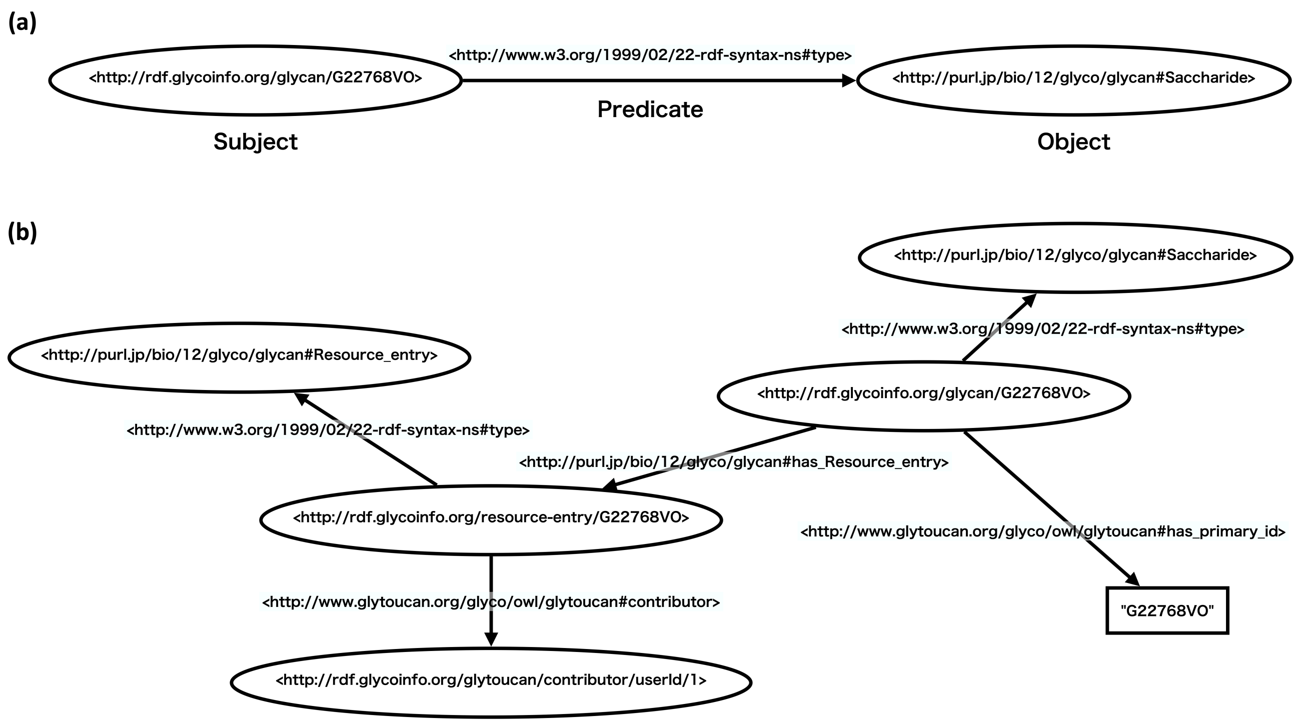

(a)
<http://rdf.glycoinfo.org/glycan/G22768VO>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
Predicate
<http://purl.jp/bio/12/glyco/glycan#Saccharide>
Subject
Object
(b)
<http://purl.jp/bio/12/glyco/glycan#Saccharide>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.jp/bio/12/glyco/glycan#Resource_entry>
<http://rdf.glycoinfo.org/glycan/G22768VO>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.jp/bio/12/glyco/glycan#has_Resource_entry>
<http://rdf.glycoinfo.org/resource-entry/G22768VO>
<http://www.glytoucan.org/glyco/owl/glytoucan#has_primary_id>
<http://www.glytoucan.org/glyco/owl/glytoucan#contributor>
"G22768VO"
<http://rdf.glycoinfo.org/glytoucan/contributor/userId/1>

FIGURE 2.6: Examples of triple and graph structure composed of a set of triples expressed in RDF. (a) The subject, predicate, and object in the triple are represented as URIs. The subject and object in a triple are linked by a predicate, mapping them to a graph structure with two nodes and one edge, respectively. (b) This is a part of the glycan structure information in GlyTouCan expressed in RDF. Each subject can be an object in another triple, and each object can be a subject in another triple, resulting in a large knowledge graph from a set of triples.

```
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix glycan:     <http://purl.jp/bio/12/glyco/glycan#> .
@prefix glytoucan:  <http://www.glytoucan.org/glyco/owl/glytoucan#> .
@prefix ns1:        <http://rdf.glycoinfo.org/resource-entry/> .
@prefix ns3:        <http://rdf.glycoinfo.org/glycan/> .

ns1:G22768VO    rdf:type                   glycan:Resource_entry .
ns1:G22768VO    glytoucan:contributor      <http://rdf.glycoinfo.org/glytoucan/contributor/userId/1> .

ns3:G22768VO    rdf:type                   glycan:Saccharide ;
                glycan:has_Resource_entry  ns1:G22768VO ;
                glytoucan:has_primary_id.  "G22768VO" .
```

FIGURE 2.7: An example of RDF described in the Turtle format. In this format, triples that share the same subject can be separated by a semicolon to avoid redundant repetition of the subject.

GlycoCoO (Yamada et al., 2021) for describing glycoconjugate-related information have been developed as ontologies. For example, `glycan:Resource_entry`, `glycan:has_Resource_entry`, and `glycan:Saccharide` in Figure 2.7 are the vocabularies defined in GlycoRDF.

Information described using RDF will be stored in a database for Semantic Web technologies called triplestore (Rohloff et al., 2007). The stored data can be retrieved by querying the database using a query language called SPARQL (Hogan, 2020). For example, Figure 2.8 shows an example of SPARQL query to retrieve a list of subjects whose predicate is `rdf:type` and whose object is `glycan:Saccharide` against a triplestore containing GlyTouCan's RDF data. By executing this query, all GlyTouCan IDs assigned to glycan structures registered in GlyTouCan, including `http://rdf.glycoinfo.org/glycan/G22768VO`, etc., can be retrieved. Figure 2.9 shows the result of this SPARQL query.

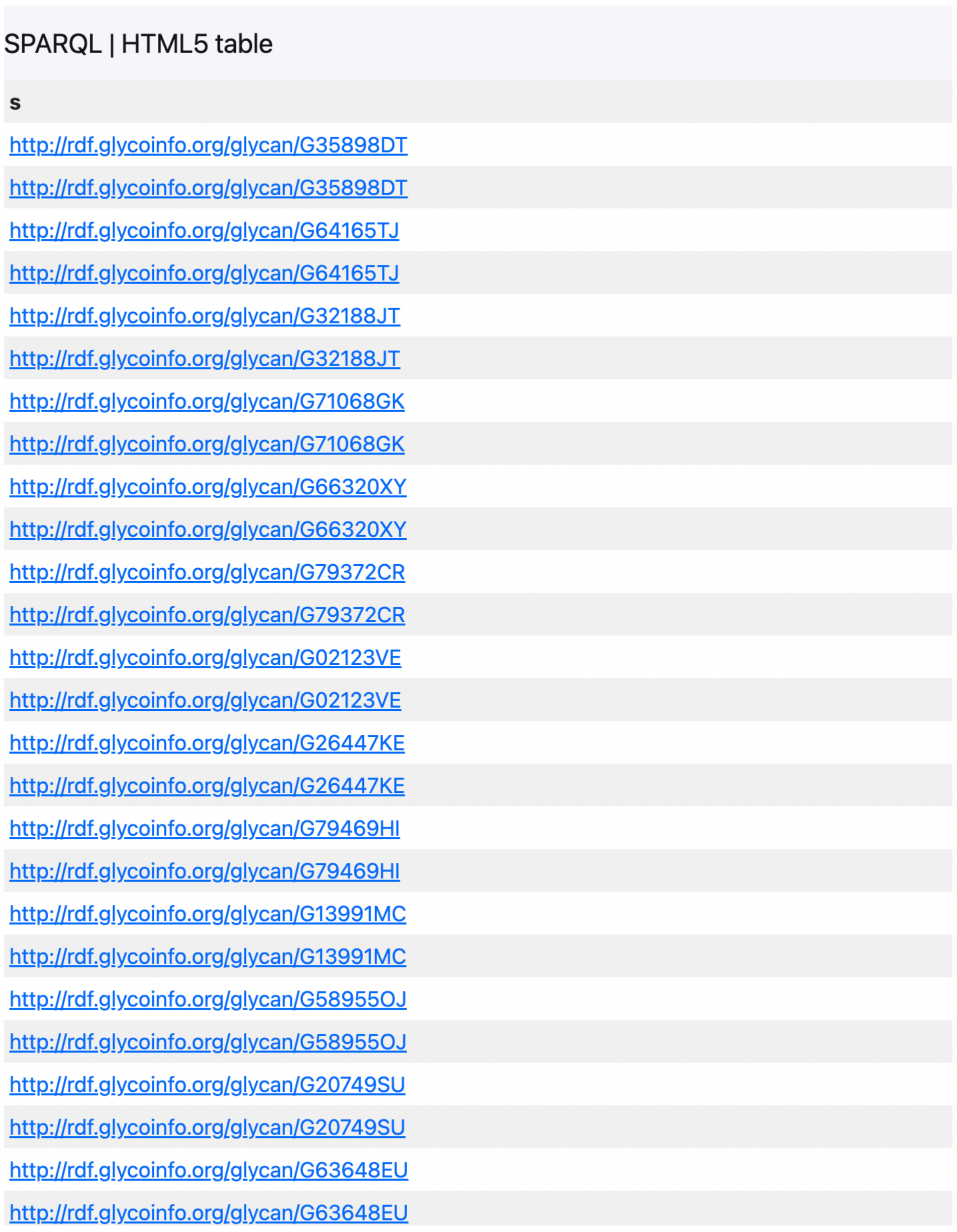
```
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX glycan: <http://purl.jp/bio/12/glyco/glycan#>

SELECT ?s WHERE {
    ?s rdf:type glycan:Saccharide
}
```

FIGURE 2.8: An example of a SPARQL query. The syntax of SPARQL is similar to the Turtle format, and defining the prefixes allows redundant URI descriptions to be omitted. Here is a SELECT statement that retrieves resources that match the conditions described in the WHERE clause.

2.2 Applying functional programming techniques to the development of data repositories

To enable continuous improvement of data repositories used in the long-term and to be flexible to future extensions, in this study I employed several statically typed programming languages with expressive type systems for the development, as well as several functional programming techniques that are the result of research in the field of computer science for safe software development. Functional programming is a declarative programming paradigm based on the theoretical foundation of the computational model called the lambda calculus formulated by Alonzo Church in the 1930s (Church, 1936; Church, 1941). It is a universal model of computation known to be equivalent to the Turing machine (Church, 1936; Kleene, 1936; Turing, 1936; Turing, 1937), the model of computation invented by Alan Turing and the basis of today's computers, and is used as one of the theoretical foundations of modern programming languages. Unlike the imperative programming paradigm adopted by many


SPARQL | HTML5 table

s
http://rdf.glycoinfo.org/glycan/G35898DT
http://rdf.glycoinfo.org/glycan/G35898DT
http://rdf.glycoinfo.org/glycan/G64165TJ
http://rdf.glycoinfo.org/glycan/G64165TJ
http://rdf.glycoinfo.org/glycan/G32188JT
http://rdf.glycoinfo.org/glycan/G32188JT
http://rdf.glycoinfo.org/glycan/G71068GK
http://rdf.glycoinfo.org/glycan/G71068GK
http://rdf.glycoinfo.org/glycan/G66320XY
http://rdf.glycoinfo.org/glycan/G66320XY
http://rdf.glycoinfo.org/glycan/G79372CR
http://rdf.glycoinfo.org/glycan/G79372CR
http://rdf.glycoinfo.org/glycan/G02123VE
http://rdf.glycoinfo.org/glycan/G02123VE
http://rdf.glycoinfo.org/glycan/G26447KE
http://rdf.glycoinfo.org/glycan/G26447KE
http://rdf.glycoinfo.org/glycan/G79469HI
http://rdf.glycoinfo.org/glycan/G79469HI
http://rdf.glycoinfo.org/glycan/G13991MC
http://rdf.glycoinfo.org/glycan/G13991MC
http://rdf.glycoinfo.org/glycan/G58955OJ
http://rdf.glycoinfo.org/glycan/G58955OJ
http://rdf.glycoinfo.org/glycan/G20749SU
http://rdf.glycoinfo.org/glycan/G20749SU
http://rdf.glycoinfo.org/glycan/G63648EU
http://rdf.glycoinfo.org/glycan/G63648EU

FIGURE 2.9: Results of an example SPARQL query. This shows the results of the SPARQL query shown in Figure 2.8 against GlyTouCan's SPARQL endpoint (`https://ts.glytoucan.org/sparql`) as of October 4, 2023. Since the query is for obtaining all glycan structure information registered in GlyTouCan, a large list of GlyTouCan IDs is obtained as a result of this operation.

programming languages, functional programming does not construct a program by a sequence of instructions to the computer, rather it constructs a program by composing and applying functions to form larger ones. Lisp (McCarthy, 1960), ML (Milner, 1972; MacQueen, Harper, and Reppy, 2020), Haskell (Marlow et al., 2010; Hudak et al., 2007), and other languages belonging to the functional programming language family have been used mainly only in academia for several decades, however, in the 2010s, C++11 (Meyers, 2014) and Java 8 (Urma, Fusco, and Mycroft, 2014), which are widely used in industry, also introduced lambda expressions, a feature derived from functional languages. Therefore, functional programming has been attracting attention from the industrial world in recent years. Furthermore, in recent years, development using the Haskell language has been reported at EMBL (European Molecular Biology Laboratory) in the bioinformatics field (Coelho et al., 2019), and it is expected that safe application development using functional programming will spread to the bioinformatics field as well.

2.2.1 Advantages of functional programming

Here I will briefly introduce some characteristics of functional programming compared to the conventional procedural or object-oriented programming. To begin, many procedural programming languages allow a series of operations to be packaged together as a reusable unit, known as a function or subroutine. Such functions can take certain inputs as parameters, perform calculations using those values, and return some value as its output. For example, the following program written in the Python language includes the function `area`, which takes the radius of a circle as an argument and calculates and returns the area of that circle.

Listing 2.1: Example of a function definition in the Python language

```python
pi = 3.141592

def area(radius):
    return radius * radius * pi
```

In this program, `pi` and `radius` are called variables and are used to give names to arbitrary values such as numbers or strings. In addition, a variable can be reassigned a value, so that the value corresponding to its name can be updated. The function `area` defined here can be called by giving a concrete value for the parameter variable, such as `area(5.0)`. As a result, the function will return the result value `78.5398`. However, this is not a function in the mathematical sense. In mathematics, a function means a mapping of inputs to outputs, and the same input values are always calculated to the same output values. In contrast, in the program shown above, the value of the variable `pi` may be reassigned at some point during the execution of the program to update the corresponding value. Consequently, giving the same input value to this function does not always result in the same output value being returned. While reassignment to variables is convenient in the development of small programs, it is not convenient in the development of large programs, such as data repository development, when the value of a variable may be updated by reassignment sometime in the future. This causes the problem that it becomes difficult for programmers to figure out if the program always calculates the correct value. This also increases the possibility of accidentally breaking existing functionality when adding new functionality to a data repository.

On the other hand, in functional programming, a function is defined as a function in the mathematical sense. In the case of the definition of the `area` function shown above, the fact that it depends on the mutable variable `pi`, which can be freely reassigned, makes the `area` function different from mathematical functions. Therefore, in functional programming, mutable variables such as `pi` are not used, but rather immutable variables whose names, once defined and associated with a particular value, cannot be reassigned thereafter. By ensuring that the value of the variable `pi` is never updated, the function is guaranteed to always calculate the correct area value for the given radius value as input. This property of a function that always returns the same output value for the same input value just like a mathematical function is called referential transparency. The following is the same function defined using Haskell, one of the most popular functional programming languages.

Listing 2.2: An example of a function definition in the Haskell language

```haskell
pi :: Double
pi = 3.141592

area :: Double -> Double
area radius = radius * radius * pi
```

In addition, functions are defined as first-class citizens in functional programming languages, allowing programmers to treat functions as values in the same way as numbers, strings, etc. This means not only that a function can be given a name with a variable as its value, but also that it can be passed as a parameter to another function, or return another function as the result value of it. Such functions that take functions as parameters or return a function as a result value are called higher-order functions. For example, suppose we write a program that computes the square of each of the integers from one to ten, extracts only the numbers less than 30, and sums them up. It can be written procedurally in the Python language as a function like the following:

Listing 2.3: Example of procedurally defined function in Python language

```python
def calc():
    nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

    squares = []
    for n in nums:
        squares.append(n * n)

    less_than_30 = []
    for n in squares:
        if n < 30:
            less_than_30.append(n)

    total = 0
    for n in less_than_30:
        total += n

    return total
```

Calling this program by writing `calc()` produces `14`, the result of `1 + 4 + 9`, as expected. In contrast, in functional programming, such programs can be written concisely using the `map` function, the `filter` function, and the `foldl` function, which are defined as higher-order functions in various functional programming languages. For example, this `calc` function can be expressed in the Haskell language as follows:

Listing 2.4: Example of a function defined in the Haskell language using higher-order functions

```haskell
import Data.Function ((&))

calc :: () -> Int
calc () =
    [1..10]
        & map (\n -> n * n)
        & filter (< 10)
        & foldl (+) 0
```

When this function is called as `calc ()`, `14` will be returned as the result value as in the previous example. In this function, the following operations are applied in order to a list containing integers from one to ten.

- The `map` function generates a new list containing 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, which is the result of squaring each integer from one to ten in the list. In this case, the part `\n -> n * n` represents an anonymous function that takes one integer parameter and returns the result of computing its square. Therefore, the `map` function is a higher-order function that takes another function as its parameter.

- A new list is generated by the `filter` function, in which integers greater than ten are removed from the list generated by the `map` function. In this case, the part `(< 10)` represents an anonymous function, or a predicate, that takes an integer as its parameter and returns a boolean value indicating whether the value is less than ten. This is a shorthand notation for the anonymous function `\n -> n < 10`.
- The `foldl` function performs convolution of each element in the list (1, 4, 9) generated by the `filter` function, and the elements are summed up in order from the left side with an initial value of zero. In this case, the part `(+)` represents an anonymous function that takes two integer parameters and returns their sum. This is a shorthand notation for the anonymous function `\a e -> a + e`. Of the two parameters `a` and `e`, `e` is set to one of the integers in the original list, from left to right. On the other hand, `a` is an accumulator variable containing the total result of the list to the left of the current `e` value.

Utilizing a series of higher-order functions as tools not only reduced the number of lines in the program, but also reduced the number of long-lived variables that exist throughout the execution of the `calc` function, such as the `squares` variable found in the version written in the Python language. Variables such as `n` that appear in functions passed as parameters to higher-order functions are extremely short-lived variables that exist only during the execution of the corresponding higher-order function. This contributes to reducing the possibility of mistakes, such as programmers accidentally referencing the value of a different variable.

Syntaxes such as `if` and `for` statements, which appear in the version written in the Python language, are fundamental tools in modern programming languages for explicitly controlling the control flow of a program in terms of conditional branches and repetition, respectively. However, before Edsger W. Dijkstra's proposal of the structured programming paradigm (Dijkstra, 1968), the control flow of these programs was controlled by utilizing the `goto` statement, a feature that allows programmers to directly access the jump instructions provided by the CPUs. However, programs written in such an unordered manner are difficult to maintain, and the development of large-scale programs is extremely difficult. As the structured programming paradigm became more common, the `if` and `for` statements became a means for programmers to utilize the CPU's basic jump instructions in an orderly fashion. As a result, they became the vocabulary for expressing program structure, such as "conditional branching" and "repetition". This allowed programmers to begin expressing the structure of their programs abstractly, away from the actual operating principles of the computer. Higher-order functions introduced by functional programming languages, such as the `map`, `filter`, and `foldl` functions, provide programmers with a rich vocabulary for expressing the intent of their programs at a higher level of abstraction than the `if` and `for` statements. For example, the `map` function, when used with a list, can express the process of transforming (mapping) each element in the list using a function passed as a parameter. Therefore, it is guaranteed that the new list generated as a result of the `map` function will always contain the same number of elements as those in the original list. This clearly gives more information to programmers reading the program than the `for` statement, which has no meaning beyond repetition. Thus, in functional programming, the use of higher-order functions not only reduces the amount of code, but also expresses the intent of the program in a way that is more easily understood by humans. This allows for more productive development of larger programs, such as data repositories.

Type systems are what further strengthen these advantages of functional programming. In programming languages, a type system is a computationally tractable syntactic method for ensuring the absence of certain behaviors by classifying every part (term) in a program according to the type of the value computed from it (Pierce, 2002). For example, integers, floating point numbers, strings, and boolean values are all distinguished by the type system as different data types. Statically typed programming languages are equipped with a type checker, which allows programmers to automatically check for the presence of a certain kind of error, called type errors, in the program. If a program contains incorrect operations, such as adding strings and integers, the type checker will report a type error. Thus, programmers can know without actually executing the program that there is a problem in it and that it needs to be fixed. This reduces the possibility of unintentionally breaking program functionality when modifying an existing program.

In many programming languages, it is possible to define new user-defined data types that contain values of multiple data types. In addition to that, functional programming languages can define "data types that may have several different structures". The following is an example of such a data type definition in the Haskell language.

Listing 2.5: Example of data type definition in the Haskell language

```haskell
data Option a = None | Some a
```

Here, the value of the data type `Option a` is constructed from either `None` or `Some` data constructors. This data type represents the fact that the value "may be absent (None), or it may be present

(Some).” For example, the data type of the value `Some 123` would be `Option Int`. Since a variable of `Option Int` data type may contain the value `None`, this data type represents the fact that there may or may not be an integer inside. If programmers try to use values of this data type for arithmetic operations as if they were ordinary integers, the type system will tell them before program execution that they are about to be used incorrectly by generating a type error. This helps programmers realize that they can perform that calculation only after checking whether this value indeed contains an integer value or not. Thus, the use of such data types in a program can explicitly indicate to programmers that there is a possibility that a particular operation may fail for some reason, such as a network failure, and the value may not be calculated. In other words, data types also serve as documentation between programmers in a program. This will facilitate the development of data repositories for multiple people and make the handover smoother. Furthermore, by defining the `map` function described earlier for data types such as `Option a` in the same way as for lists, it is possible to safely describe an operation that is executed only when the value exists. While functional programming languages represent the possibility of program errors as data types in this way, existing industrial languages have used the convention of returning an invalid value, called “null,” in the event of an error, or have utilized a mechanism called an exception handling. However, none of these allow the type system to detect possible errors before the program is executed, and as a result, if a programmer forgets to write code to check for errors in the program, the program will crash due to run-time errors. If such a crash occurs in a data repository, for example, the data upload process by researchers will fail, or the data extraction process for uploaded files will terminate abnormally in the middle. Even if an analysis function that takes a long time to execute is incorporated into the data repository, it may become an unreliable function that causes the program to terminate abnormally in the middle of the process. Therefore, in the development of large programs such as data repositories, it is considered extremely effective to utilize data types and type systems through functional programming to deal with errors to improve the reliability of the system as a whole.

2.2.2 Application of functional programming to this study

In this study, I used several statically typed languages as programming languages for developing the data repositories. First, the user interface of GlyComb was developed primarily using the TypeScript (Cherny, 2019) language, with some functionality developed using the Scala (Odersky, Spoon, and Venners, 2008) language. Although the TypeScript language itself does not strictly belong to the family of functional programming languages, it takes advantage of the benefits of functional programming through the combined use of a library called `fp-ts`[1]. These programming languages make it easy to write reliable programs that rely on the referential transparent functions described in the previous section, and to write programs using a variety of vocabularies with higher-order functions. In addition, the program for UniCarb-DR, which was previously written in the old-style Java language, had several broken functions such as search and dataset submission functions, and it was difficult to identify the cause of errors because checks for the invalid "null" values were missing everywhere in the program. Therefore, the system was completely rewritten in the Scala language to improve it into a more robust and reliable system using more functional programming techniques. The Scala language is compatible with the Java language, which is very widely adopted in the software development industry, and is a language that combines the different programming paradigms of object-oriented programming and functional programming. For example, the `Option a` data type shown in the previous section can be defined in the Scala language as follows, with compatibility with object-oriented programming such as the Java language.

Listing 2.6: Example of data type definition in the Scala language

```
sealed abstract class Option[+A]
case object None extends Option[Nothing]
final case class Some[+A](value: A) extends Option[A]
```

As described in more detail in later sections, the application of functional programming techniques with these languages enabled the GlyComb and UniCarb-DR programs to check for possible errors not only through programmer attention but also through type checking by the type systems. It is expected that this will allow the data repositories to continue to expand their functionality over time without breaking existing functionality in the future.

2.3 The design of the GlyComb system

2.3.1 Fundamental technologies used to build the GlyComb system

The GlyComb server-side system consists of several components. We developed most of the server-side logic, including the application programming interface (API), using the Java 17 programming language. As a relational database management system to directly store each entry submitted by users, we adopted the MariaDB Community Server. We also employed OpenLink Virtuoso Open Source

[1] https://gcanti.github.io/fp-ts/

Edition 7 for our implementation of a triplestore that stores the RDF-ized information converted from the relational database. For the RDF data insertion into this triplestore, we utilize SPARQList (Katayama and Kawashima, 2017). All these server-side components are running on our on-premise servers, which are built on containerized virtualization technology based on Docker (Merkel, 2014).

To develop the front-end application, including the user interface that runs in web browsers, we used TypeScript for the majority of the development. We also used Node.js (Tilkov and Vinoski, 2010) with a number of JavaScript libraries, including Webpack (Zammetti, 2020), to generate a JavaScript bundle file that web browsers can interpret. We also developed a conversion software to extract GlyComb input entries from the summary spreadsheet file generated from PMI-Byonic. For the development of this tool, we adopted the Scala programming language and used the Scala.js (Doeraene, 2013) compiler to generate a JavaScript bundle file that runs in web browsers. Then, we embedded the bundle file in the front-end application so that it is available from the user interface.

In the development of GlyComb, each component is managed separately by Git (Loeliger and McCullough, 2012), a popular version control software for source code and configuration files, and each is hosted on an on-premise server in Kinoshita Lab. The following is the eight major components related to the GlyComb system (the URLs of the respective Git repositories are listed in the footnotes).

(i) User interface programs that run on web browsers[2]

(ii) Reusable custom component library for the user interface[3]

(iii) Utility software for extracting the identification results of glycoproteomics experiments in PMI Byonic-derived spreadsheet files[4]

(iv) Public APIs (application programming interfaces) for user interface programs[5]

(v) Relational database-related configurations and private APIs for communicating with the database[6]

(vi) DTO (data transfer object) declarations commonly used by both public and private API programs[7]

[2]https://gitlab.glyco.info/glycosmos/glycombgroup/components
[3]https://gitlab.glyco.info/glycosmos/glycombgroup/library/glycomb-ui-libraries
[4]https://gitlab.glyco.info/glycosmos/glycombgroup/library/comberter
[5]https://gitlab.glyco.info/glycosmos/glycombgroup/api
[6]https://gitlab.glyco.info/glycosmos/glycombgroup/backend
[7]https://gitlab.glyco.info/glycosmos/glycombgroup/library/glycomb-data-model

(vii) Batch processing programs to validate submitted data stored in the relational database and RDF-ize it[8]

(viii) Triple store-related configuration files for storing RDF-ized submission data[9]

Of these components, all but (i), (ii), and (iii), which run in the web browser of the user's computer, run in a separate Docker container and interoperate with each other as shown in the conceptual diagram in Figure 2.10.

2.3.2 Error checking of communication between the user interface and the API using functional programming techniques

GlyComb is implemented as a web application that researchers access using a web browser on their own computers and browse and submit their data from the user interface displayed on the browser. Consequently, it is necessary to have frequent communication between the user interface and the APIs running on the server. However, since the communication is between programs running on different computers, this communication may inevitably fail due to network failures or other factors. Moreover, if the data uploaded by a researcher is in a format that GlyComb cannot accept, it will also result in an error. Therefore, such communication requires proper handling of errors caused by various factors. If such errors are not properly handled, for example, the programs on the user interface could implicitly crash in the background without the researchers noticing while they are in the middle of using GlyComb, and it could stop working as they intended. To prevent such problems, GlyComb's user interface was developed by applying functional programming techniques to force programmers to properly handle errors in any sort of communication that might fail. For example, the following is a program written in the TypeScript language that the user interface executes in the web browsers to retrieve the number of glycopeptide entries currently published in GlyComb from the API.

Listing 2.7: Example program to get the number of published glycopeptide entries from GlyComb's API

```
const responseOpt = await fetchFromGlyCombApi<AccessionCountResponse>({
    url: 'https://glycomb.glycosmos.org/api/accessions-count?datatype=Glycopeptide',
    method: 'GET',
    contentType: 'application/json',
    authorizationRequired: false,
});

if (Option.isSome(responseOpt)) {
    this.glycopeptideAccessionCount = responseOpt.value.contents;
```

[8]https://gitlab.glyco.info/glycosmos/glycombgroup/batch
[9]https://gitlab.glyco.info/glycosmos/glycombgroup/endpoint

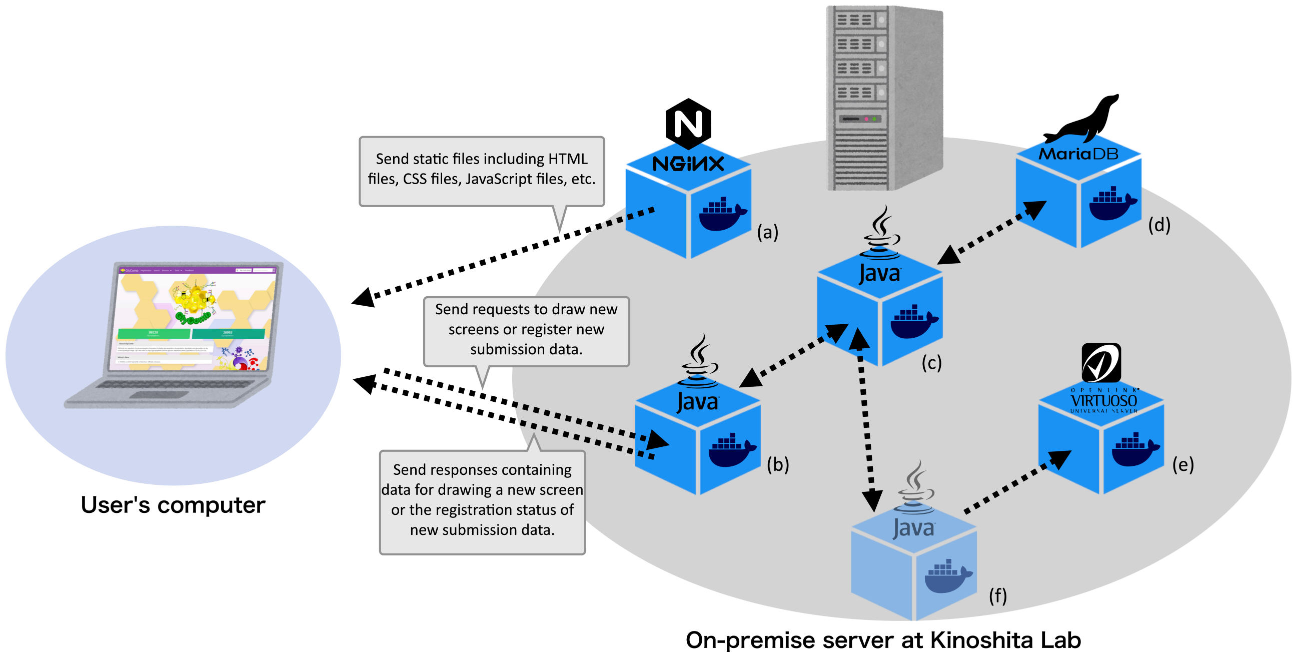
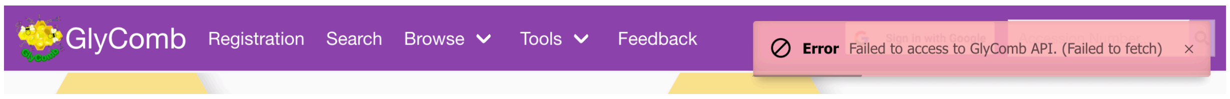

Send static files including HTML files, CSS files, JavaScript files, etc.
Send requests to draw new screens or register new submission data.
Send responses containing data for drawing a new screen or the registration status of new submission data.
NGINX
(a)
Java
(b)
Java
(c)
MariaDB
(d)
VIRTUOSO
(e)
Java
(f)
User's computer
On-premise server at Kinoshita Lab

FIGURE 2.10: Conceptual diagram of the entire GlyComb system composed of multiple Docker containers. The dashed arrows in the figure represent the flow of data between each container and the web browser on the user's computer. (a) The user interface program (i), the custom component library (ii), and the utility software for extracting data from spreadsheets derived from PMI-Byonic (iii) in the above list of components, are combined by Webpack with other JavaScript libraries into a bundle file that runs in the web browser. This artifact will be hosted as a static file on the container running the Nginx web server. (b) The container running the public APIs written in Java receives all requests directly from the user's web browser and returns responses to it by interacting with the private APIs. (c) The private APIs are also written in Java and retrieve data from the relational database or insert data into it in response to requests from the public API container and return the results. In this way, GlyComb keeps the data in the database container secure by allowing only private APIs that are not exposed to users to interact with the database container. (d) The database container running MariaDB, which is an RDBMS, stores raw submission data submitted by users. (e) The container running the OpenLink Virtuoso server, which is a triplestore, accumulates RDF-ized submission data that is normalized and generated during batch processing. (f) The batch processes that validates user submission data, RDF-izes it, and stores it in the triplestore using SPARQList are also written in Java. These processes are scheduled to run automatically every several hours in a temporary container that is short-lived and immediately discarded after the batch execution.

```
}
```

In this program, `responseOpt` is a variable having `Option` data type that expresses the possibility of error in functional programming. The `fetchFromGlyCombApi` function is a generic function that is frequently used throughout the GlyComb user interface to communicate with the APIs. It returns a `Some` value if the data was successfully fetched from the API, and returns a `None` value if the fetch failed. Thanks to this function, GlyComb's user interface was written without the need for exhaustive checking for invalid "null" values or cumbersome exception handling, and with a minimum effort to perform the necessary error handling. Here, this result value is directly assigned to the `responseOpt` variable.

In addition, if the fetch of data from the API fails, the `fetchFromGlyCombApi` function will automatically display an error message on the user interface as shown in Figure 2.11 to clearly indicate to the user that the communication has failed before returning a `None` value. Therefore, this user interface program eliminates the need to explicitly write a process to display error messages for each communication between the server and the web browser. This prevents, in principle, the issue of forgetting to display error messages.



Error Failed to access to GlyComb API. (Failed to fetch)

FIGURE 2.11: Example of a pop-up on the GlyComb user interface displaying an error message. `fetchFromGlyCombApi` function automatically displays such error messages when communication fails or an error is returned from the API. Thus, there is no implicit program crash without the user's awareness.

It is expected here to get the number of glycopeptide entries from the API on the server in JSON (JavaScript Object Notation) (Bray, 2014) format as shown below.

LISTING 2.8: Example response from the API showing the number of published glycopeptide entries

```
{
    "errcode": "",
    "contents": 99228,
    "logs": {
        "url": "http://glycomb-api:4567/api/accessions-count"
    }
}
```

That is, the number of published glycopeptide entries returned by the server will be stored in the `contents` field in the response. Additionally, the `errcode` field is set to an error code when the server determines that the request from the web browser is invalid. In that case, the `contents` field would be empty and would not contain the data requested by the web browser. The `fetchFromGlyCombApi` function used by the GlyComb user interface is designed to automatically check for error codes in the `errcode` field. If it does, it automatically displays the appropriate error message as in Figure 2.11 and returns a `None` value as the result.

As a result, the `responseOpt` variable will take one of two states, "may contain the JSON-formatted data described above, or may not". Although the web browser does send a request to the API on the server to obtain the number of published glycopeptide entries, since the `responseOpt` variable is of type `Option`, if a programmer mistakenly tries to treat it as a number, then the type checker will report a type error. This allows the programmer to fix the mistake without actually running the program on the web browser. This avoids the problem that programmers release the programs without realizing their mistakes. To verify whether a value actually exists in the `responseOpt` variable, and to retrieve the value if it does, we can use the `Option.isSome` function provided by the `fp-ts` library, as shown in the program above, to check the status of the `responseOpt` variable.

2.3.3 Batch processing and validation of submitted data

GlyComb accepts GlyTouCan ID and PMI-Byonic-like own monosaccharide composition notation as glycan structure notation in glycopeptide and glycoprotein entries, thus it is necessary to normalize the submitted entries in order to check whether they have already been registered by other users previously. However, such verification and validation could be a time consuming process. Therefore, GlyComb adopts a data validation system based on batch processing in order to validate submitted data without sacrificing the ease of use for the users. The system uses two separate databases for different purposes: a relational database (MariaDB) and a triplestore (OpenLink Virtuoso). First, any submission data entered by users is initially stored in the MariaDB relational database via GlyComb's public and private APIs. During this data upload process, a brief data check is performed in which only the format of the submitted data by users is verified. At this point, all user submitted data is not yet publicly accessible on GlyComb. The users can publish their submitted data to the public only after the formal data validation process is completed. The formal data validation process runs automatically on the server as a batch process, a periodically executed process. All newly registered submissions to the relational database are validated by this process, and the submission data are normalized by converting any glycan structure data in them to GlyTouCan IDs or by retrieving the actual amino acid sequence data of the glycoprotein entry specified with the UniProt ID from UniProt's API. Figure 2.12 (a) illustrates this series of processes.

Only submitted data that has passed this validation process with no errors found can be made publicly available later. The user can make the passed entries publicly available on the "User Profile" screen of on GlyComb. The published entries are later automatically converted to RDF data (RDF-ization) by another batch process and stored in the triplestore, OpenLink Virtuoso. Figure 2.12 (b) illustrates the flow of this process.

2.3.4 RDF-ization of data registered in GlyComb

In order to facilitate integration with various existing resources, including GlyTouCan, GlyComb is built on top of Semantic Web technologies. To convert the glycopeptide and glycoprotein entries submitted by researchers into RDF data that can be stored in a triplestore, we used the glycoconjugate ontology (GlycoCoO), an ontology for standardizing glycoconjugate data as RDF data. This ontology is an extension of GlycoRDF for standardizing glycomics information. Currently, GlyComb only stores glycopeptide and glycoprotein entries, whereas GlycoCoO is an ontology that can represent various glycoconjugate data including glycolipids. Therefore, we converted each entry of GlyComb into RDF data by adopting only a part of the definition of this GlycoCoO ontology. Figure 2.13 shows the RDF schema for representing a peptide sequence with only one glycosylated peptide residue as RDF data using GlycoCoO. In RDF, all data is encoded into a triple of subject, predicate, and object, which form a directed graph with the subject and object as nodes and the predicate as edges.

The RDF data stored in the triplestore can be freely retrieved, even by users not registered with GlyComb, by issuing a SPARQL query on GlyComb's SPARQL endpoint (`https://glycomb.glycosmos.org/sparql`). Users can also submit SPARQL queries to the SPARQL endpoint programmatically. This allows GlyComb to provide flexible data retrieval functionalities along with data from existing glycoscience research resources such as GlyTouCan and GlyCosmos. Figure 2.14 shows an example SPARQL query for this SPARQL endpoint and its search results.

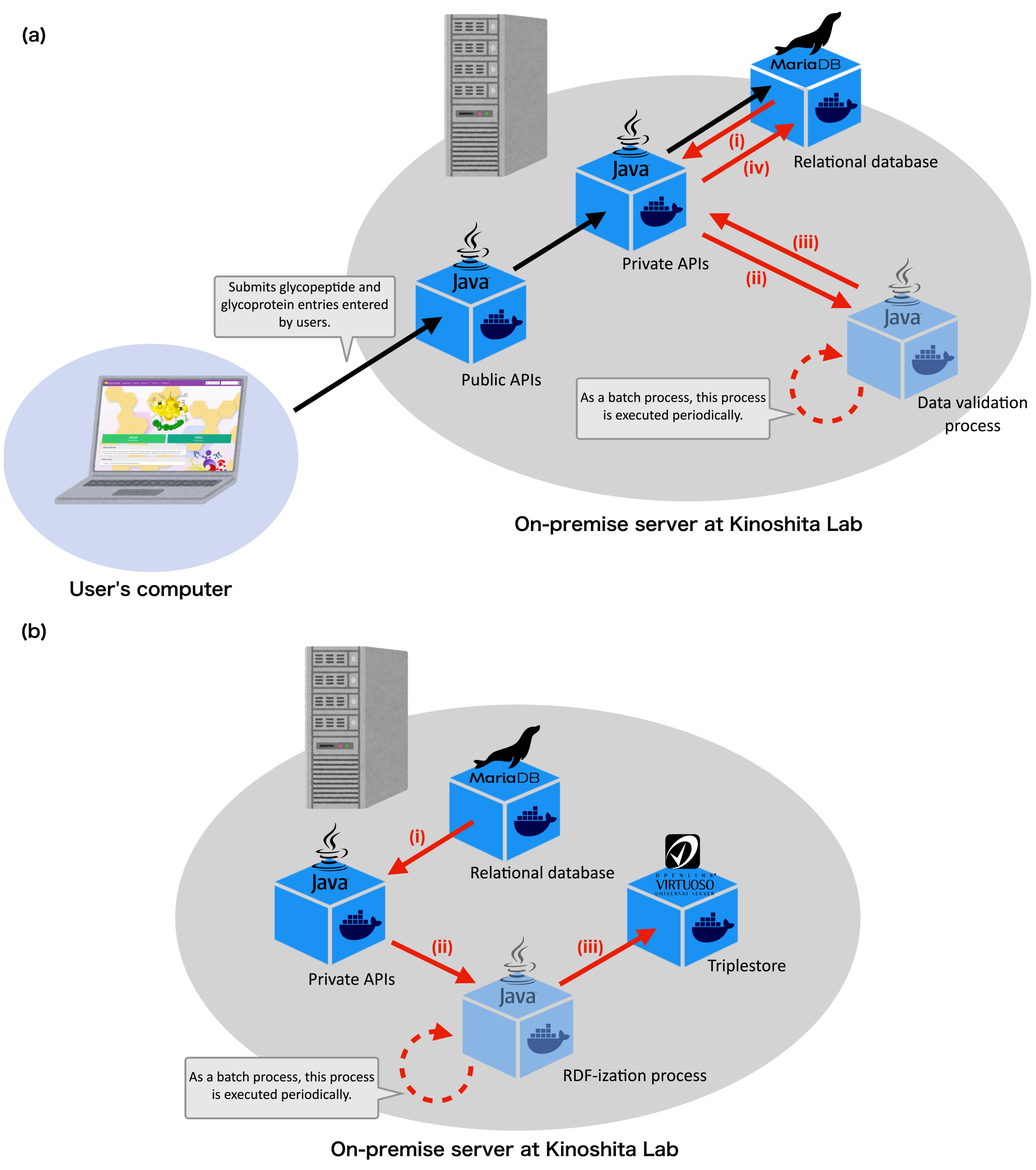

(a)
MariaDB
Java
(i)
(iv)
Relational database
Private APIs
(iii)
(ii)
Submits glycopeptide and glycoprotein entries entered by users.
Java
Public APIs
Java
As a batch process, this process is executed periodically.
Data validation process
On-premise server at Kinoshita Lab
User's computer
(b)
MariaDB
(i)
Java
Relational database
VIRTUOSO
(ii)
(iii)
Private APIs
Java
Triplestore
RDF-ization process
As a batch process, this process is executed periodically.
On-premise server at Kinoshita Lab

FIGURE 2.12: The flow of validation and RDF-ization of submitted data through batch processing. (a) First, user submission data is stored in the relational database, MariaDB, via the public and private APIs, as represented by the solid black line. Then, a Docker container for validation of the submission data will be launched by a periodically executed batch process. This container retrieves the newly registered submission data from the relational database via the private APIs as shown in (i) and (ii) and performs the data validation and normalization. Subsequently, the status of this process (success or failure) and the normalized data will be stored again in the relational database as shown in (iii) and (iv). (b) For RDF-ization process, a temporary working Docker container will be launched by a batch process in the same way. Published submission data is retrieved from the relational database via the private APIs as shown in (i) and (ii), and the RDFization process is performed. The resulting RDF data will be stored in the triple store, OpenLink Virtuoso as shown in (iii).

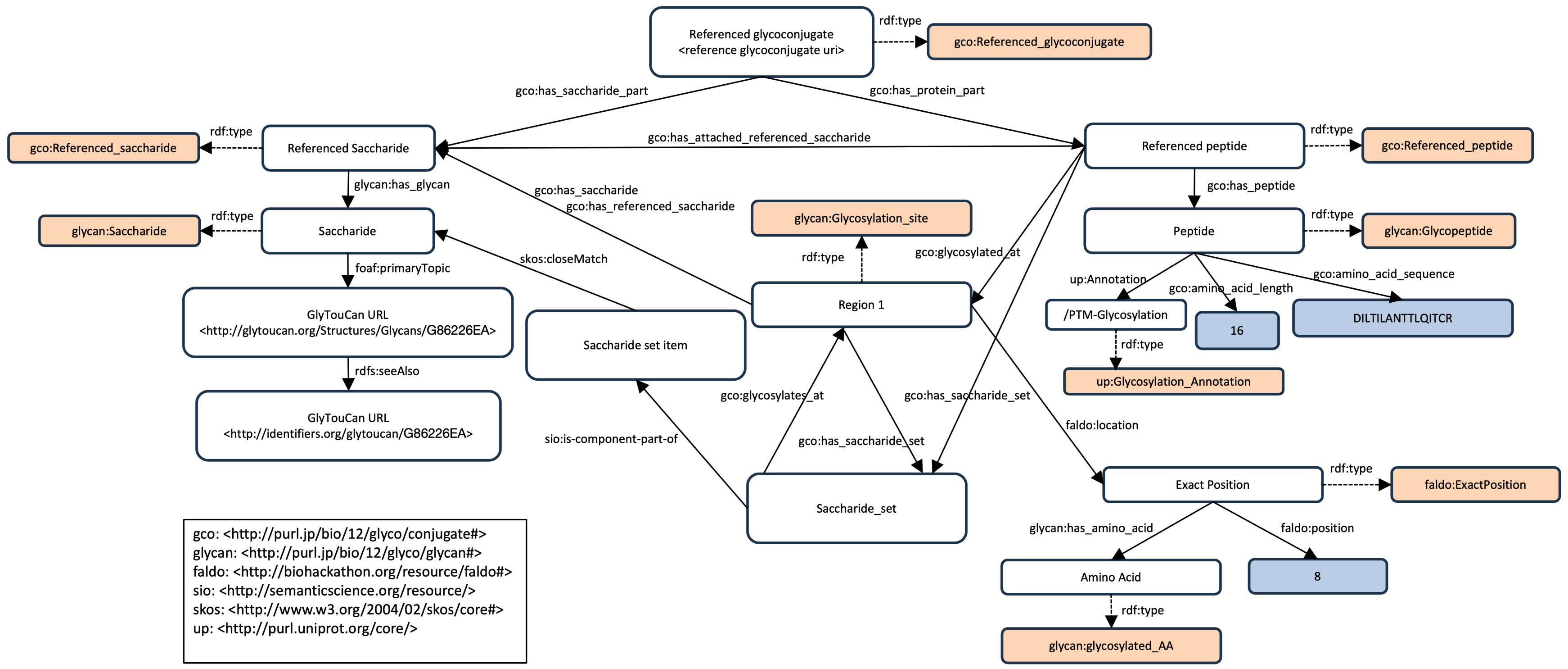
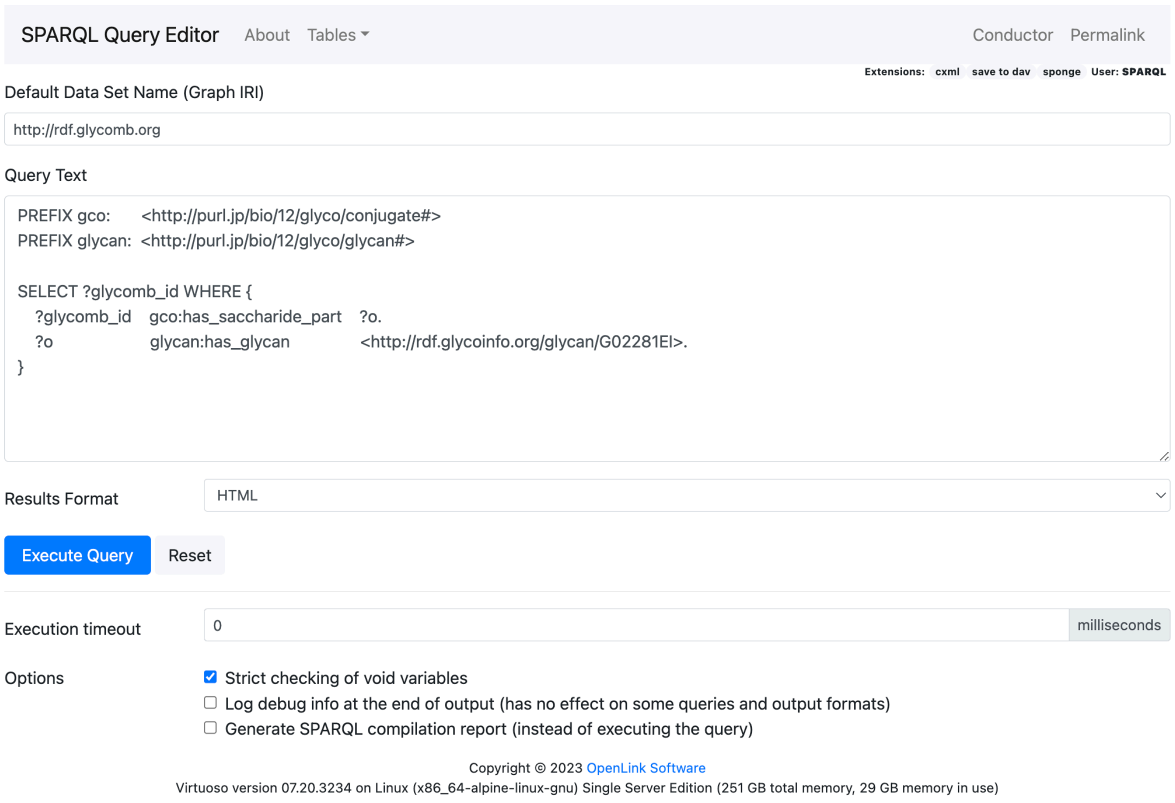

Referenced glycoconjugate <reference glycoconjugate uri>
rdf:type
gco:Referenced_glycoconjugate
gco:has_saccharide_part
gco:has_protein_part
gco:has_attached_referenced_saccharide
gco:Referenced_saccharide
Referenced Saccharide
Referenced peptide
gco:Referenced_peptide
glycan:has_glycan
gco:has_peptide
gco:has_saccharide
gco:has_referenced_saccharide
glycan:Saccharide
Saccharide
glycan:Glycosylation_site
Peptide
glycan:Glycopeptide
skos:closeMatch
gco:glycosylated_at
up:Annotation
gco:amino_acid_length
gco:amino_acid_sequence
foaf:primaryTopic
Region 1
/PTM-Glycosylation
16
DILTILANTTLQITCR
GlyTouCan URL <http://glytoucan.org/Structures/Glycans/G86226EA>
Saccharide set item
up:Glycosylation_Annotation
rdfs:seeAlso
gco:glycosylates_at
gco:has_saccharide_set
faldo:location
GlyTouCan URL <http://identifiers.org/glytoucan/G86226EA>
sio:is-component-part-of
Exact Position
faldo:ExactPosition
Saccharide_set
glycan:has_amino_acid
faldo:position
gco: <http://purl.jp/bio/12/glyco/conjugate#>
glycan: <http://purl.jp/bio/12/glyco/glycan#>
faldo: <http://biohackathon.org/resource/faldo#>
sio: <http://semanticscience.org/resource/>
skos: <http://www.w3.org/2004/02/skos/core#>
up: <http://purl.uniprot.org/core/>
Amino Acid
8
glycan:glycosylated_AA

FIGURE 2.13: RDF schema of GlyComb representing a glycopeptide entry with the glycan structure "HexNAc(4)Hex(7)NeuAc(1)" binding to the asparagine residue at residue number 8 of the peptide sequence "DILTILANTTLQITCR". This RDF data is part of a glycopeptide entry with GlyComb ID GC0017EF, which was derived from mouse vascular endothelial growth factor receptor 2 (VEGFR2) (UniProt ID: P35918). In this directed graph, each node represents a subject or an object, and a predicate is represented between the subject and the object as an edge in the subject-to-object direction. All glycan structure information entered by researchers is normalized to the corresponding GlyTouCan IDs and converted to RDF data through batch processing on the server. Orange boxes represent classes in RDF, and blue boxes represent literal values in RDF.

(a)



Default Data Set Name (Graph IRI)

http://rdf.glycomb.org

Query Text

```
PREFIX gco:     <http://purl.jp/bio/12/glyco/conjugate#>
PREFIX glycan:  <http://purl.jp/bio/12/glyco/glycan#>

SELECT ?glycomb_id WHERE {
    ?glycomb_id   gco:has_saccharide_part   ?o.
    ?o            glycan:has_glycan         <http://rdf.glycoinfo.org/glycan/G02281EI>.
}
```

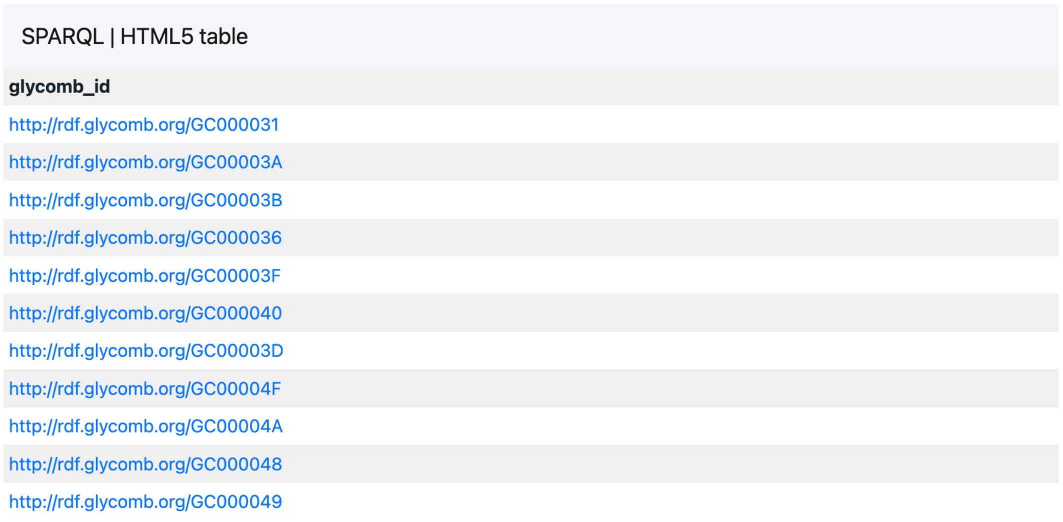
Results Format: HTML

Execution timeout: 0 milliseconds

Options
- [x] Strict checking of void variables
- [ ] Log debug info at the end of output (has no effect on some queries and output formats)
- [ ] Generate SPARQL compilation report (instead of executing the query)

Copyright © 2023 OpenLink Software
Virtuoso version 07.20.3234 on Linux (x86_64-alpine-linux-gnu) Single Server Edition (251 GB total memory, 29 GB memory in use)

(b)



SPARQL | HTML5 table

glycomb_id
http://rdf.glycomb.org/GC000031
http://rdf.glycomb.org/GC00003A
http://rdf.glycomb.org/GC00003B
http://rdf.glycomb.org/GC000036
http://rdf.glycomb.org/GC00003F
http://rdf.glycomb.org/GC000040
http://rdf.glycomb.org/GC00003D
http://rdf.glycomb.org/GC00004F
http://rdf.glycomb.org/GC00004A
http://rdf.glycomb.org/GC000048
http://rdf.glycomb.org/GC000049

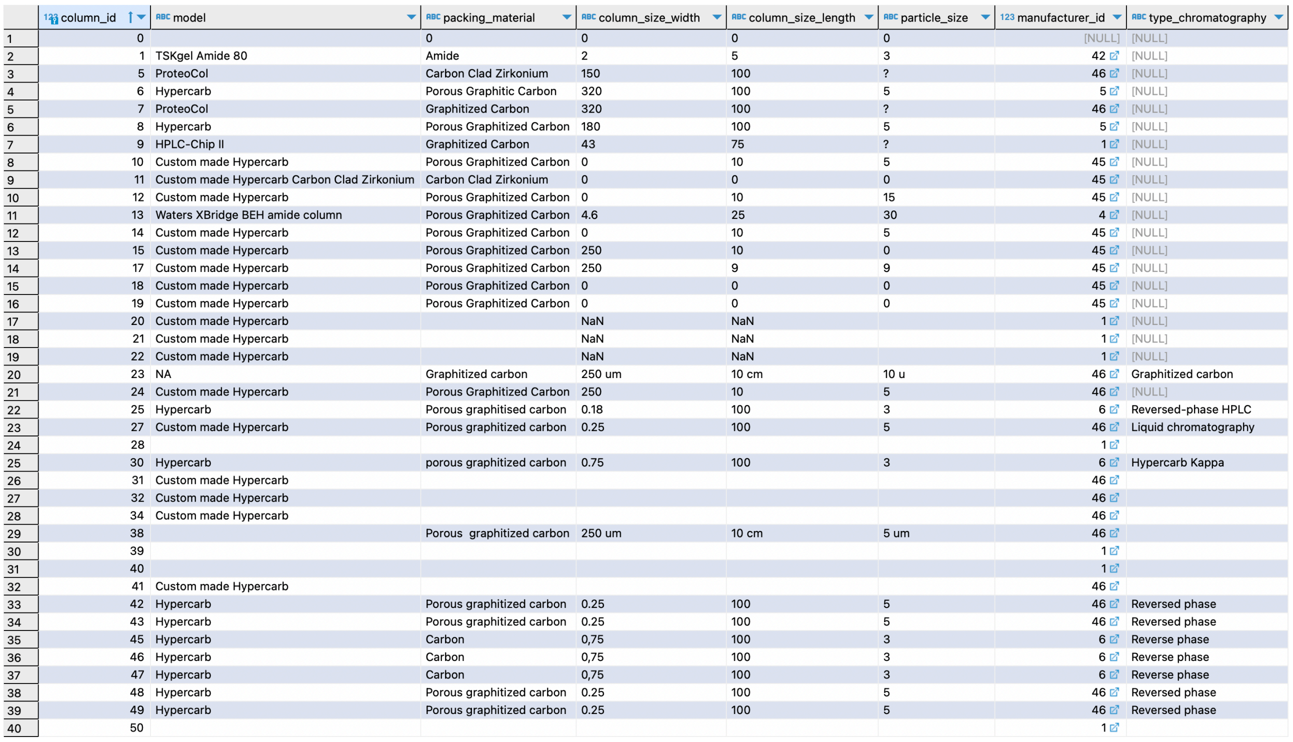
FIGURE 2.14: Example of a SPARQL query against the GlyComb SPARQL endpoint and its search results. (a) A screenshot of GlyComb's SPARQL endpoint (`https://glycomb.glycosmos.org/sparql`) as of December 30, 2023. Here, a SPARQL query is entered in the text area to retrieve the list of GlyComb entry IDs, containing the GlyTouCan ID G02281EI (`https://glytoucan.org/Structures/Glycans/G02281EI`) corresponding to the monosaccharide composition "Hex(4)HexNAc(5)Fuc(1)" as the glycosylation data. (b) By clicking the "Execute Query" button, the search against the triplestore is executed and a list of GlyComb entries that match the condition will be retrieved.

2.4 System design to enable further collaboration between UniCarb-DR and GlycoPOST

2.4.1 Re-designing the UniCarb-DR system to improve its reliability

First of all, it was necessary to recover the functionality and improve the stability of UniCarb-DR, whose search and data submission functions were broken. To accomplish this, I first investigated the entire source code of UniCarb-DR. As a result, it was found that the UniCarb-DR program at that time was scattered with ad hoc checks for the invalid "null" values everywhere, and it was also difficult to distinguish the parts that potentially caused errors from those that did not. Therefore, I attempted to reimplement UniCarb-DR aiming to obtain the benefits of functional programming in the development of UniCarb-DR as well. I rewrote all of UniCarb-DR's server-side programs, which were written in the older version of the Java programming language, entirely in the Scala programming language. Eventually, all previously broken UniCarb-DR functionality was restored, dramatically improving the stability and maintainability of UniCarb-DR.

Here I will briefly introduce some of the techniques that contributed to the dramatic improvement in maintainability of UniCarb-DR. First, the primary cause of the error-prone behavior and the need for ad hoc error checking everywhere in the previous UniCarb-DR was due to the data stored in the relational database. Figure 2.15 shows the actual data of the table that stores column information used in HPLC experiments in UniCarb-DR. The most notable data in this table is the data stored in the rightmost column named `type_chromatography`. This attribute is supposed to be filled in with the type of chromatography experiment performed, however, since it is not mandatory, this attribute is left empty in some rows. However, besides those rows, there are also rows where this attribute is marked as `[NULL]`. Both of these mean that no value was entered for the `type_chromatography` attribute, but one means "an empty valid string is stored," while the other expresses the absence of data by storing a "null" value. This means that when these values are retrieved from the database by a program, they must be checked twice: once for being an empty string and once for being an invalid "null" value. The reason why this is troublesome is that programmers must manually map the myriad of tables and their attributes in the database to the variables in the UniCarb-DR server-side program, making it very difficult to write a program that does not make mistakes about which attributes need to be checked for a "null" value and which others need to be checked for an empty string. Even worse, updating the definitions of tables in the database still requires the programmer to manually modify the server-side program, however, if this modification contains leaks, it can cause unintended program crashes, which are generally quite time-consuming to identify the cause.



	column_id	model	packing_material	column_size_width	column_size_length	particle_size	manufacturer_id	type_chromatography
1	0		0	0	0	0	[NULL]	[NULL]
2	1	TSKgel Amide 80	Amide	2	5	3	42	[NULL]
3	5	ProteoCol	Carbon Clad Zirkonium	150	100	?	46	[NULL]
4	6	Hypercarb	Porous Graphitic Carbon	320	100	5	5	[NULL]
5	7	ProteoCol	Graphitized Carbon	320	100	?	46	[NULL]
6	8	Hypercarb	Porous Graphitized Carbon	180	100	5	5	[NULL]
7	9	HPLC-Chip II	Graphitized Carbon	43	75	?	1	[NULL]
8	10	Custom made Hypercarb	Porous Graphitized Carbon	0	10	5	45	[NULL]
9	11	Custom made Hypercarb Carbon Clad Zirkonium	Carbon Clad Zirkonium	0	0	0	45	[NULL]
10	12	Custom made Hypercarb	Porous Graphitized Carbon	0	10	15	45	[NULL]
11	13	Waters XBridge BEH amide column	Porous Graphitized Carbon	4.6	25	30	4	[NULL]
12	14	Custom made Hypercarb	Porous Graphitized Carbon	0	10	5	45	[NULL]
13	15	Custom made Hypercarb	Porous Graphitized Carbon	250	10	0	45	[NULL]
14	17	Custom made Hypercarb	Porous Graphitized Carbon	250	9	9	45	[NULL]
15	18	Custom made Hypercarb	Porous Graphitized Carbon	0	0	0	45	[NULL]
16	19	Custom made Hypercarb	Porous Graphitized Carbon	0	0	0	45	[NULL]
17	20	Custom made Hypercarb		NaN	NaN		1	[NULL]
18	21	Custom made Hypercarb		NaN	NaN		1	[NULL]
19	22	Custom made Hypercarb		NaN	NaN		1	[NULL]
20	23	NA	Graphitized carbon	250 um	10 cm	10 u	46	Graphitized carbon
21	24	Custom made Hypercarb	Porous Graphitized Carbon	250	10	5	46	[NULL]
22	25	Hypercarb	Porous graphitised carbon	0.18	100	3	6	Reversed-phase HPLC
23	27	Custom made Hypercarb	Porous graphitized carbon	0.25	100	5	46	Liquid chromatography
24	28						1	
25	30	Hypercarb	porous graphitized carbon	0.75	100	3	6	Hypercarb Kappa
26	31	Custom made Hypercarb					46	
27	32	Custom made Hypercarb					46	
28	34	Custom made Hypercarb					46	
29	38		Porous graphitized carbon	250 um	10 cm	5 um	46	
30	39						1	
31	40						1	
32	41	Custom made Hypercarb					46	
33	42	Hypercarb	Porous graphitized carbon	0.25	100	5	46	Reversed phase
34	43	Hypercarb	Porous graphitized carbon	0.25	100	5	46	Reversed phase
35	45	Hypercarb	Carbon	0,75	100	3	6	Reverse phase
36	46	Hypercarb	Carbon	0,75	100	3	6	Reverse phase
37	47	Hypercarb	Carbon	0,75	100	3	6	Reverse phase
38	48	Hypercarb	Porous graphitized carbon	0.25	100	5	46	Reversed phase
39	49	Hypercarb	Porous graphitized carbon	0.25	100	5	46	Reversed phase
40	50						1	

FIGURE 2.15: Actual table that stores column information used in HPLC experiments in the UniCarb-DR database. Each row in this table is a record representing an entry in one column used in the experiment, and each column in the table represents its attributes. Due to the mixing of empty strings and "null" values in the table to indicate the absence of data, the error checking of the previous UniCarb-DR server-side program had resulted in an error-prone. This screenshot was generated using DBeaver 23.3.1 Community edition.

To fundamentally solve these inefficient issues, I adopted `Slick`[10] as a library for accessing relational databases from the Scala programming language. `Slick` is a functional programming-friendly library, and its feature called `Slick code generator`[11] can exhaustively collect the names and type information of the myriad tables and their attributes in a database, and automatically generate appropriate data type definitions for programs written in the Scala language to access the database based on that information. Using this feature, when an attribute in a table that may contain a "null" value is retrieved from the database using the Scala language, its data type will automatically be set to the `Option` type, allowing efficient development while benefiting from static type checking by the type checker. For example, since the `type_chromatography` attribute in the above table contains some "null" values, if this attribute is retrieved using `Slick`, the data type of this data is automatically

[10] https://scala-slick.org
[11] https://scala-slick.org/doc/3.2.0/code-generation.html

considered as the `Option[String]` type in the Scala language. For `Option[String]` type values, it is easy to convert the value to an empty string if it is a `None` value and to retrieve the actual string if it is a `Some` value. This eliminates the problem of having to double-check for errors when the result is a "null" value or an empty string. Furthermore, if the definitions of tables or attributes in the database have been changed, by simply re-running `Slick code generator`, the latest data type information in the database can be retrieved and reflected in the code base of the Scala program. This fundamentally avoids the problem of mismatch between server-side source code and table definitions in the database.

When a programmer attempts to retrieve a string in the database that may contain a "null" value using `Slick`, a value of data type `DBIOAction[Option[String], NoStream, Effect.Read]` will be generated. This means that `Slick` will now access the database and read the data (`Effect.Read`), resulting in a value of type `Option[String]`. In fact, the `map` function is defined for this data type as well as the list and `Option` types. By using this function, it is possible to write a conversion operation for a value of type `Option[String]` obtained from a database while preserving the context of "this value is obtained by reading from a database" as the data type information. This allows the programmer to interpret the contextual information that the value was generated depending on some read operation from the database by looking at the data type of the value when reading this source code. Thus, by applying functional programming techniques, programmers can easily read detailed information about the program's behavior without actually executing the it. As a result, it is expected that the future development of UniCarb-DR will be even more productive than before and the system can be used as a more reliable system.

2.4.2 Design of unified data submission flow between UniCarb-DR and Glyco-POST

There was an issue between UniCarb-DR and GlycoPOST, as submitting data separately would have resulted in inconsistent cross-referencing between them. Additionally, in fact, the size of GlycoWorkbench files submitted to UniCarb-DR greatly varies from file to file depending on the amount of stored data. Owing to this fact, when researchers try to register large files to UniCarb-DR, the upload process sometimes fails due to timeouts. To solve these issues, it is necessary to establish a unified new data submission flow for the two data repositories. Fortunately, GlycoPOST implements the public APIs to programmatically retrieve various information about a submission that has already been published. With the public APIs, the data contained in the submission can also be downloaded programmatically. Therefore, if a submission published on GlycoPOST contains GlycoWorkbench files that UniCarb-DR can interpret, a program that automatically downloads the files and registers them to UniCarb-DR can be periodically executed on the on-premise server every day as a batch process. By running this program, the identification results of mass spectrometry experiments included in newly published submissions on GlycoPOST can be automatically registered to UniCarb-DR and their contents can be visualized with it. That is, it is possible to achieve unification of data submission flows to both data repositories by allowing researchers to submit all data, including GlycoWorkbench files, to GlycoPOST instead of UniCarb-DR. The beauty of this mechanism is that all of the previous problems of UniCarb-DR can be avoided automatically. First, since GlycoPOST's public APIs only search and retrieve submissions that have already been published on GlycoPOST, it is no longer necessary to implement a new function to set the data publication date in UniCarb-DR. Furthermore, this mechanism also eliminates concerns about registration timeouts with UniCarb-DR, since UniCarb-DR downloads data from GlycoPOST rather than accepting file uploads. Consequently, we decided to withdraw the data submission form of UniCarb-DR. Figure 2.16 shows a conceptual diagram of the unified new data submission flow for both data repositories.

2.4.3 Additional MIRAGE-LC guideline support for two data repositories

The UniCarb-DR data submission form allows users to enter not only information regarding MIRAGE sample preparation guidelines and mass spectrometric analysis guidelines, but also non-MIRAGE-compliant metadata regarding HPLC experiments associated with mass spectrometry experiments. This is due to the fact that UniCarb-DR was developed at such time that the MIRAGE liquid chromatography guidelines had not yet been proposed. Therefore, if the UniCarb-DR submission form is removed, HPLC-related information that could previously be registered in UniCarb-DR will no longer be able to be registered. To solve this problem, we have updated GlycoPOST so that the metadata for liquid chromatography experiment guidelines, which were officially published in MIRAGE, can now be entered into GlycoPOST. This allows metadata that could previously be entered in UniCarb-DR to be entered in GlycoPOST, thus unifying the data submission flow without reducing the amount of metadata.

2.4.4 Visualization of GlycoWorkbench files contained in projects in embargo submitted to GlycoPOST

Furthermore, in order to visualize the identification results of MS/MS experiments in glycomics during the peer review period, we have implemented a new function called MiniCarb-Viewer in GlycoPOST,

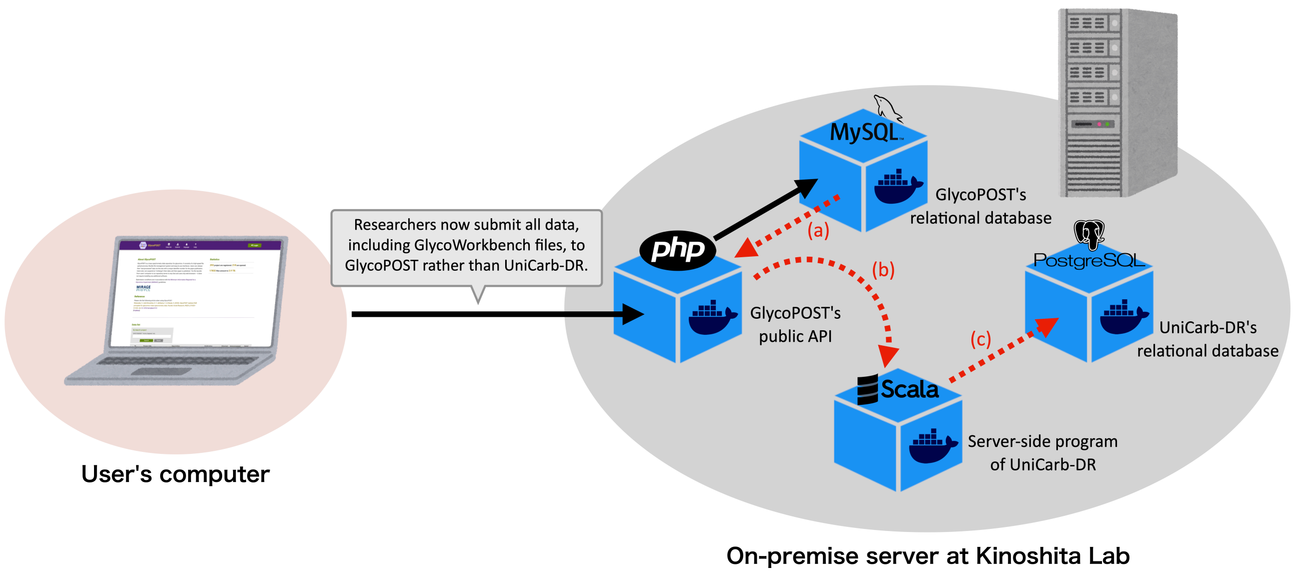
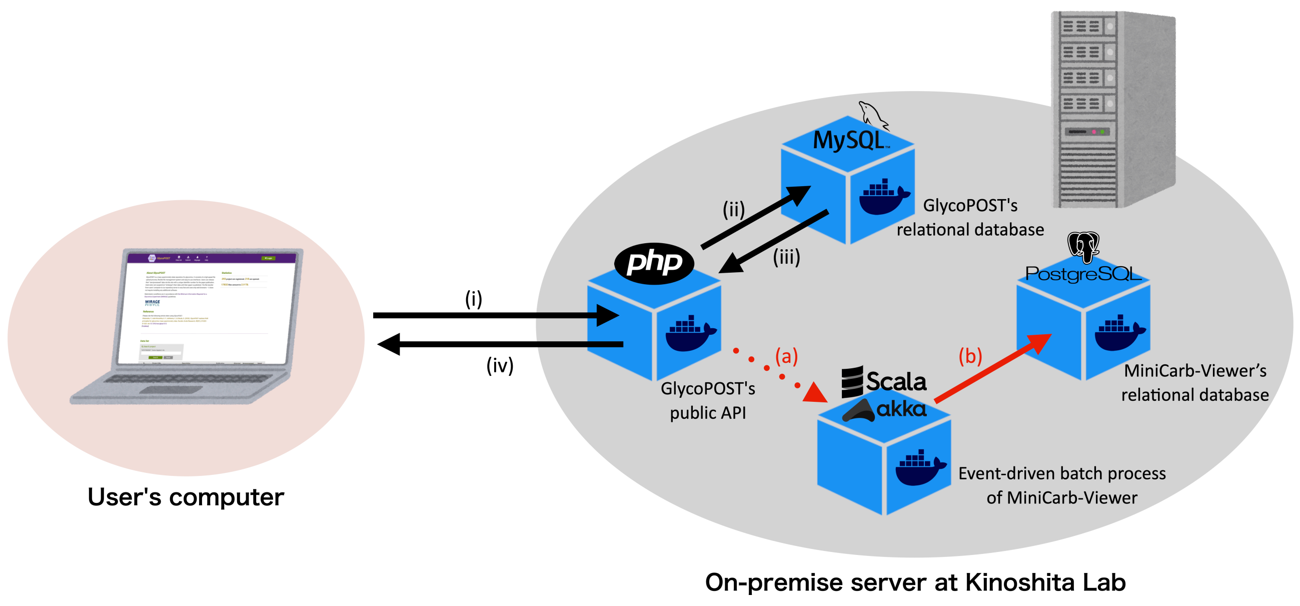

Researchers now submit all data, including GlycoWorkbench files, to GlycoPOST rather than UniCarb-DR.
User's computer
php
GlycoPOST's public API
MySQL
GlycoPOST's relational database
(a)
(b)
(c)
Scala
Server-side program of UniCarb-DR
PostgreSQL
UniCarb-DR's relational database
On-premise server at Kinoshita Lab

FIGURE 2.16: Conceptual diagram showing the overall picture of the unified new submission flow to the two data repositories. As indicated by the solid black line, all submissions from researchers, including GlycoWorkbench files, will now be submitted to GlycoPOST rather than UniCarb-DR, and these submissions will be stored in its relational database via GlycoPOST's public API. Then, after the embargo period, the data in the submissions can be retrieved from outside using GlycoPOST's public APIs. As a batch process, the server-side program of UniCarb-DR can query the public APIs of GlycoPOST every other day to fetch information about newly published submissions from the database of GlycoPOST, as shown in (a) and (b). At that time, if the newly published submission contains some GlycoWorkbench files, UniCarb-DR will store the contents in its own relational database, as shown in (c). During the construction of such a workflow, the GlycoPOST source code was modified to run within multiple Docker containers rather than within the native environment.

which enables the visualization of identification results in GlycoWorkbench files included in the submissions that are not yet publicly available in GlycoPOST by extracting the visualization function embedded in UniCarb-DR as a separate component. This enabled the reviewers to visually check the identification results contained in unpublished submissions in GlycoPOST that are referenced from papers in the peer review period. Figure 2.17 shows a conceptual diagram of the linkage between MiniCarb-Viewer and the GlycoPOST containers. Unlike UniCarb-DR, the data registered in MiniCarb-Viewer are the contents of GlycoWorkbench files contained in private submissions during the embargo period. Thus, the data stored in the MiniCarb-Viewer database needed to be reachable only for users who are logged in to GlycoPOST. To achieve this, access to MiniCarb-Viewer can only be done via the public API of GlycoPOST. One of the major potential problems is that the size of GlycoWorkbench files varies widely from file to file, as mentioned earlier, causing timeout errors when registering their contents to the MiniCarb-Viewer database. To overcome this obstacle, we focused on the implementation of event-driven batch processing, which runs automatically and asynchronously in the background for every data registration request. The logic of MiniCarb-Viewer running on the server is built on top of Play Framework (Richard-Foy, 2014), a web application framework for the Scala programming language, which is powered by Akka (Roestenburg, Williams, and Bakker, 2016), a proven asynchronous processing engine. By utilizing the Akka engine, we successfully implemented a event-driven batch process that fires with each registration request of a submission to GlycoPOST, resulting in the registration of GlycoWorkbench file contents to MiniCarb-Viewer without generating timeout errors. The registration process of the GlycoWorkbench file by this batch process is set to retry up to 10 times taking into account that some errors may occur during the execution. The waiting time between each attempt is doubled, from 5 minutes, to 10 minutes, to 20 minutes, and so on. This reduces the probability of registration process failure due to short-term server failure without consuming excessive server resources.

Moreover, with the implementation of `gws2wurcs` API (`https://api.glycosmos.org/gws2wurcs`) as an API of the GlyCosmos portal, which can obtain GlyTouCan IDs from GWS format strings, GlyTouCan IDs can now be obtained from the glycan structures in GWS format that are included in GlycoWorkbench, and this function enables collaboration between these two data repositories and GlyTouCan.



MySQL
GlycoPOST's relational database
(ii)
(iii)
php
(i)
(iv)
GlycoPOST's public API
(a)
PostgreSQL
(b)
MiniCarb-Viewer's relational database
Scala
akka
Event-driven batch process of MiniCarb-Viewer
User's computer
On-premise server at Kinoshita Lab

FIGURE 2.17: Conceptual diagram showing the collaboration between GlycoPOST containers and MiniCarb-Viewer containers. When a submission request is sent to GlycoPOST by a researcher, as shown in (i), GlycoPOST updates the status in the relational database, as shown in (ii), and then returns the result of the request acceptance, as shown in (iii) and (iv). In this process, when GlycoPOST receives the request, it will also asynchronously send a registration request to MiniCarb-Viewer, as indicated by the red dotted line (a), if the GlycoWorkbench files are included in the submission. MiniCarb-Viewer then attempts to register the contents of those files to its relational database as indicated by the solid red line (b). This process may take some time depending on the size of the files. However, since this is an asynchronous process independent of the registration request to GlycoPOST, GlycoPOST does not need to wait until it receives notification from MiniCarb-Viewer that the registration is complete. Thus, it can return its response to the researcher immediately. Compared to regular batch processes, event-driven batch processing, which is fired by such a request, has the advantage of extremely short latency until the process is started.

Chapter 3

Results

3.1 Development of a novel glycoconjugate data repository GlyComb

3.1.1 Design of input data format for GlyComb

LC-MS/MS is now the primary method used by researchers around the world for glycoproteomics experiments to determine which residues in peptide and protein amino acid sequences are attached by which glycan structures (Thaysen-Andersen and Packer, 2014). Because monosaccharides generally have multiple isomers (e.g., D-glucose, D-galactose, D-mannose, etc.), it is very difficult to assign a specific isomer to each monosaccharide in a glycan for the molecular weight observed in an experiment. Therefore, most of the results obtained from this mass spectrometry-based approach only contain the monosaccharide composition that constitutes a glycan molecule, and not the fully-defined structure of the glycan molecule (Lee et al., 2016). Figure 3.1 (a) shows a conceptual diagram of a glycopeptide using the SNFG monosaccharide symbol notation. Also, Figure 3.1 (b) shows a conceptual diagram of a glycopeptide in which the glycan structures attached to the peptide are represented by GlyTouCan IDs.

In GlyComb, we adopted a tab-separated values (TSV) format in which the amino acid sequence of a peptide, the residue numbers of glycan modifications, and the glycan structures are delimited by tab characters as the input format for submitting glycopeptide entries. For glycoprotein entry submissions, we adopted a similar TSV format, in which the UniProt ID for the protein, the attached residue numbers, and the glycan structures are delimited by tab characters. If a attached residue number is unknown, users can use the "?" character instead of the residue number. For the glycan representation, we adopted a glycan composition notation similar to that used in PMI-Byonic as one of the glycan structure notations. Moreover, to be able to accept ambiguous linkage information

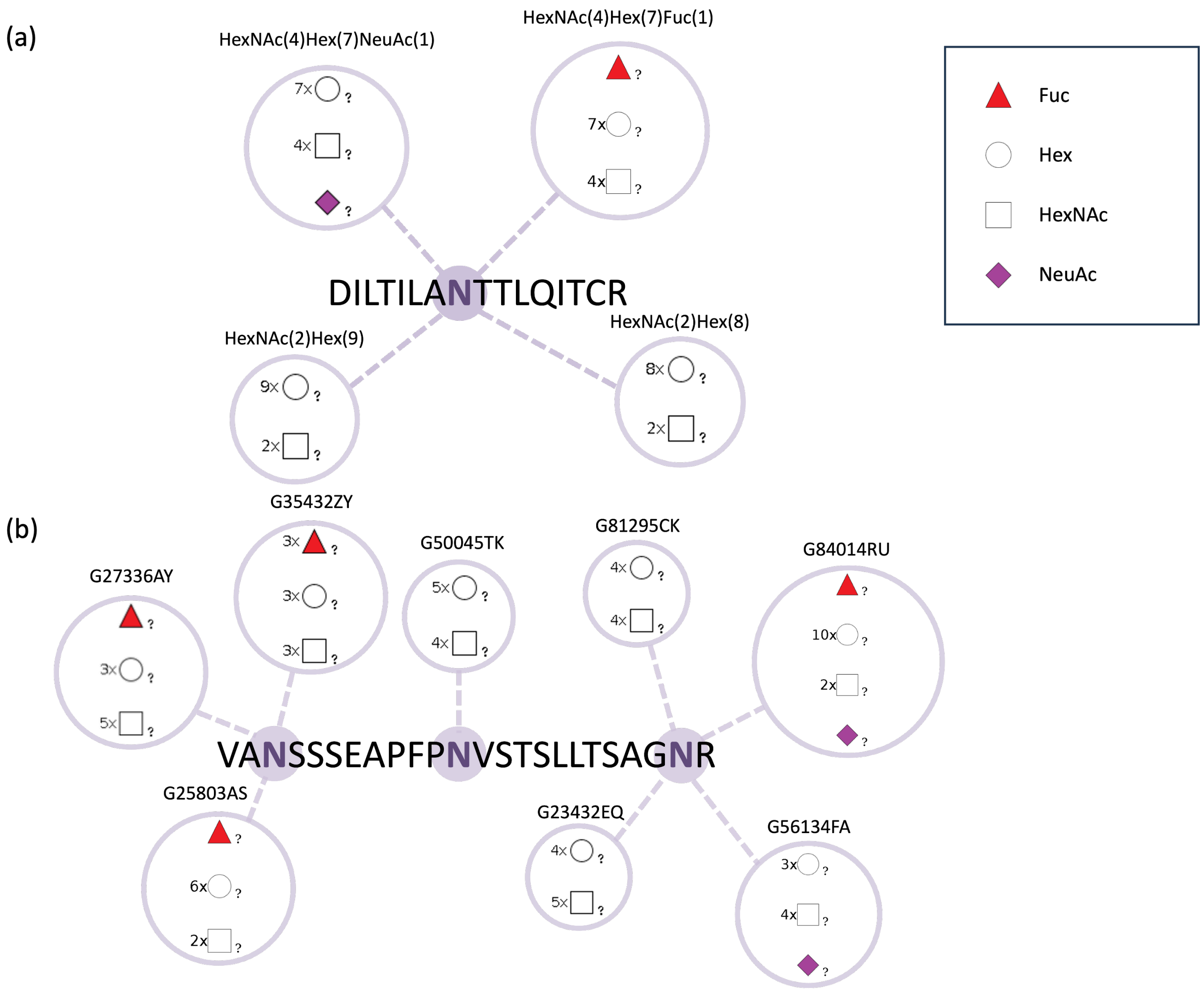


FIGURE 3.1: Conceptual diagram of glycopeptides illustrated using SNFG monosaccharide notation symbols. The "?" character assigned to each symbol indicates the linkage information (number of backbone carbon atoms participating in the covalent bond and anomeric configuration) of the monosaccharide molecule is unknown. Glycoproteomics experiments may result in multiple potential glycosylation possibilities for the same amino acid residue on a peptide obtained by protease treatment of the protein sequence with trypsin or other proteases. For each glycan structure represented symbolically, only the monosaccharide composition information is provided instead of the exact molecular structure using (a) PMI Byonic-like notation and (b) GlyTouCan IDs. In GlyTouCan, each glycan structure assigned a GlyTouCan ID is described using WURCS format, which can also express glycan composition information. Therefore, GlyTouCan IDs can also be used to represent glycan composition information.

and accurate monosaccharide composition information, GlyTouCan IDs can be used to specify the glycan structures. Thus currently, GlyComb accepts a peptide sequence or a UniProt ID and a set of related glycosylation information - that is, the number of the glycosylated residue and the glycan structure binding to it - as a valid GlyComb entry and assigns a unique GlyComb ID to it. Figure 3.2 (a) and (b) show examples of valid glycopeptide inputs to GlyComb, described using two different glycan structure notations. As shown in Figure 3.2 (b), additional UniProt ID information can be specified in the fourth column for glycopeptide entries to specify the source protein of the specified peptide sequence. Figure 3.2 (c) shows an example of a glycoprotein entry in GlyComb. Unlike glycopeptide entries, glycoprotein entries contain a UniProt ID in the first column. Since protein sequence information corresponding to a UniProt ID is occasionally updated, changed, inactivated, or deleted in UniProt, glycoprotein information submitted with a UniProt ID is subject to strict validation by referring to the exact sequence information during the registration process in GlyComb.

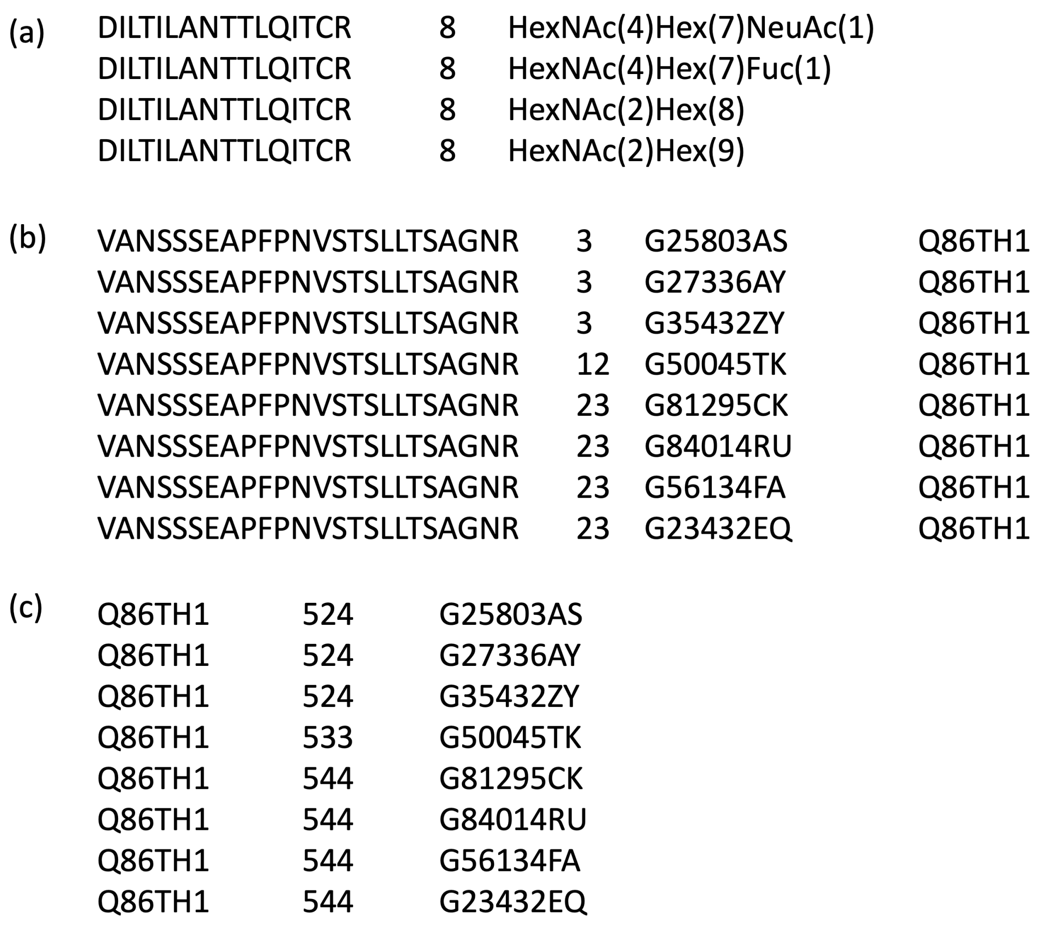
3.1.2 Overview of the GlyComb user interface

In this section, an overview of GlyComb's user interface is described, and the functions of each screen are briefly summarized.

Welcome screen and user login process

Figure 3.3 (a) shows a screenshot of the GlyComb welcome screen. By using the public APIs running on the server, the number of unique glycopeptide and glycoprotein entries registered in GlyComb is displayed on the screen as statistical information. By entering a GlyComb ID, which is the accession number of GlyComb, in the text field at the top right of the screen, users can jump to the detail screen of the corresponding entry having that GlyComb ID. In order to log in to GlyComb, users first must agree to our cookie policy by clicking the "Accept & Close" button on the pop-up window at the bottom of this screen. This cookie policy pop-up will not appear again if the user agrees to it only once. After that, users can log in to GlyComb by clicking the "Sign in with Google" button at the top right of the screen and entering their Google account email address and password.

If the login process is successful, a green pop-up window appears at the top right of the screen, as shown in Figure 3.3 (b), with a message indicating that the login is complete. However, if the login process fails for some reason, a red pop-up window with an error message, as shown in Figure 3.3 (c), will appear instead. GlyComb defines the following error codes as a way to indicate the reason for



(a)

DILTILANTTLQITCR	8	HexNAc(4)Hex(7)NeuAc(1)
DILTILANTTLQITCR	8	HexNAc(4)Hex(7)Fuc(1)
DILTILANTTLQITCR	8	HexNAc(2)Hex(8)
DILTILANTTLQITCR	8	HexNAc(2)Hex(9)

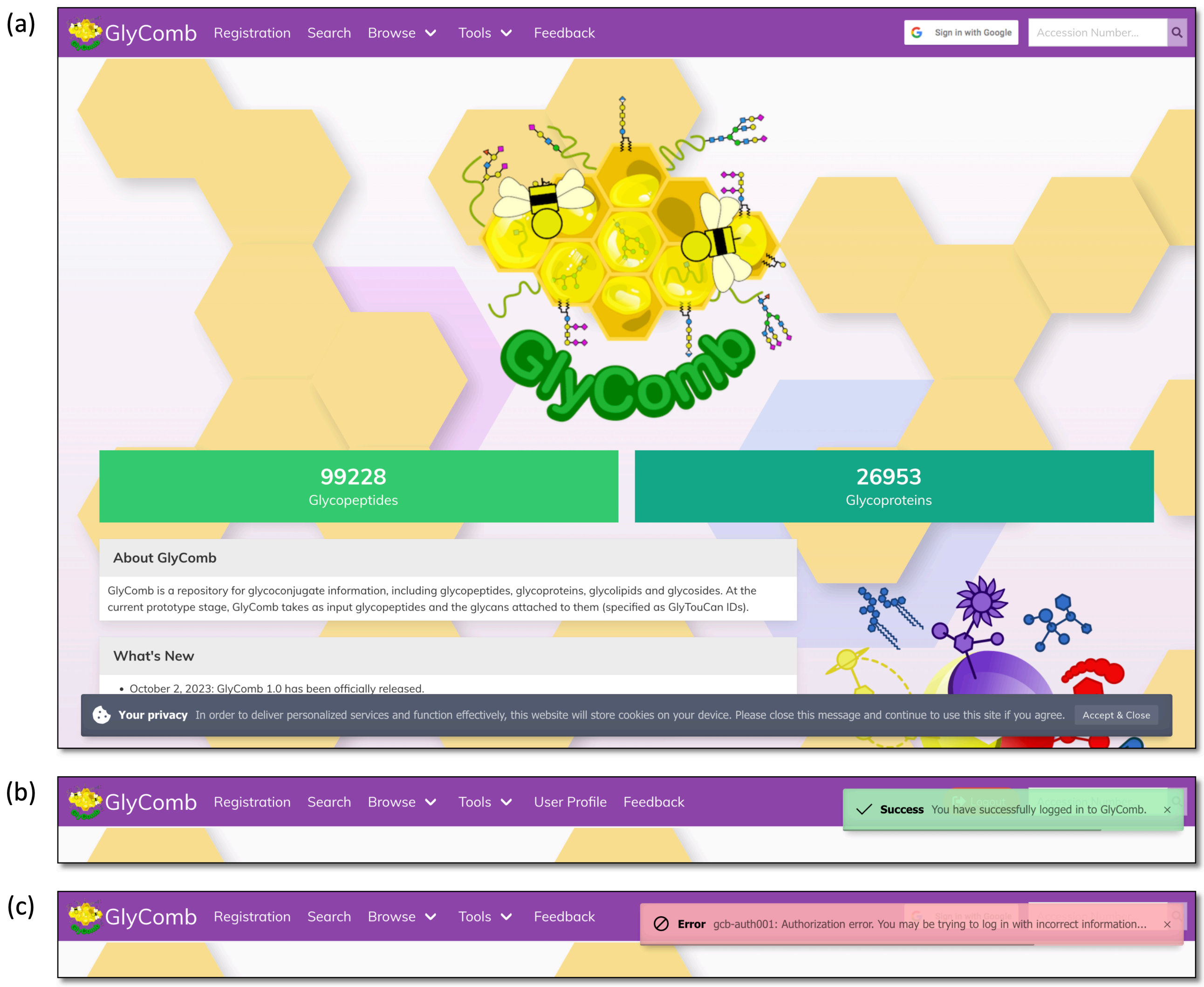
(b)

VANSSSEAPFPNVSTSLLTSAGNR	3	G25803AS	Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR	3	G27336AY	Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR	3	G35432ZY	Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR	12	G50045TK	Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR	23	G81295CK	Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR	23	G84014RU	Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR	23	G56134FA	Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR	23	G23432EQ	Q86TH1

(c)

Q86TH1	524	G25803AS
Q86TH1	524	G27336AY
Q86TH1	524	G35432ZY
Q86TH1	533	G50045TK
Q86TH1	544	G81295CK
Q86TH1	544	G84014RU
Q86TH1	544	G56134FA
Q86TH1	544	G23432EQ

Figure 3.2: (a) Example of glycopeptide input entry to GlyComb using PMI Byonic-like composition notation of glycan structures, corresponding to the conceptual diagram of the glycopeptide shown in Figure 3.1 (a). (b) Example of a glycopeptide input entry to GlyComb representing a glycopeptide described using the GlyTouCan IDs shown in Figure 3.1 (b). In the glycopeptide input, a UniProt ID can also be optionally specified in the fourth column. Detailed information on the glycan structures corresponding to each GlyTouCan ID is available at `https://glytoucan.org`. (c) Example of a glycoprotein entry to enter into GlyComb described using GlyTouCan IDs.



About GlyComb

GlyComb is a repository for glycoconjugate information, including glycopeptides, glycoproteins, glycolipids and glycosides. At the current prototype stage, GlyComb takes as input glycopeptides and the glycans attached to them (specified as GlyTouCan IDs).

What's New

- October 2, 2023: GlyComb 1.0 has been officially released.

Your privacy In order to deliver personalized services and function effectively, this website will store cookies on your device. Please close this message and continue to use this site if you agree.

Success You have successfully logged in to GlyComb.

Error gcb-auth001: Authorization error. You may be trying to log in with incorrect information...

FIGURE 3.3: Welcome screen and login process to GlyComb using Google account. (a) A screenshot of the welcome screen as of December 31, 2023. Users can log in to GlyComb with their Google account after accepting our cookie policy. (b) If the login process is successful, a green pop-up window will appear on the top right of the screen, indicating that the user has successfully logged in. (c) If an error occurs during the login process for any reason, a red pop-up window containing a message indicating the reason for the error appears instead.

Table 3.1: Error codes for errors that may occur during the login process. These error codes are shared between the user interface and server-side logic, and appropriate error messages will be displayed on the user interface when an error occurs on the server side.

Error code	Reason
gcb-auth001	Failed to retrieve user information from Google servers.
gcb-auth002	Failed to save and update user information to the relational database.

login failure and displays appropriate error messages on the user interface when an error occurs on the server (Table 3.1).

Registration screen

Figure 3.4 (a) through (d) are screenshots of the data registration screen of GlyComb. Researchers need to submit glycopeptide or glycoprotein entries to GlyComb in TSV format as described in the previous section. Entry data sent to GlyComb, either using the text area of the screen in Figure 3.4 (a) or by file upload, are first briefly checked for their format on the server, and the results are displayed on the screen in Figure 3.4 (b). Entries with no problems are marked with a green "✓", but if any errors or warnings occur, error or warning messages will appear in red or yellow, as shown in Figure 3.4 (c). Similar to the login process described earlier, GlyComb defines a list of error and warning codes that can occur in this quick input check process as Table 3.2.

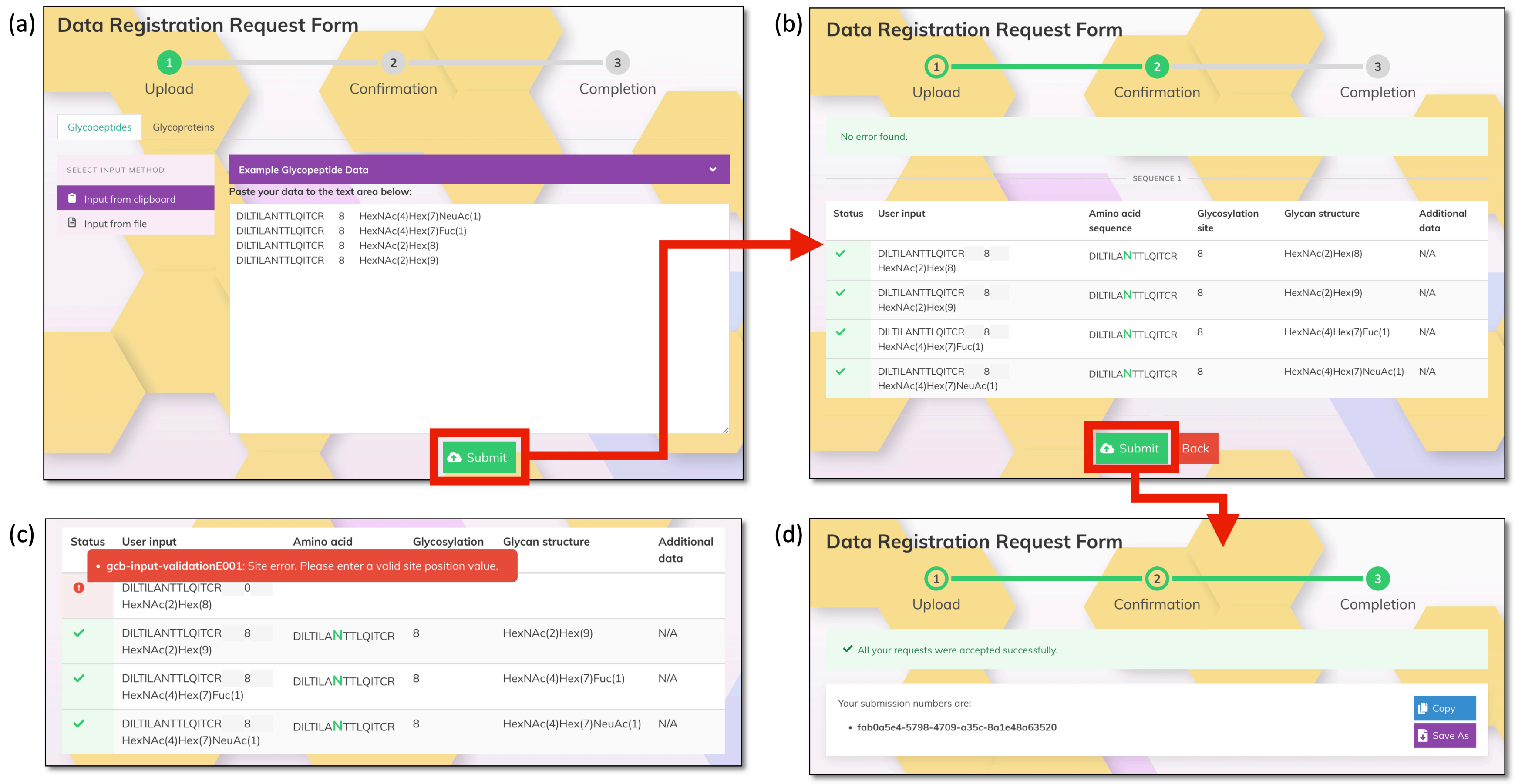
If there are no errors in Figure 3.4 (b), then the registration of the entries to GlyComb is completed by clicking the "Submit" button at the bottom of the screen as shown in Figure 3.4 (d). As mentioned previously, these registered entries are not immediately published to the public on GlyComb, rather, they need to undergo data validation in a batch process afterward.

User profile screen and submission details screen

The details of entries that a researcher previously registered to GlyComb will be listed on the user profile screen. Figure 3.5 (a) is a screenshot of the user profile screen. The contents submitted by the user previously are displayed at the bottom of the screen as a table for each glycoconjugate type. By clicking on the submission number corresponding to each submission, the user can browse the details of that submission (Figure 3.5 (b)). Submissions that have completed validation but are still unpublished can be published to the public from this screen.

Table 3.2: Error and warning codes for errors that may occur during the entry registration process. These error codes are shared between the user interface and server-side logic, and appropriate error messages will be displayed on the user interface when an error occurs on the server side. Note that the error code "gcb-input-validationE003" has been omitted here because it is already obsoleted.

Error code	**Reason**
gcb-input-validationE001	An error code indicating that the amino acid residue number bound to a glycan is out of the range of the specified amino acid sequence.
gcb-input-validationE002	An error code indicating that the number of columns in a line of the TSV file is too small.
gcb-input-validationE004	An error code indicating that a non-existent GlyTouCan ID was found in the input entry.
gcb-input-validationE005	An error code indicating that the amino acid residue number bound to a glycan is invalid.
gcb-input-validationE006	An error code indicating that an internal SPARQList API call failed inside the GlyComb server.
gcb-input-validationE007	An error code indicating that the GlyTouCan ID corresponding to the input monosaccharide composition data could not be found.
gcb-input-validationE008	An error code indicating that an internal GlyCosmos API call failed inside the GlyComb server.
gcb-input-validationW001	A warning code indicating that a duplicate line was found in the input entry.



Data Registration Request Form

Paste your data to the text area below:

```
DILTILANTTLQITCR	8	HexNAc(4)Hex(7)NeuAc(1)
DILTILANTTLQITCR	8	HexNAc(4)Hex(7)Fuc(1)
DILTILANTTLQITCR	8	HexNAc(2)Hex(8)
DILTILANTTLQITCR	8	HexNAc(2)Hex(9)
```

No error found.

Status	User input	Amino acid sequence	Glycosylation site	Glycan structure	Additional data
✓	DILTILANTTLQITCR 8 HexNAc(2)Hex(8)	DILTILANTTLQITCR	8	HexNAc(2)Hex(8)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(2)Hex(9)	DILTILANTTLQITCR	8	HexNAc(2)Hex(9)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(4)Hex(7)Fuc(1)	DILTILANTTLQITCR	8	HexNAc(4)Hex(7)Fuc(1)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(4)Hex(7)NeuAc(1)	DILTILANTTLQITCR	8	HexNAc(4)Hex(7)NeuAc(1)	N/A

• **gcb-input-validationE001**: Site error. Please enter a valid site position value.

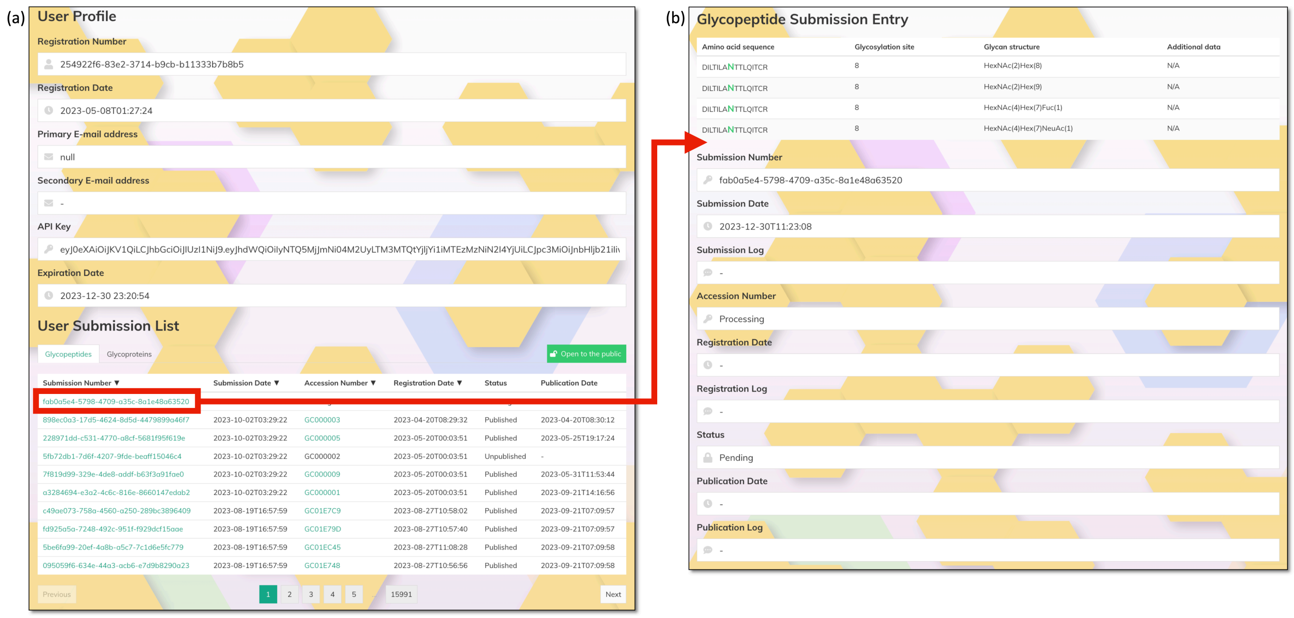
Status	User input	Amino acid	Glycosylation	Glycan structure	Additional data
!	DILTILANTTLQITCR 0 HexNAc(2)Hex(8)				
✓	DILTILANTTLQITCR 8 HexNAc(2)Hex(9)	DILTILANTTLQITCR	8	HexNAc(2)Hex(9)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(4)Hex(7)Fuc(1)	DILTILANTTLQITCR	8	HexNAc(4)Hex(7)Fuc(1)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(4)Hex(7)NeuAc(1)	DILTILANTTLQITCR	8	HexNAc(4)Hex(7)NeuAc(1)	N/A

✓ All your requests were accepted successfully.

Your submission numbers are:

- fab0a5e4-5798-4709-a35c-8a1e48a63520

FIGURE 3.4: Registration screen for glycopeptide and glycoprotein entries to GlyComb as of December 31, 2023. (a) The researchers must first select whether to register glycopeptide or glycoprotein entries by clicking either the "Glycopeptides" tab or the "Glycoproteins" tab at the top of the screen. Entry data in TSV format can then be uploaded to GlyComb by copying and pasting it into the text area, or by selecting a file to upload from the "Input from file" menu on the left hand side of the screen and then clicking the "Submit" button at the bottom of the screen. (b) The uploaded entries undergo a quick format check on the server, and then a confirmation screen will be displayed. If no problems were found in the entered entry, a green "✓" mark will appear on the left hand side of the screen. (c) If any errors or warnings occur while checking the entered entry, a message will be displayed in red or yellow indicating the reason for the error or warning. (d) If no problems are found with the uploaded entries, the registration process will be completed by clicking the "Submit" button at the bottom of the screen.



User Profile

Registration Number

254922f6-83e2-3714-b9cb-b11333b7b8b5

Registration Date

2023-05-08T01:27:24

Primary E-mail address

null

Secondary E-mail address

-

API Key

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOilyNTQ5MjJmNi04M2UyLTM3MTQtYjljYi1iMTEzMzNiN2I4YjUiLCJpc3MiOiJnbHljb21ili

Expiration Date

2023-12-30 23:20:54

User Submission List

Glycopeptides | Glycoproteins | Open to the public

Submission Number ▼	Submission Date ▼	Accession Number ▼	Registration Date ▼	Status	Publication Date
fab0a5e4-5798-4709-a35c-8a1e48a63520					
898ec0a3-17d5-4624-8d5d-4479899a46f7	2023-10-02T03:29:22	GC000003	2023-04-20T08:29:32	Published	2023-04-20T08:30:12
228971dd-c531-4770-a8cf-5681f95f619e	2023-10-02T03:29:22	GC000005	2023-05-20T00:03:51	Published	2023-05-25T19:17:24
5fb72db1-7d6f-4207-9fde-beaff15046c4	2023-10-02T03:29:22	GC000002	2023-05-20T00:03:51	Unpublished	-
7f819d99-329e-4de8-addf-b63f3a91fae0	2023-10-02T03:29:22	GC000009	2023-05-20T00:03:51	Published	2023-05-31T11:53:44
a3284694-e3a2-4c6c-816e-8660147edab2	2023-10-02T03:29:22	GC000001	2023-05-20T00:03:51	Published	2023-09-21T14:16:56
c49ae073-758a-4560-a250-289bc3896409	2023-08-19T16:57:59	GC01E7C9	2023-08-27T10:58:02	Published	2023-09-21T07:09:57
fd925a5a-7248-492c-951f-f929dcf15aae	2023-08-19T16:57:59	GC01E79D	2023-08-27T10:57:40	Published	2023-09-21T07:09:57
5be6fa99-20ef-4a8b-a5c7-7c1d6e5fc779	2023-08-19T16:57:59	GC01EC45	2023-08-27T11:08:28	Published	2023-09-21T07:09:58
095059f6-634e-44a3-acb6-e7d9b8290a23	2023-08-19T16:57:59	GC01E748	2023-08-27T10:56:56	Published	2023-09-21T07:09:58

Previous 1 2 3 4 5 … 15991 Next

Glycopeptide Submission Entry

Amino acid sequence	Glycosylation site	Glycan structure	Additional data
DILTILANTTLQITCR	8	HexNAc(2)Hex(8)	N/A
DILTILANTTLQITCR	8	HexNAc(2)Hex(9)	N/A
DILTILANTTLQITCR	8	HexNAc(4)Hex(7)Fuc(1)	N/A
DILTILANTTLQITCR	8	HexNAc(4)Hex(7)NeuAc(1)	N/A

Submission Number

fab0a5e4-5798-4709-a35c-8a1e48a63520

Submission Date

2023-12-30T11:23:08

Submission Log

-

Accession Number

Processing

Registration Date

-

Registration Log

-

Status

Pending

Publication Date

-

Publication Log

-

Figure 3.5: User profile screen and submission details screen as of December 31, 2023. (a) At the bottom of the user profile screen, a table listing the user's previous submissions is displayed. The status of each submission is as follows: "pending", a submission that has not yet passed the validation in batch processing; "unpublished", a submission that has been successfully validated; "published", a submission that has already been published to the public. (b) The submission details screen allows the researcher to review the content and, if the submission is still unpublished after the validation, to publish it to the public.

Entry list screen and entry details screen

Entries that have been registered in GlyComb, validated by batch processing, and published to the public by the researcher are freely accessible to users who have not registered as users with GlyComb. GlyComb displays the list of published glycopeptide entries and the list of published glycoprotein entries separately. Figure 3.6 is the list screen and the individual entry detail screen for published glycopeptide entries, and Figure 3.7 is those for published glycoprotein entries.

Search screen

As of December 2023, GlyComb only supports the exact match search for published entries. Figure 3.8 shows an example of the use of this search function. By entering entry data in the TSV format in the text area shown in Figure 3.8 (a) and clicking the "Search" button at the bottom of the screen, the corresponding entry detail screen will be displayed as shown in Figure 3.8 (b) if the entered entry is already registered in GlyComb. In the next release of GlyComb, we plan to greatly improve this functionality. With this update, it will be possible, for example, to search only for glycopeptide entries with a specific amino acid sequence or glycoprotein entries bound to a specific glycan structure.

PMI Byonic-derived summary spreadsheet converter screen

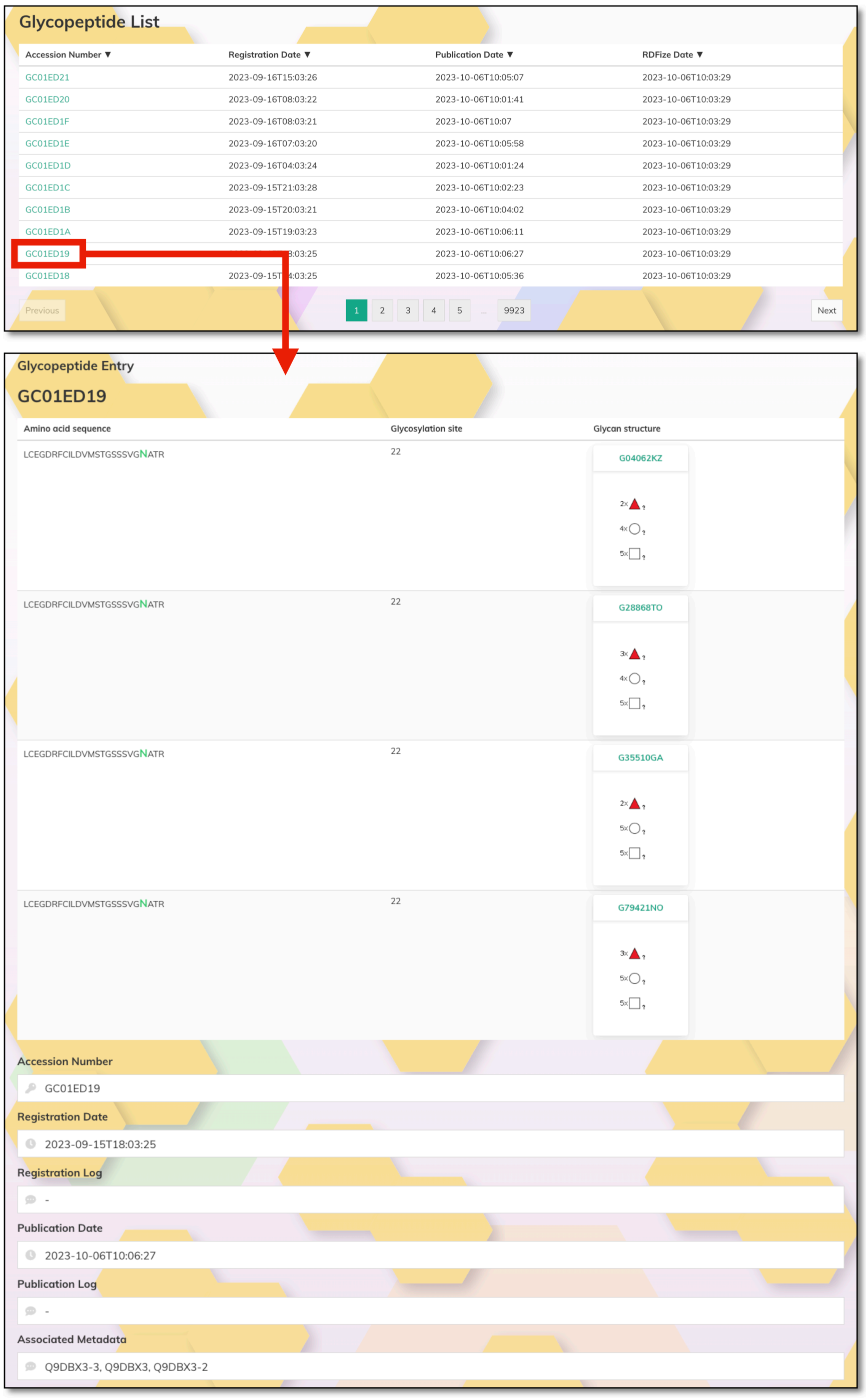
In order to facilitate the preparation of TSV files that GlyComb can interpret, GlyComb provides a function that allows researchers to extract glycopeptide and glycoprotein entries for GlyComb directly from the analysis results of glycoproteomics experiments exported from PMI-Byonic. Figure 3.9 shows an example of the use of this function. Since PMI-Byonic outputs the analysis results as a spreadsheet file compatible with MS Excel, it is possible to read this file on this screen to extract the entry data as shown in Figure 3.9 (b).

Help screen

A help screen is also provided that explains the use of GlyComb and the TSV format for entry submission in detail. Figure 3.10 is a screenshot of the help screen. In addition, I have also prepared a video that briefly explains the process of submitting entries to GlyComb and how to publish the submission, which is available on the GlyCosmos channel on YouTube (https://www.youtube.com/watch?v=-1-MUYW_wtk). This video is also embedded in the help screen.

(a)

(b)



Glycopeptide List

Accession Number ▼	Registration Date ▼	Publication Date ▼	RDFize Date ▼
GC01ED21	2023-09-16T15:03:26	2023-10-06T10:05:07	2023-10-06T10:03:29
GC01ED20	2023-09-16T08:03:22	2023-10-06T10:01:41	2023-10-06T10:03:29
GC01ED1F	2023-09-16T08:03:21	2023-10-06T10:07	2023-10-06T10:03:29
GC01ED1E	2023-09-16T07:03:20	2023-10-06T10:05:58	2023-10-06T10:03:29
GC01ED1D	2023-09-16T04:03:24	2023-10-06T10:01:24	2023-10-06T10:03:29
GC01ED1C	2023-09-15T21:03:28	2023-10-06T10:02:23	2023-10-06T10:03:29
GC01ED1B	2023-09-15T20:03:21	2023-10-06T10:04:02	2023-10-06T10:03:29
GC01ED1A	2023-09-15T19:03:23	2023-10-06T10:06:11	2023-10-06T10:03:29
GC01ED19	[illegible]3:03:25	2023-10-06T10:06:27	2023-10-06T10:03:29
GC01ED18	2023-09-15T[illegible]4:03:25	2023-10-06T10:05:36	2023-10-06T10:03:29

Glycopeptide Entry

GC01ED19

Amino acid sequence	Glycosylation site	Glycan structure
LCEGDRFCILDVMSTGSSSVGNATR	22	G04062KZ 2x▲? 4x○? 5x□?
LCEGDRFCILDVMSTGSSSVGNATR	22	G28868TO 3x▲? 4x○? 5x□?
LCEGDRFCILDVMSTGSSSVGNATR	22	G35510GA 2x▲? 5x○? 5x□?
LCEGDRFCILDVMSTGSSSVGNATR	22	G79421NO 3x▲? 5x○? 5x□?

Accession Number
GC01ED19

Registration Date
2023-09-15T18:03:25

Registration Log
-

Publication Date
2023-10-06T10:06:27

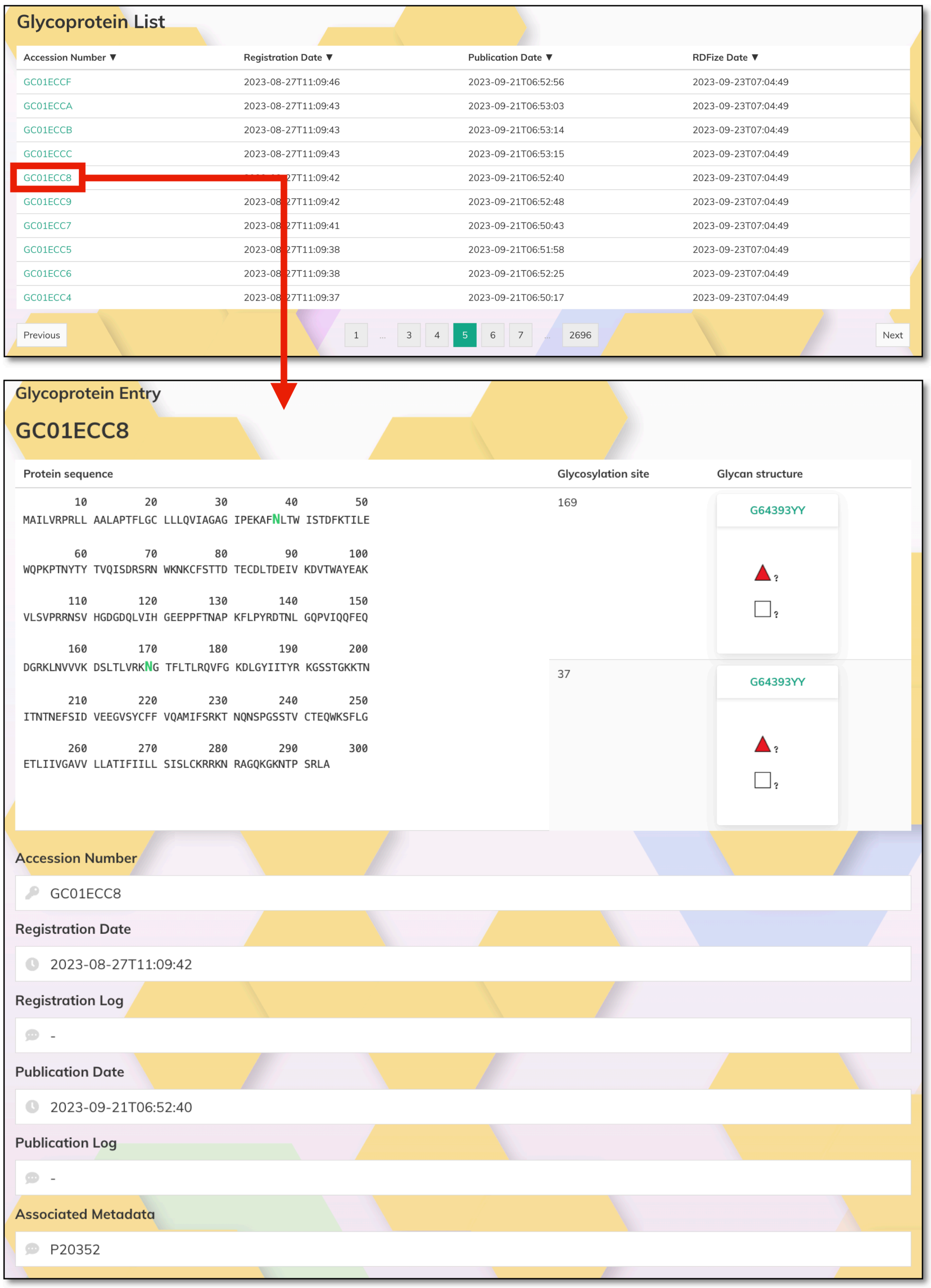
Publication Log
-

Associated Metadata
Q9DBX3-3, Q9DBX3, Q9DBX3-2

FIGURE 3.6: The list screen and the individual entry detail screen for published glycopeptide entries as of December 31, 2023. (a) The list screen of published glycopeptide entries. (b) In the detail screen of each entry, which glycan structure is bound to which residue of the amino acid sequence is displayed in a table format. Residues in the amino acid sequence bound to the glycans are highlighted in green. In addition, since all glycosylation data are normalized to a format using GlyTouCan IDs during the validation process, all glycan structures displayed here are assigned GlyTouCan IDs. At the bottom of this screen, the UniProt IDs associated with this entry are listed.

(a)

(b)



Glycoprotein List

Accession Number ▼	Registration Date ▼	Publication Date ▼	RDFize Date ▼
GC01ECCF	2023-08-27T11:09:46	2023-09-21T06:52:56	2023-09-23T07:04:49
GC01ECCA	2023-08-27T11:09:43	2023-09-21T06:53:03	2023-09-23T07:04:49
GC01ECCB	2023-08-27T11:09:43	2023-09-21T06:53:14	2023-09-23T07:04:49
GC01ECCC	2023-08-27T11:09:43	2023-09-21T06:53:15	2023-09-23T07:04:49
GC01ECC8	2023-08-27T11:09:42	2023-09-21T06:52:40	2023-09-23T07:04:49
GC01ECC9	2023-08-27T11:09:42	2023-09-21T06:52:48	2023-09-23T07:04:49
GC01ECC7	2023-08-27T11:09:41	2023-09-21T06:50:43	2023-09-23T07:04:49
GC01ECC5	2023-08-27T11:09:38	2023-09-21T06:51:58	2023-09-23T07:04:49
GC01ECC6	2023-08-27T11:09:38	2023-09-21T06:52:25	2023-09-23T07:04:49
GC01ECC4	2023-08-27T11:09:37	2023-09-21T06:50:17	2023-09-23T07:04:49

Glycoprotein Entry

GC01ECC8

Protein sequence

```
        10         20         30         40         50
MAILVRPRLL AALAPTFLGC LLLQVIAGAG IPEKAFNLTW ISTDFKTILE

        60         70         80         90        100
WQPKPTNYTY TVQISDRSRN WKNKCFSTTD TECDLTDEIV KDVTWAYEAK

       110        120        130        140        150
VLSVPRRNSV HGDGDQLVIH GEEPPFTNAP KFLPYRDTNL GQPVIQQFEQ

       160        170        180        190        200
DGRKLNVVVK DSLTLVRKNG TFLTLRQVFG KDLGYIITYR KGSSTGKKTN

       210        220        230        240        250
ITNTNEFSID VEEGVSYCFF VQAMIFSRKT NQNSPGSSTV CTEQWKSFLG

       260        270        280        290        300
ETLIIVGAVV LLATIFIILL SISLCKRRKN RAGQKGKNTP SRLA
```

Glycosylation site	Glycan structure
169	G64393YY
37	G64393YY

Accession Number

GC01ECC8

Registration Date

2023-08-27T11:09:42

Registration Log

-

Publication Date

2023-09-21T06:52:40

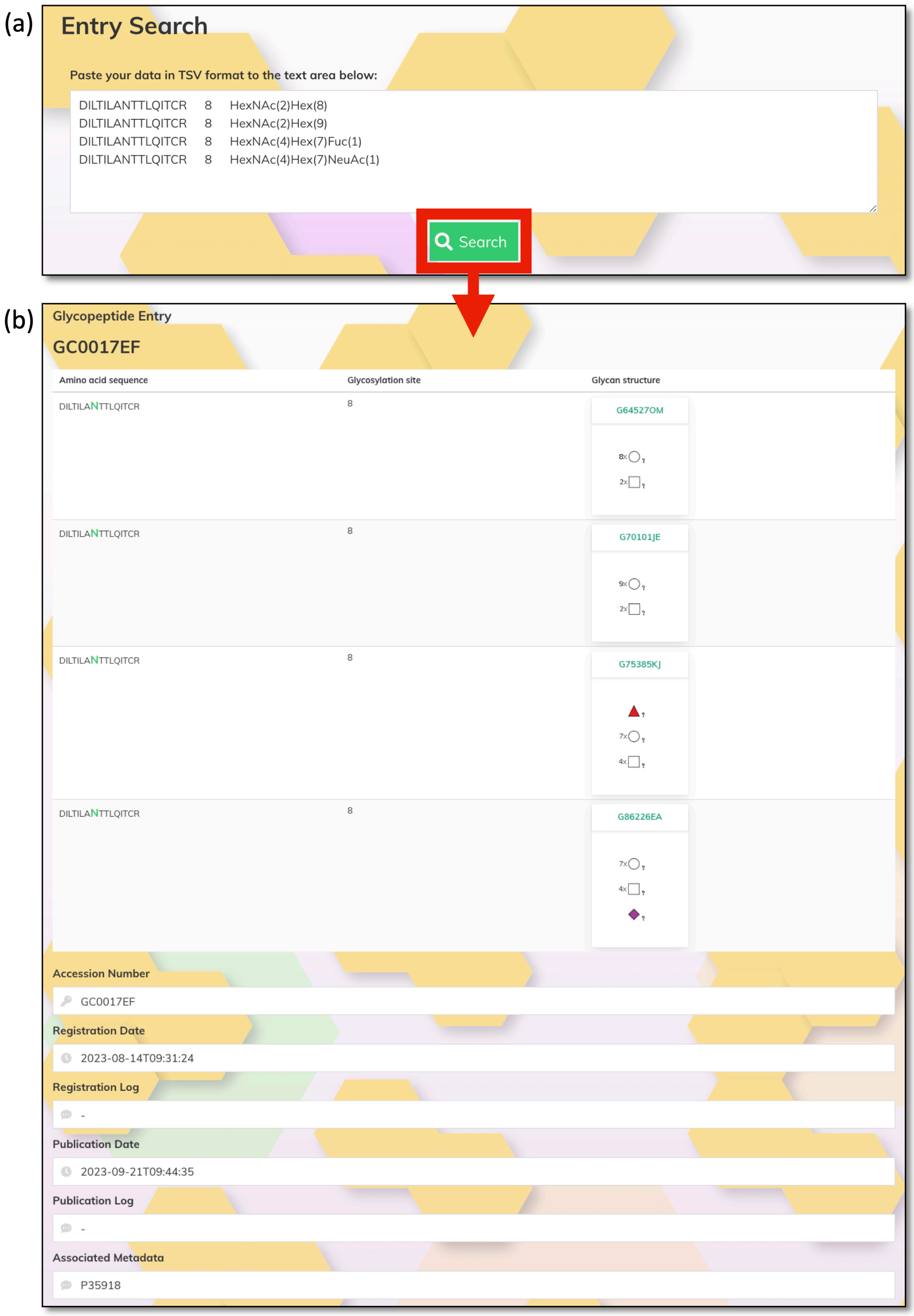
Publication Log

-

Associated Metadata

P20352

FIGURE 3.7: The list screen and the individual entry detail screen for published glycoprotein entries as of December 31, 2023. (a) The list screen of published glycoprotein entries. (b) In the detail screen of each entry, the amino acid sequence corresponding to the input UniProt ID and a set of glycosylation information are displayed. Since the amino acid sequence corresponding to a UniProt ID may change occasionally, GlyComb obtains the sequence corresponding to the input UniProt ID and stores it in the database during the validation and normalization processes performed in batch processing. This fixes the amino acid sequence data corresponding to the glycoprotein entry, thus the amino acid sequence displayed on this screen will never change.



Entry Search

Paste your data in TSV format to the text area below:

```
DILTILANTTLQITCR    8    HexNAc(2)Hex(8)
DILTILANTTLQITCR    8    HexNAc(2)Hex(9)
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)Fuc(1)
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)NeuAc(1)
```

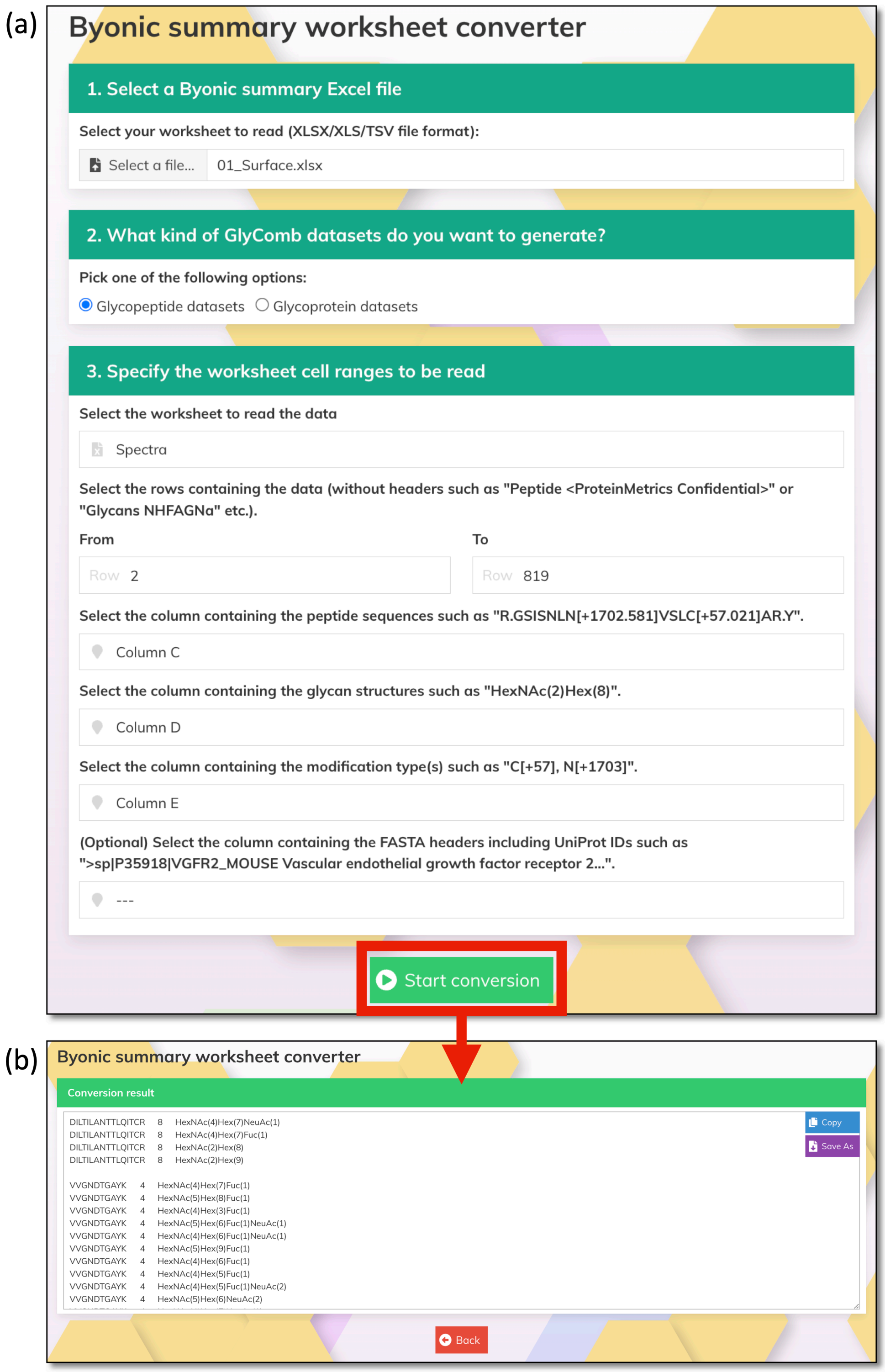
Glycopeptide Entry

GC0017EF

Amino acid sequence	Glycosylation site	Glycan structure
DILTILANTTLQITCR	8	G64527OM 8× ○? 2× □?
DILTILANTTLQITCR	8	G70101JE 9× ○? 2× □?
DILTILANTTLQITCR	8	G75385KJ ▲? 7× ○? 4× □?
DILTILANTTLQITCR	8	G86226EA 7× ○? 4× □? ◆?

Accession Number	GC0017EF
Registration Date	2023-08-14T09:31:24
Registration Log	-
Publication Date	2023-09-21T09:44:35
Publication Log	-
Associated Metadata	P35918

FIGURE 3.8: Exact match search screen for entries registered in GlyComb as of December 31, 2023. (a) By entering entry data in TSV format in the text area and clicking the “Search” button at the bottom of the screen, researchers can execute an exact match search. (b) If the entered entry has already been registered and published in GlyComb, the detail screen for that entry will be displayed.



(a)

Byonic summary worksheet converter

1. Select a Byonic summary Excel file

Select your worksheet to read (XLSX/XLS/TSV file format):

01_Surface.xlsx

2. What kind of GlyComb datasets do you want to generate?

Pick one of the following options:

[✓ Glycopeptide datasets] Glycoprotein datasets

3. Specify the worksheet cell ranges to be read

Select the worksheet to read the data

Spectra

Select the rows containing the data (without headers such as "Peptide <ProteinMetrics Confidential>" or "Glycans NHFAGNa" etc.).

From	To
Row 2	Row 819

Select the column containing the peptide sequences such as "R.GSISNLN[+1702.581]VSLC[+57.021]AR.Y".

Column C

Select the column containing the glycan structures such as "HexNAc(2)Hex(8)".

Column D

Select the column containing the modification type(s) such as "C[+57], N[+1703]".

Column E

(Optional) Select the column containing the FASTA headers including UniProt IDs such as ">sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2...".

(b)

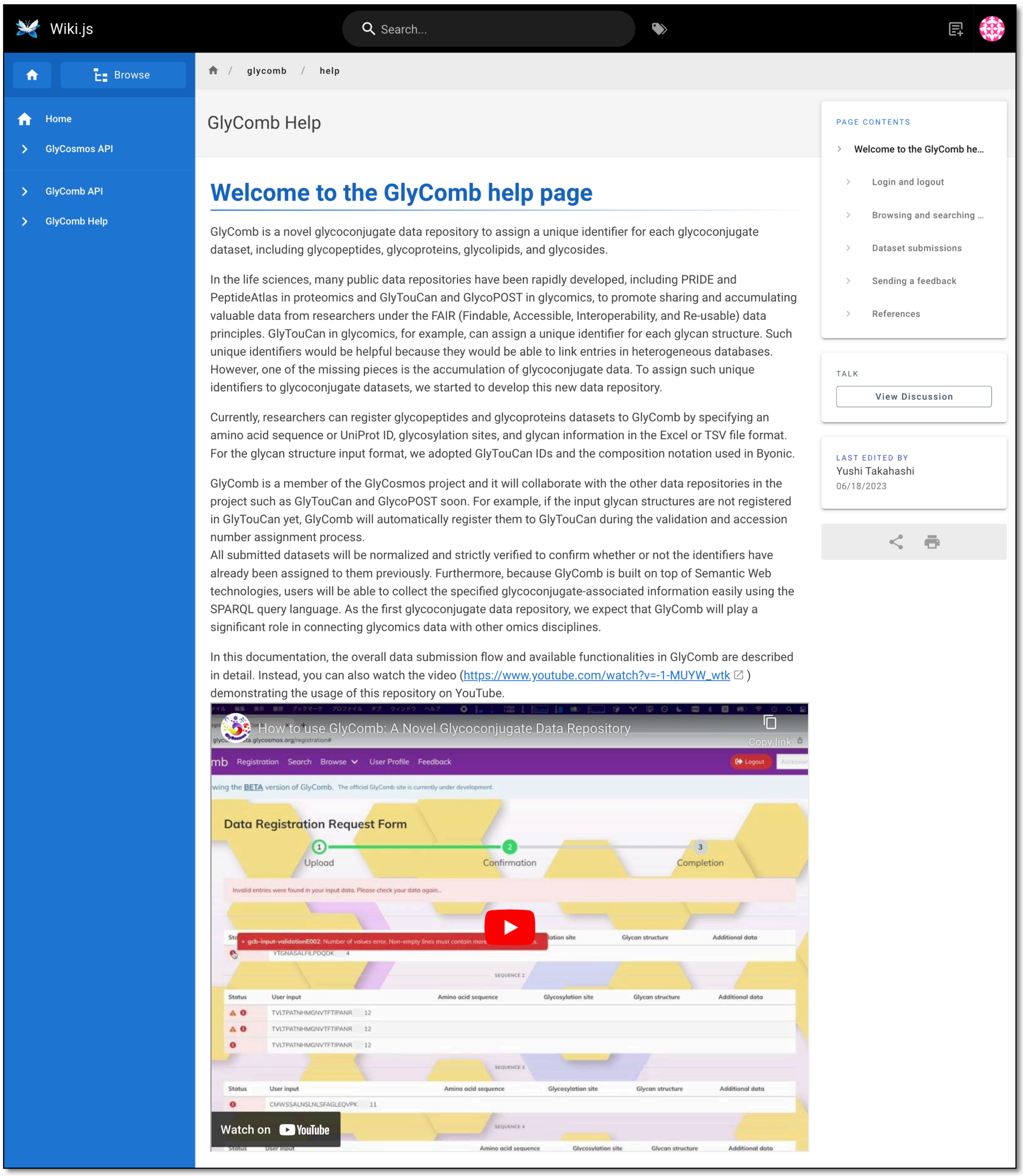
Byonic summary worksheet converter

Conversion result

```
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)NeuAc(1)
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)Fuc(1)
DILTILANTTLQITCR    8    HexNAc(2)Hex(8)
DILTILANTTLQITCR    8    HexNAc(2)Hex(9)

VVGNDTGAYK    4    HexNAc(4)Hex(7)Fuc(1)
VVGNDTGAYK    4    HexNAc(5)Hex(8)Fuc(1)
VVGNDTGAYK    4    HexNAc(4)Hex(3)Fuc(1)
VVGNDTGAYK    4    HexNAc(5)Hex(6)Fuc(1)NeuAc(1)
VVGNDTGAYK    4    HexNAc(4)Hex(6)Fuc(1)NeuAc(1)
VVGNDTGAYK    4    HexNAc(5)Hex(9)Fuc(1)
VVGNDTGAYK    4    HexNAc(4)Hex(6)Fuc(1)
VVGNDTGAYK    4    HexNAc(4)Hex(5)Fuc(1)
VVGNDTGAYK    4    HexNAc(4)Hex(5)Fuc(1)NeuAc(2)
VVGNDTGAYK    4    HexNAc(5)Hex(6)NeuAc(2)
```

FIGURE 3.9: Example usage of PMI Byonic-derived summary spreadsheet converter. (a) As described in detail later, researchers select the spreadsheet file from which they intend to extract entry data and specify the range from which to read the data on this screen. (b) The extracted entry data is displayed in the text area. Researchers can also save the extraction results as a plain text file by clicking the button displayed in this text area.



GlyComb Help

Welcome to the GlyComb help page

GlyComb is a novel glycoconjugate data repository to assign a unique identifier for each glycoconjugate dataset, including glycopeptides, glycoproteins, glycolipids, and glycosides.

In the life sciences, many public data repositories have been rapidly developed, including PRIDE and PeptideAtlas in proteomics and GlyTouCan and GlycoPOST in glycomics, to promote sharing and accumulating valuable data from researchers under the FAIR (Findable, Accessible, Interoperability, and Re-usable) data principles. GlyTouCan in glycomics, for example, can assign a unique identifier for each glycan structure. Such unique identifiers would be helpful because they would be able to link entries in heterogeneous databases. However, one of the missing pieces is the accumulation of glycoconjugate data. To assign such unique identifiers to glycoconjugate datasets, we started to develop this new data repository.

Currently, researchers can register glycopeptides and glycoproteins datasets to GlyComb by specifying an amino acid sequence or UniProt ID, glycosylation sites, and glycan information in the Excel or TSV file format. For the glycan structure input format, we adopted GlyTouCan IDs and the composition notation used in Byonic.

GlyComb is a member of the GlyCosmos project and it will collaborate with the other data repositories in the project such as GlyTouCan and GlycoPOST soon. For example, if the input glycan structures are not registered in GlyTouCan yet, GlyComb will automatically register them to GlyTouCan during the validation and accession number assignment process.
All submitted datasets will be normalized and strictly verified to confirm whether or not the identifiers have already been assigned to them previously. Furthermore, because GlyComb is built on top of Semantic Web technologies, users will be able to collect the specified glycoconjugate-associated information easily using the SPARQL query language. As the first glycoconjugate data repository, we expect that GlyComb will play a significant role in connecting glycomics data with other omics disciplines.

In this documentation, the overall data submission flow and available functionalities in GlyComb are described in detail. Instead, you can also watch the video (https://www.youtube.com/watch?v=-1-MUYW_wtk) demonstrating the usage of this repository on YouTube.

Invalid entries were found in your input data. Please check your data again...

gcb-input-validationE002: Number of values error. Non-empty lines must contain more

FIGURE 3.10: Help screen of GlyComb as of December 31, 2023. It includes detailed instructions on how to use GlyComb, such as how to log in to GlyComb and how to submit entries, etc. It also explains in detail the format of entries to be submitted to GlyComb.

3.1.3 Workflow for submitting and publishing data to GlyComb

GlyComb is currently available on the Internet, and researchers can submit their glycopeptide and glycoprotein entries in the format described in the previous section at `https://glycomb.glycosmos.org/registration` after user login with their own Google account. If they are not logged in, they can only browse the list of entries that have already been published. In addition, in order to accept input from researchers who do not have a Google account, we plan to add login methods other than Google accounts, such as identifiers issued by the Open Researcher and Contributor ID (ORCID) (Haak et al., 2012).

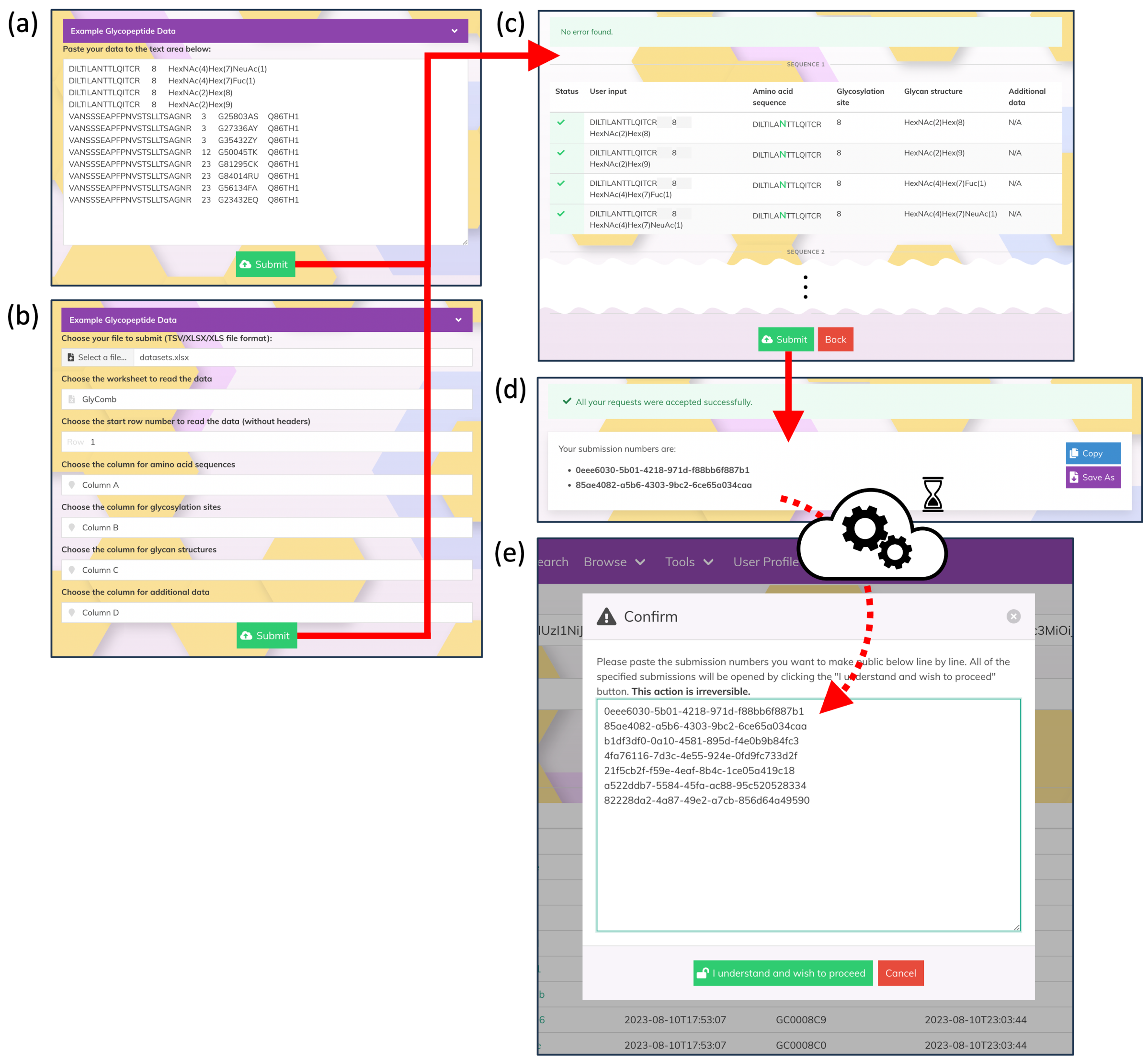
GlyComb provides two ways for researchers to submit their input entries: copy and paste input from the clipboard using the text area on the screen (Figure 3.11 (a)) or by file upload of TSV or MS Excel files (Figure 3.11 (b)). Multiple entries can be submitted at once by consecutively specifying them in the same file or text area. Blank lines can be included between the entries. Since GlyComb assigns different GlyComb IDs to different peptide sequences or UniProt IDs, GlyComb automatically distinguishes different peptide sequences and UniProt ID when multiple input entries are submitted at once. As shown in Figure 3.11 (c), a confirmation screen is displayed when submitting input data, allowing researchers to confirm their submission to GlyComb after their submission.

When input entries are submitted to GlyComb, researchers will first receive submission numbers for each input entry (Figure 3.11 (d)). These submission numbers can be downloaded and saved as a text file. Instead of immediately issuing a GlyComb ID for each registered entry, GlyComb adopts a batch validation system whereby input entries are validated together later during batch processing before accession numbers are issued. Furthermore, glycopeptide and glycoprotein entries assigned GlyComb IDs are not automatically opened to the public, and each GlyComb entry with a GlyComb ID will not be opened to the public unless the researcher who registered the entry explicitly follows the publication procedure for each entry using the submission number issued when the registration was completed. Each entry can be made public from the user profile screen, which can be accessed via the URL `https://glycomb.glycosmos.org/user_profile` after logging in to GlyComb. Figure 3.11 (e) shows the form for making each entry public. By clicking on the “Open to the public” button on the screen, an input form for entering the submission number corresponding to the entry the user wishes to make public will be displayed. By entering the submission numbers on each line in this text area and clicking on the “I understand and wish to proceed” button, the GlyComb entry publication procedure will be completed. If other researchers have already registered the exact same glycopeptide or glycoprotein entries in GlyComb, each entry will be assigned the same GlyComb ID. It is of course possible for a researcher to publish his/her own entry, which is one of those entries, even if other researchers have not published them. GlyComb guarantees that entries assigned a GlyComb ID will not have their content changed or deleted at all in the future. This means that even if researchers notice a mistake in an entry once they have registered it, they will not be able to update its content. Therefore, if there is an entry in GlyComb that contains any errors, it must be re-registered with the corrected data as a new entry in GlyComb. In fact, we assume that registering data in GlyComb is only the initial phase of the accumulation of glycoconjugate information. As a next phase, we plan to curate and annotate the deposited data and link it to existing knowledge in knowledge bases such as GlyCosmos. There, it is expected that only accurate data, or data with evidence, will be incorporated. Consequently, incorrect data will be left as unreferenced information in the GlyComb repository.

3.1.4 Automatic extraction of glycopeptide and glycoprotein entries from summary worksheets generated by PMI-Byonic

In order to assist users in uploading their glycoproteomics data into GlyComb, we decided to provide utility software to convert glycopeptide and glycoprotein identification results generated from glycoproteomics experimental analysis software tools commonly used by glycoproteomics researchers directly into a set of input entries for GlyComb. As a first step, we have implemented and embedded into the GlyComb system a conversion software to extract input entries for GlyComb from the summary spreadsheet file generated from PMI-Byonic. This software is now available at `https://glycomb.glycosmos.org/byonic-summary-worksheet-converter` and can be used in web browsers.

PMI-Byonic performs *in silico* variable modification searches of protein sequences against reference databases to identify glycopeptide sequences, and outputs sequence information with sufficient peptide fragment ion evidence as a peptide-spectrum match (PSM) into the summary spreadsheet (Bern, Kil, and Becker, 2012; Bagdonaite et al., 2022). Figure 3.12 shows the workflow of our conversion software. Figure 3.12 (a) shows an example of a summary worksheet generated by PMI-Byonic. This data was prepared by partially modifying the contents of a Byonic summary spreadsheet contained in one of the datasets published in the PRIDE Data Repository as PXD012629 (`https://www.ebi.ac.uk/pride/archive/projects/PXD012629`). Users can first select which entry to generate, glycopeptide entry or glycoprotein entry, in our conversion software. In order to generate glycopeptide entries for GlyComb, the peptide sequence information contained in column C, the attached glycan structure information contained in column D, and the type of modification information contained in column



Paste your data to the text area below:

```
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)NeuAc(1)
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)Fuc(1)
DILTILANTTLQITCR    8    HexNAc(2)Hex(8)
DILTILANTTLQITCR    8    HexNAc(2)Hex(9)
VANSSSEAPFPNVSTSLLTSAGNR   3    G25803AS   Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR   3    G27336AY   Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR   3    G35432ZY   Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR   12   G50045TK   Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR   23   G81295CK   Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR   23   G84014RU   Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR   23   G56134FA   Q86TH1
VANSSSEAPFPNVSTSLLTSAGNR   23   G23432EQ   Q86TH1
```

Choose your file to submit (TSV/XLSX/XLS file format):
Select a file... datasets.xlsx

Choose the worksheet to read the data
GlyComb

Choose the start row number to read the data (without headers)
Row 1

Choose the column for amino acid sequences
Column A

Choose the column for glycosylation sites
Column B

Choose the column for glycan structures
Column C

Choose the column for additional data
Column D

No error found.

SEQUENCE 1

Status	User input	Amino acid sequence	Glycosylation site	Glycan structure	Additional data
✓	DILTILANTTLQITCR 8 HexNAc(2)Hex(8)	DILTILANTTLQITCR	8	HexNAc(2)Hex(8)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(2)Hex(9)	DILTILANTTLQITCR	8	HexNAc(2)Hex(9)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(4)Hex(7)Fuc(1)	DILTILANTTLQITCR	8	HexNAc(4)Hex(7)Fuc(1)	N/A
✓	DILTILANTTLQITCR 8 HexNAc(4)Hex(7)NeuAc(1)	DILTILANTTLQITCR	8	HexNAc(4)Hex(7)NeuAc(1)	N/A

SEQUENCE 2

✓ All your requests were accepted successfully.

Your submission numbers are:

- 0eee6030-5b01-4218-971d-f88bb6f887b1
- 85ae4082-a5b6-4303-9bc2-6ce65a034caa

Please paste the submission numbers you want to make public below line by line. All of the specified submissions will be opened by clicking the "I understand and wish to proceed" button. **This action is irreversible.**

```
0eee6030-5b01-4218-971d-f88bb6f887b1
85ae4082-a5b6-4303-9bc2-6ce65a034caa
b1df3df0-0a10-4581-895d-f4e0b9b84fc3
4fa76116-7d3c-4e55-924e-0fd9fc733d2f
21f5cb2f-f59e-4eaf-8b4c-1ce05a419c18
a522ddb7-5584-45fa-ac88-95c520528334
82228da2-4a87-49e2-a7cb-856d64a49590
```

2023-08-10T17:53:07 GC0008C9 2023-08-10T23:03:44
2023-08-10T17:53:07 GC0008C0 2023-08-10T23:03:44

Figure 3.11: Workflow for registering and making glycopeptide entries publicly available in GlyComb. First, researchers access the entry registration screen (https://glycomb.glycosmos.org/registration) after logging into GlyComb with their Google account. They can either (a) copy and paste the entries they want to register from the clipboard into the text area on the screen, or (b) select an MS Excel file or TSV file to upload. When uploading a file, they can choose the worksheet to be read or select the columns to be read. (c) A confirmation screen to verify that the submission content is correct will be displayed. (d) GlyComb displays a submission number instead of the GlyComb ID, which is the accession number, when the submission of an input entry is completed. By using these submission numbers, researchers can later make their registered entries open to the public. (e) A confirmation screen to make each registered entry available to the public (https://glycomb.glycosmos.org/user_profile). Through batch processing on the GlyComb server, they are assigned unique GlyComb IDs within a few hours after the input entries are submitted and each entry is ready to be published. By entering the submission numbers generated when the glycopeptide or glycoprotein entries were submitted, one per line, and clicking the button at the bottom of the pop-up window, researchers can make multiple GlyComb entries they have submitted open to the public at once.

E in this worksheet are required. On the other hand, to generate a glycoprotein entry, the peptide start residue number in the amino acid sequence of the protein in column L and the protein name containing a UniProt ID in column S are additionally required.

The conversion software expects the strings contained in each column to follow the grammars, written in the extended Backus-Naur form (EBNF) (McCracken and Reilly, 2003), which is commonly used to define the grammar of programming languages. Furthermore, the parser program for the input summary spreadsheet was written in the Scala language using a parser combinator library[1] so that it could be written with a similar appearance to the grammar written using EBNF, and then converted to JavaScript using the Scala.js compiler so that it can run in web browsers. In the definition of a grammar using EBNF, the pattern that appears on the left hand side of the ::= symbol is the pattern that expands to the pattern on its right hand side. The pattern on the right-hand side contains references to other patterns and string literals surrounded by double quotation characters such as `"Hex"`, and consequently it is possible to determine whether the root pattern matches the entire input string recursively. In addition, $\{\ ptn\ \}^*$ is used to represent a pattern that matches zero or more repeats of *ptn*, $\{\ ptn\ \}^+$ is used to represent a pattern that matches one or more repeats of *ptn*, and [*ptn*] is used to represent a pattern that matches zero or only one match of *ptn*. Furthermore, $ptn1 \mid ptn2$ represents a pattern that matches either *ptn*1 or *ptn*2. Here, the following six patterns are assumed to have already been defined as primitives for simplicity.

- *uppercaseLetter* : A pattern that matches any single uppercase alphabetic character from `"A"` to `"Z"`.
- *lowercaseLetter* : A pattern that matches any single lowercase alphabetic character from `"a"` to `"z"`.
- *digit* : A pattern that matches any single character from `"0"` to `"9"`.
- *anyChar* : A pattern that matches any single character.
- *integer* : A pattern that matches any integer, such as `"12"`.
- *realNumber* : A pattern that matches a signed real number, such as `"+287.139"`.

The peptide sequence string with the mass of the modifications such as `"L.HQDIDS[+730.264][+287.139]T[+730.264][+287.139]R.G"` is expected to match the *peptide* pattern defined as follows.

[1] https://github.com/scala/scala-parser-combinators

Note that the `"."` characters at both ends of the string represent cleavages in the peptide sequence by proteases.

$$\begin{array}{lll}
\textit{peptide} & ::= & [\ \textit{anyChar}\ \texttt{"."}\ [\ \texttt{"["}\ \textit{score}\ \texttt{"]"}\ ]\ ]\ \{\ \textit{aaWithScore}\ \}^{+}\ [\ \texttt{"."}\ \textit{anyChar}\ ] \\
\textit{aaWithScore} & ::= & \textit{aa}\ \{\ \texttt{"["}\ \textit{score}\ \texttt{"]"}\ \}^{*} \\
\textit{score} & ::= & \textit{realNumber} \\
\textit{aa} & ::= & \textit{uppercaseLetter}
\end{array}$$

The glycan structure string attached to the peptide sequence is expected to match the following *glycans* pattern. Since the glycan structure strings may be described by several completely different patterns, this conversion software expects the *glycans* pattern to be one of two different patterns, *glycansPtn*1 and *glycansPtn*2. The *glycansPtn*1 pattern expects comma-separated pairs of monosaccharide composition strings and their masses, such as `"HexNAc(2)Hex(2)NeuAc(1) 287.1389,HexNAc(2)Hex(2)NeuAc(1) 287.1389"` On the other hand, the *glycansPtn*2 pattern expects a comma-separated sequence of only monosaccharide composition strings, such as `"HexNAc(2)Hex(6)Phospho(1),HexNAc(2)Hex(2)Fuc(1)"`.

$$\begin{array}{lll}
\textit{glycans} & ::= & \textit{glycansPtn}1 \\
 & | & \textit{glycansPtn}2 \\
\textit{glycansPtn}1 & ::= & \textit{glycan}\ \{\ \texttt{","}\ \textit{glycan}\ \}^{*} \\
\textit{glycan} & ::= & \textit{glycanComposition}\ \texttt{" "}\ \textit{score} \\
\textit{score} & ::= & \textit{realNumber} \\
\textit{glycansPtn}2 & ::= & \textit{glycanComposition}\ \{\ \texttt{","}\ \textit{glycanComposition}\ \}^{*} \\
\textit{glycanComposition} & ::= & \{\ \textit{composition}\ \texttt{"("}\ \textit{count}\ \texttt{")"}\ \}^{+} \\
\textit{composition} & ::= & \texttt{"HexNAc"} \\
 & | & \texttt{"Hex"} \\
 & | & \texttt{"Fuc"} \\
 & | & \texttt{"dHex"} \\
 & | & \texttt{"NeuAc"} \\
 & | & \texttt{"NeuGc"} \\
 & | & \texttt{"Phospho"} \\
 & | & \texttt{"Sulphate"} \\
\textit{count} & ::= & \textit{integer}
\end{array}$$

The type of modification data is expected to match the *modifications* pattern defined below. Similar to the glycan structure strings, this data can also be entered in several different formats. The *modificationsPtn*1 pattern can accept strings such as `"C[+57],N[+1956]"`, `"C[+57]*3,N[+1038]"`, `"C[+57],N[+714][+1794]"`, and `"T[+287]*2[+730][+1021]"` while the *modificationsPtn*2 pattern can accept strings such as `"S10(OGlycan/802.2855);M34(Oxidation/15.9949);M36(Oxidation/15.9949)"` and `"N35(NGlycan/2974.9306);M46(Oxidation/15.9949)"`.

```
modifications       ::=  modificationsPtn1
                     |   modificationsPtn2
modificationsPtn1   ::=  modPtn1 { ", " modPtn1 }*
modPtn1             ::=  uppercaseLetter { "[" score "]" [ "*" count ] }*
count               ::=  integer
modificationsPtn2   ::=  modPtn2 { "; " modPtn2 }
modPtn2             ::=  uppercaseLetter pos "(" modType " / " score ")"
pos                 ::=  integer
modType             ::=  "NGlycan"
                     |   "OGlycan"
                     |   "Oxidation"
score               ::=  realNumber
```

The format of the peptide start residue number in the amino acid sequence of the protein is simply expected to match an unsigned integer string such as `"67"`.

```
position  ::=  integer
```

Finally, the protein name strings containing UniProt IDs such as `">sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2..."` are expected to match the *proteinName* pattern defined as follows.

$$\begin{array}{lcl}
proteinName & ::= & prefix\ uniProtId\ suffix \\
prefix & ::= & anyChar\ \texttt{"|"} \\
 & | & anyChar\ prefix \\
uniProtId & ::= & \{\ idChar\ \}^{+} \\
idChar & ::= & uppercaseLetter \\
 & | & lowercaseLetter \\
 & | & digit \\
 & | & \texttt{"-"} \\
suffix & ::= & \texttt{"|"}\ \{\ anyChar\ \}^{*}
\end{array}$$

In order to determine the residue to which the glycan structure is binding, this software compares the mass information of the modification and looks for a residue whose mass information of the modification to the amino acid residue is very close (this threshold value is set to <1.0). Taking into account that some Byonic summary spreadsheets provide the mass of the glycan structure as an integer, we have determined the value of this threshold. When a worksheet is selected, a data range setting form is displayed, as shown in Figure 3.12 (b). With this form, users can choose whether to generate glycopeptide or glycoprotein entries, select the worksheets to be scanned in that spreadsheet file, and specify the columns that contain each kind of information. After filling out this form, by clicking the button at the bottom of the screen, the conversion will be executed and the conversion results will be displayed as shown in Figure 3.12 (c).

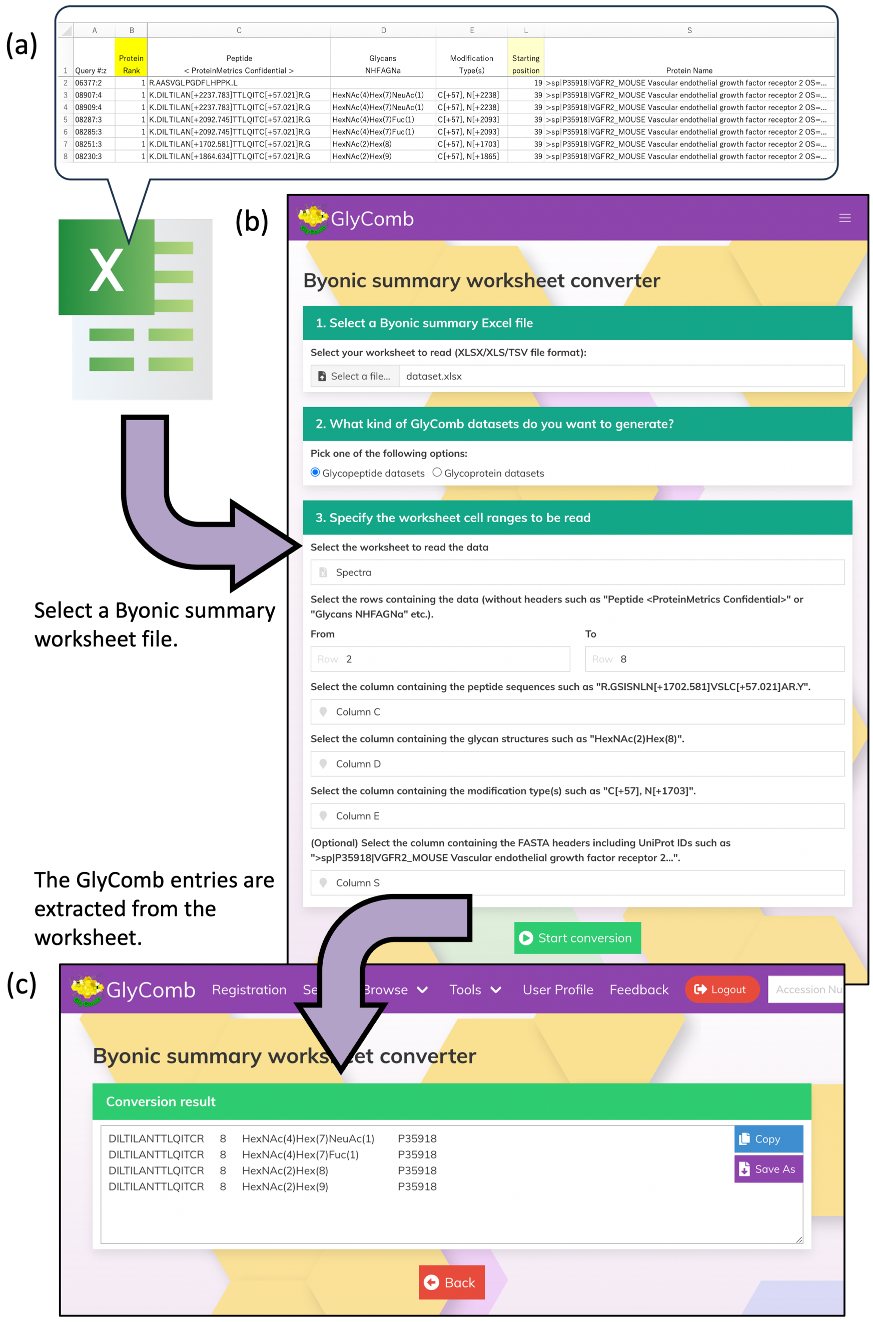
When extracting glycoprotein entries, this software automatically retrieves the amino acid sequence information corresponding to the UniProt IDs in the spreadsheet from the UniProt database. Then, the glycosylated peptide sequence information is matched with the residue number information where that peptide sequence starts on the protein to ensure that these data are consistent. This ensures that users will generate glycoprotein entries for proteins having valid UniProt IDs. As for the substituent information in glycan structures, this software currently supports only two substituents, "Phospho" and "Sulfate". This is due to the limited number of substituent types supported by the program provided by GlyCosmos for converting glycan composition strings to WURCS format, which is used by our software to verify whether the detected glycan composition strings are in a valid format. We plan to address other substituents such as "Methyl" and "Acetyl" in collaboration with the GlyCosmos development team. During the conversion process, communications with the GlyCosmos server occur to check whether the GlyTouCan IDs have already been assigned for each glycan structure. However, all other information in the spreadsheet selected by the user will not be sent to the server side. Therefore, all conversion processes are performed locally in the user's web browser.

Although PMI-Byonic is indeed a software used by a large number of glycoproteomics researchers, it is a paid software and not all researchers use it. In addition to commercial software such as PMI-Byonic and Mascot (Perkins et al., 1999), other freely available software such as Protein Prospector and GlycReSoft (Maxwell et al., 2012) are also being used to analyze the results of glycoproteomics experiments. Therefore, to encourage more researchers to submit glycoproteomics data to GlyComb, we plan to add more conversion software so that identification results generated from a wide range of glycoproteomics software can be input into GlyComb.

3.1.5 Registration of glycopeptide and glycoprotein entries extracted from the PRIDE and MS-Viewer repositories

Figure 3.13 shows a breakdown of glycopeptide and glycoprotein entries registered in GlyComb as of August 31, 2023. In order to collect glycopeptide and glycoprotein information identified from existing glycoproteomics studies available on the Internet, we surveyed a number of data repositories for proteomics experiments that are members of the ProteomeXchange Consortium. We found that identification results were uploaded as comma-separated values (CSV), TSV or MS Excel files in each contributor's own format. First, in order to extract glycopeptide and glycoprotein entries from those research results using the conversion software mentioned in the previous section, we collected PMI-Byonic-generated identification summary spreadsheets of glycoproteomics experiments from the submissions published in the PRIDE database. As a result, we obtained 1,465 PMI-Byonic-generated summary spreadsheet files from 24,145 submissions currently available in the PRIDE Archive (https://www.ebi.ac.uk/pride/archive/) as of August 8, 2023. We generated input datasets for GlyComb from these spreadsheet files using our conversion software, resulting in 95,125 unique glycopeptide entries and 24,831 unique glycoprotein entries.

In addition, we examined 683 entries corresponding to MS-Viewer Keys published in the MS-Viewer repository (https://msviewer.ucsf.edu/prospector/cgi-bin/msform.cgi?form=msviewrep), which is built into Protein Prospector, a suite of proteomics experimental analysis software and database. As a result, we successfully obtained a total of 3,189 glycopeptide entries that can be entered into GlyComb and 1,096 glycoprotein entries that can be entered into GlyComb out of 19 MS-Viewer entries. The glycosylation information published in the MS-Viewer repository included ambiguous data that had multiple candidates for the amino acid residue number to be modified in the peptide sequence,


(a)
(b)
(c)
Select a Byonic summary worksheet file.
The GlyComb entries are extracted from the worksheet.

	A	B	C	D	E	L	S
1	Query #:z	Protein Rank	Peptide < ProteinMetrics Confidential >	Glycans NHFAGNa	Modification Type(s)	Starting position	Protein Name
2	06377:2	1	R.AASVGLPGDFLHPPK.L			19	>sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2 OS=...
3	08907:4	1	K.DILTILAN[+2237.783]TTLQITC[+57.021]R.G	HexNAc(4)Hex(7)NeuAc(1)	C[+57], N[+2238]	39	>sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2 OS=...
4	08909:4	1	K.DILTILAN[+2237.783]TTLQITC[+57.021]R.G	HexNAc(4)Hex(7)NeuAc(1)	C[+57], N[+2238]	39	>sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2 OS=...
5	08287:3	1	K.DILTILAN[+2092.745]TTLQITC[+57.021]R.G	HexNAc(4)Hex(7)Fuc(1)	C[+57], N[+2093]	39	>sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2 OS=...
6	08285:3	1	K.DILTILAN[+2092.745]TTLQITC[+57.021]R.G	HexNAc(4)Hex(7)Fuc(1)	C[+57], N[+2093]	39	>sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2 OS=...
7	08251:3	1	K.DILTILAN[+1702.581]TTLQITC[+57.021]R.G	HexNAc(2)Hex(8)	C[+57], N[+1703]	39	>sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2 OS=...
8	08230:3	1	K.DILTILAN[+1864.634]TTLQITC[+57.021]R.G	HexNAc(2)Hex(9)	C[+57], N[+1865]	39	>sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2 OS=...

GlyComb

Byonic summary worksheet converter

1. Select a Byonic summary Excel file

Select your worksheet to read (XLSX/XLS/TSV file format):

Select a file... dataset.xlsx

2. What kind of GlyComb datasets do you want to generate?

Pick one of the following options:

[✓ Glycopeptide datasets] Glycoprotein datasets

3. Specify the worksheet cell ranges to be read

Select the worksheet to read the data

Spectra

Select the rows containing the data (without headers such as "Peptide <ProteinMetrics Confidential>" or "Glycans NHFAGNa" etc.).

From: Row 2 To: Row 8

Select the column containing the peptide sequences such as "R.GSISNLN[+1702.581]VSLC[+57.021]AR.Y".

Column C

Select the column containing the glycan structures such as "HexNAc(2)Hex(8)".

Column D

Select the column containing the modification type(s) such as "C[+57], N[+1703]".

Column E

(Optional) Select the column containing the FASTA headers including UniProt IDs such as ">sp|P35918|VGFR2_MOUSE Vascular endothelial growth factor receptor 2...".

Column S

Start conversion

GlyComb Registration Se... Browse Tools User Profile Feedback Logout

Byonic summary worksheet converter

Conversion result

```
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)NeuAc(1)    P35918
DILTILANTTLQITCR    8    HexNAc(4)Hex(7)Fuc(1)      P35918
DILTILANTTLQITCR    8    HexNAc(2)Hex(8)            P35918
DILTILANTTLQITCR    8    HexNAc(2)Hex(9)            P35918
```

Copy Save As

Back

FIGURE 3.12: Data extraction workflow from summary worksheets generated by PMI-Byonic. (a) Example summary worksheet of a glycoproteomics experiment generated from PMI-Byonic. (b) By selecting one spreadsheet file, a form for specifying whether to generate glycopeptide or glycoprotein entries and setting the range of data to be read will be displayed. (c) The conversion results of the selected spreadsheet information can be easily copied to the clipboard or saved as a text file. In this example, we specified that rows 2 through 8 in the selected worksheet are to be read, and any duplicate results of each row are automatically removed from the conversion results. In the same way, rows in the worksheet that do not have glycosylation information are not reflected in the conversion result.

as well as multiple candidates for the glycan structure to be bound. These ambiguous glycosylation data were excluded from the conversion candidates when we extracted the input entries to GlyComb. We are planning to extend the GlyComb system so that such ambiguous data can also be stored in GlyComb in the future.

After registration of all of these data, each entry was assigned a GlyComb ID. Based on the assigned IDs, we investigated how many entries extracted from the PRIDE repository overlapped with those from the MS-Viewer repository (Figure 3.14). As a result, we found that 13 glycopeptide entries and 4 glycoprotein entries overlapped. Such a comparison would have been extremely difficult to perform without these GlyComb IDs. Table 3.3 is a breakdown of these overlapped glycopeptide entries, showing the GlyComb ID of each entry, the corresponding accession numbers of the PRIDE and MS-Viewer repositories containing them, and the associated UniProt IDs.

3.2 Strengthening the integration between UniCarb-DR and GlycoPOST

3.2.1 Unifying the data submission flow and cross-referencing

As mentioned in the previous section, we first removed the data submission form from UniCarb-DR in order to unify the data submission flow between GlycoPOST and UniCarb-DR. With this change, all future additions of any data to UniCarb-DR will be made from a batch program that retrieves newly published submission data in GlycoPOST. This batch program will run once daily on the GlyCosmos server. This program obtains the submission list data published in GlycoPOST by invoking GlycoPOST's public APIs. It compares the retrieved list of all published submissions with the list of submissions already registered in the UniCarb-DR database, and obtains a list of submissions that have been newly published in GlycoPOST since the last time the batch program was executed. Then, it downloads the GlycoWorkbench files included in the list from GlycoPOST.

During this process, the project id (`GPST000128`, etc.) assigned to each GlycoPOST submission and the file id (`f_0000005579`, etc.) assigned to each file in the submission can also be obtained from the GlycoPOST API at the same time. We included these two IDs in the URL of the screen for browsing the contents of the GlycoWorkbench file contained in each submission. This allows us to determine the URL of the screen for browsing the GlycoWorkbench file in each submission to GlycoPOST as follows:

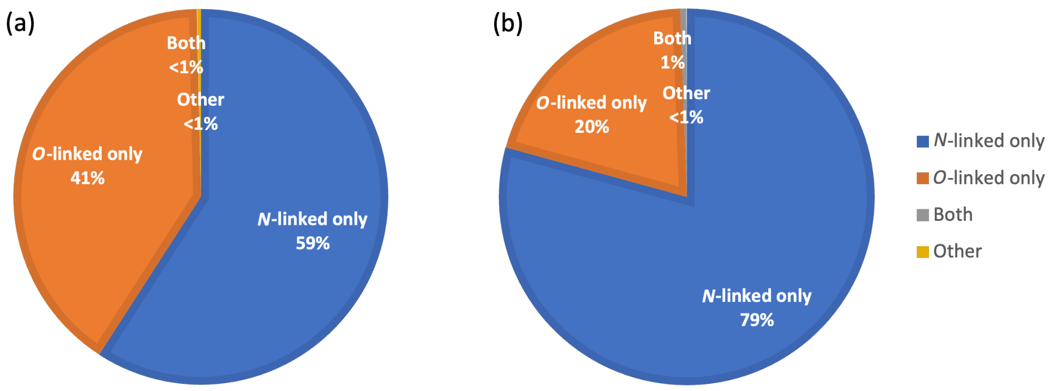


FIGURE 3.13: Percentage of (a) glycopeptide entries and (b) glycoprotein entries registered in GlyComb categorized by glycosylation class as of August 31, 2023. There are 99,240 glycopeptide entries registered in GlyComb in total, of which 59 % (58,682 entries) contain only *N*-linked glycosylation and 41 % (40,203 entries) contain only *O*-linked glycosylation. There were 74 registered glycopeptide entries containing both *N*-linked and *O*-linked glycosylation, and 281 glycopeptide entries containing neither of these. For glycoprotein entries (b), a total of 26,953 glycoprotein entries were registered in GlyComb, of which 79 % (21,359 entries) contained only *N*-linked glycosylation and 20 % (5,440 entries) contained only *O*-linked glycosylation. There are 137 registered glycoprotein entries that contain both *N*-linked and *O*-linked glycosylation, and 17 entries that are neither of these. There are 40 glycopeptide entries with UniProt IDs among the 281 glycopeptide entries having neither *N*-linked nor *O*-linked glycosylation, whereas the remaining 241 entries do not contain any UniProt ID information. Among these 40 entries, 30 entries are derived from 17 glycoprotein entries having neither *N*-linked nor *O*-linked glycosylation, while the remaining 10 entries form *N*-linked or *O*-linked glycoproteins with other glycopeptide entries. Furthermore, out of these 281 glycopeptide entries, the modified residues of 270 entries are undefined, while the modified residues of the remaining 11 entries are amino acid residues other than asparagine, serine, and threonine. Two of these entries are considered *C*-Man modified to tryptophan residues, while the remaining nine entries are modifications to other amino acid residues, including alanine, arginine, and tyrosine.

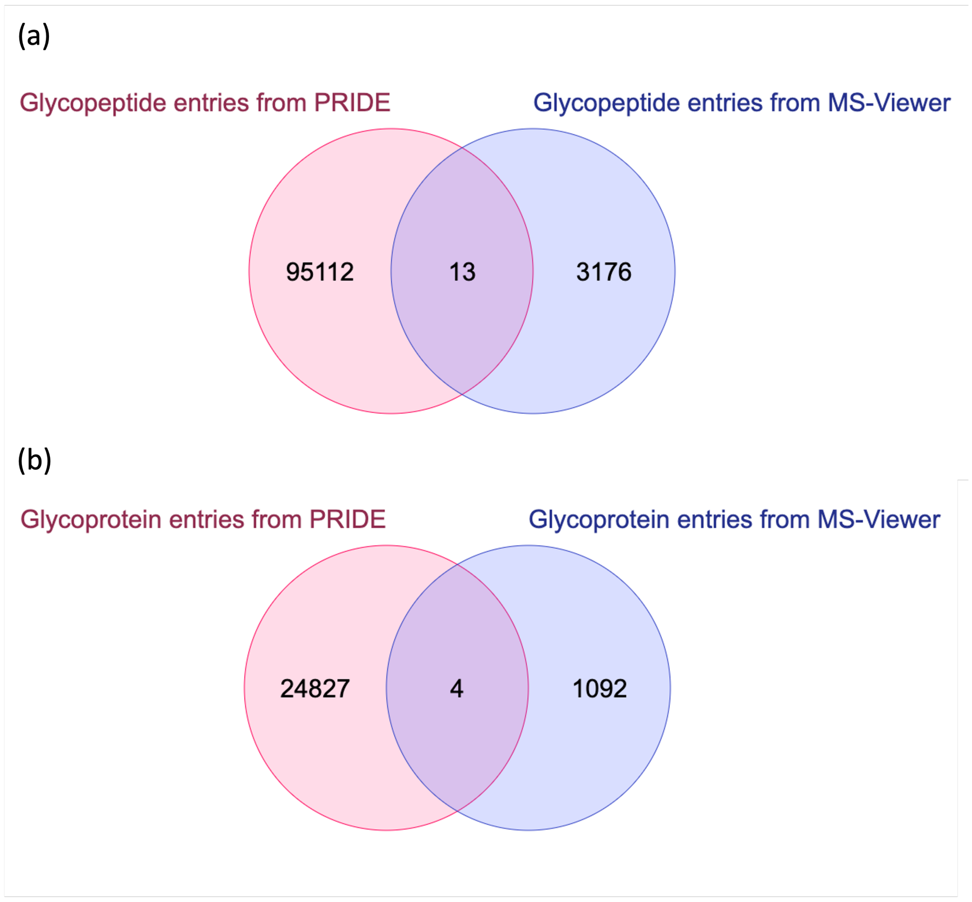
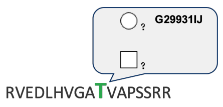
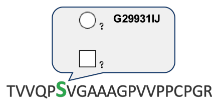
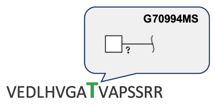
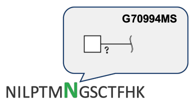
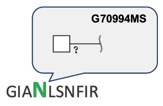
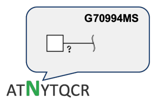
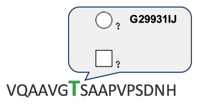
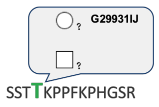
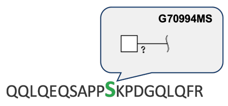
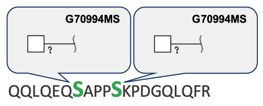
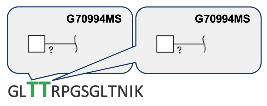
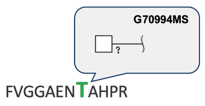
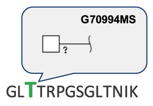


FIGURE 3.14: (a) A Venn diagram showing the number of glycopeptide entries extracted from PRIDE and MS-Viewer overlapped with each other, and (b) a Venn diagram showing the number of glycoprotein entries extracted from each repository overlapped with each other. As of August 8, 2023, 95,125 glycopeptide entries and 24,831 glycoprotein entries were extracted from the PRIDE repository and 3,189 glycopeptide entries and 1,096 glycoprotein entries were extracted from the MS-Viewer repository. By comparing the GlyComb IDs assigned to these entries extracted from both repositories, 13 glycopeptide entries and 4 glycoprotein entries were found in both.

TABLE 3.3: Breakdown of the 13 overlapping glycopeptide entries extracted from the PRIDE and MS-viewer repositories. Note that there are two GlyComb ID pairs, GC017098-GC0170AE (red pair) and GC0170B3-GC0173AC (blue pair), sharing the same peptide sequence, but they were assigned different GlyComb IDs because of the different glycosylation patterns. By assigning IDs to these glycopeptide entries, GlyComb could facilitate comparison of the results of glycoproteomics experiments submitted to different data repositories.

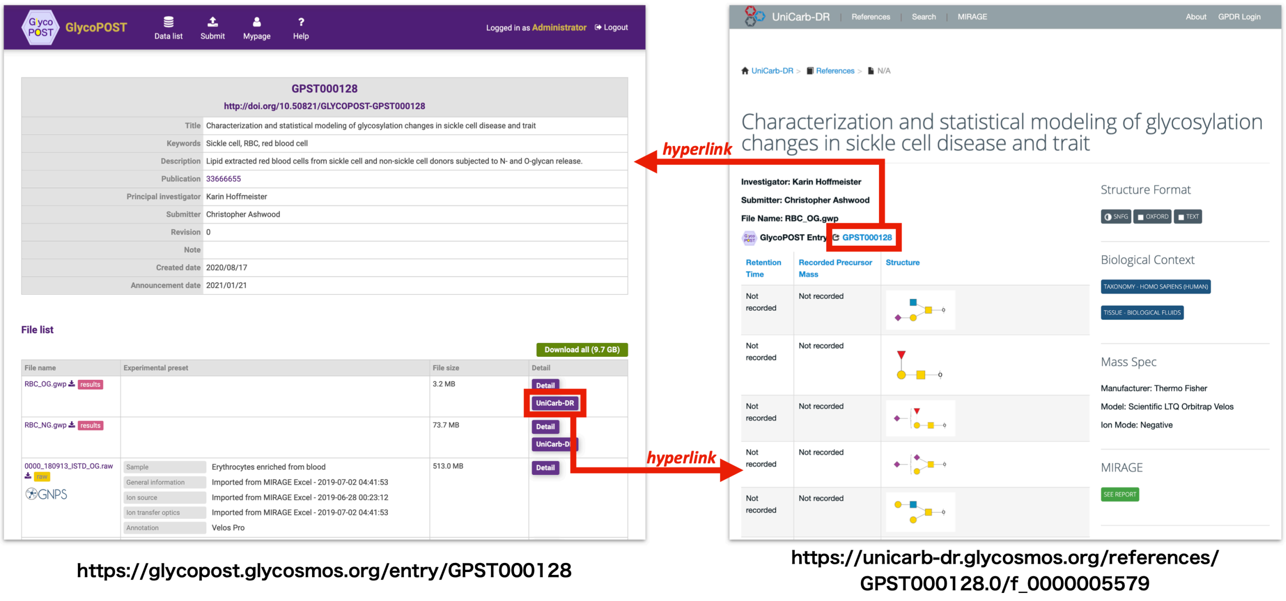
GlyComb ID	Glycopeptide Entry	PRIDE Project Accession ID	MS-Viewer Accession ID (MS-Viewer Key)	UniProt ID(s); Protein Name
GC001E29	 G29931IJ RVEDLHVGATVAPSSRR	PXD018048	PPV000097 (xjfsua8jyd)	P05154; Human plasma serine protease inhibitor
GC001E35	 G29931IJ TVVQPSVGAAAGPVVPPCPGR	PXD018048	PPV000097 (xjfsua8jyd)	P02765; Human alpha-2-HS-glycoprotein
GC001E55	 G70994MS VEDLHVGATVAPSSRR	PXD018048	PPV000108 (ix1tmqvo4h)	P05154; Human plasma serine protease inhibitor
GC00A221	 G70994MS NILPTMNGSCTFHK	PXD020196	PPV000002 (dil8ljwmkv)	Q9Z1M0, A2VCP3; Mouse P2X purinoceptor 7, Mouse P2X purinoceptor
GC00A31C	 G70994MS GIANLSNFIR	PXD020196	PPV000002 (dil8ljwmkv), PPV000046 (tbguwfn09u)	P97797; Mouse tyrosine-protein phosphatase non-receptor type substrate 1
GC00A8D9	 G70994MS ATNYTQCR	PXD020196	PPV000045 (qunwxqxpy4), PPV000094 (ncoivkfko4)	P97426; Mouse eosinophil cationic protein 1
GC00B193	 G29931IJ VQAAVGTSAAPVPSDNH	PXD020648	PPV000097 (xjfsua8jyd)	P02649; Human apolipoprotein E
GC00BCAE	 G29931IJ SSTTKPPFKPHGSR	PXD023943	PPV000097 (xjfsua8jyd)	P04196; Human histidine-rich glycoprotein
GC017098	 G70994MS QQLQEQSAPPSKPDGQLQFR	PXD035445	PPV000046 (tbguwfn09u)	Q02819; Mouse nucleobindin-1
GC0170AE	 G70994MS G70994MS QQLQEQSAPPSKPDGQLQFR	PXD035445	PPV000002 (dil8ljwmkv), PPV000045 (qunwxqxpy4)	Q02819; Mouse nucleobindin-1
GC0170B3	 G70994MS G70994MS GLTTRPGSGLTNIK	PXD035445	PPV000002 (dil8ljwmkv)	P12023, Q6GR78; Mouse amyloid-beta precursor protein, Mouse amyloid-beta A4 protein
GC017134	 G70994MS FVGGAENTAHPR	PXD035445	PPV000046 (tbguwfn09u)	P24593, Q07079; Human insulin-like growth factor-binding protein 5, Mouse insulin-like growth factor-binding protein 5
GC0173AC	 G70994MS GLTTRPGSGLTNIK	PXD035445	PPV000046 (tbguwfn09u)	P12023; Mouse amyloid-beta precursor protein

```
https://unicarb-dr.glycosmos.org/references/PROJECT_ID/FILE_ID
```

As a result, hyperlinks to corresponding files on GlycoPOST can be obtained. Conversely, by storing the project id of each GlycoPOST submission on UniCarb-DR, the URL of the corresponding GlycoPOST submission where the GlycoWorkbench file is published can be obtained as shown below:

```
https://glycopost.glycosmos.org/entry/PROJECT_ID
```

Consequently, bidirectional hyperlinks between the corresponding submissions of GlycoPOST and UniCarb-DR were obtained, thus implementing a cross-referencing function between them. Figure 3.15 shows the cross-reference relationship between GlycoPOST and UniCarb-DR. As shown in this figure, identification results included in submissions published on GlycoPOST are automatically registered in UniCarb-DR, and the two data repositories can now be cross-referenced. Users can now automatically visualize the identification results simply by submitting their raw experimental data with the identification results to GlycoPOST. By clicking on the images of each glycan structure displayed on the right hand side in Figure 3.15, each identification result will be visualized as shown in Figure 3.16 with a chart of the actual peak list.


GlycoPOST
Data list
Submit
Mypage
Help
Logged in as Administrator
Logout
hyperlink
hyperlink
UniCarb-DR
References
Search
MIRAGE
About
GPDR Login
https://glycopost.glycosmos.org/entry/GPST000128
https://unicarb-dr.glycosmos.org/references/GPST000128.0/f_0000005579

GPST000128
http://doi.org/10.50821/GLYCOPOST-GPST000128

Title	Characterization and statistical modeling of glycosylation changes in sickle cell disease and trait
Keywords	Sickle cell, RBC, red blood cell
Description	Lipid extracted red blood cells from sickle cell and non-sickle cell donors subjected to N- and O-glycan release.
Publication	33666655
Principal investigator	Karin Hoffmeister
Submitter	Christopher Ashwood
Revision	0
Note	
Created date	2020/08/17
Announcement date	2021/01/21

File list

Download all (9.7 GB)

File name	Experimental preset	File size	Detail
RBC_OG.gwp results		3.2 MB	Detail UniCarb-DR
RBC_NG.gwp results		73.7 MB	Detail UniCarb-DR
0000_180913_ISTD_OG.raw raw GNPS	Sample: Erythrocytes enriched from blood; General information: Imported from MIRAGE Excel - 2019-07-02 04:41:53; Ion source: Imported from MIRAGE Excel - 2019-06-28 00:23:12; Ion transfer optics: Imported from MIRAGE Excel - 2019-07-02 04:41:53; Annotation: Velos Pro	513.0 MB	Detail

UniCarb-DR > References > N/A

Characterization and statistical modeling of glycosylation changes in sickle cell disease and trait

Investigator: Karin Hoffmeister
Submitter: Christopher Ashwood
File Name: RBC_OG.gwp
GlycoPOST Entry GPST000128

Retention Time	Recorded Precursor Mass	Structure
Not recorded	Not recorded	
Not recorded	Not recorded	
Not recorded	Not recorded	
Not recorded	Not recorded	
Not recorded	Not recorded	

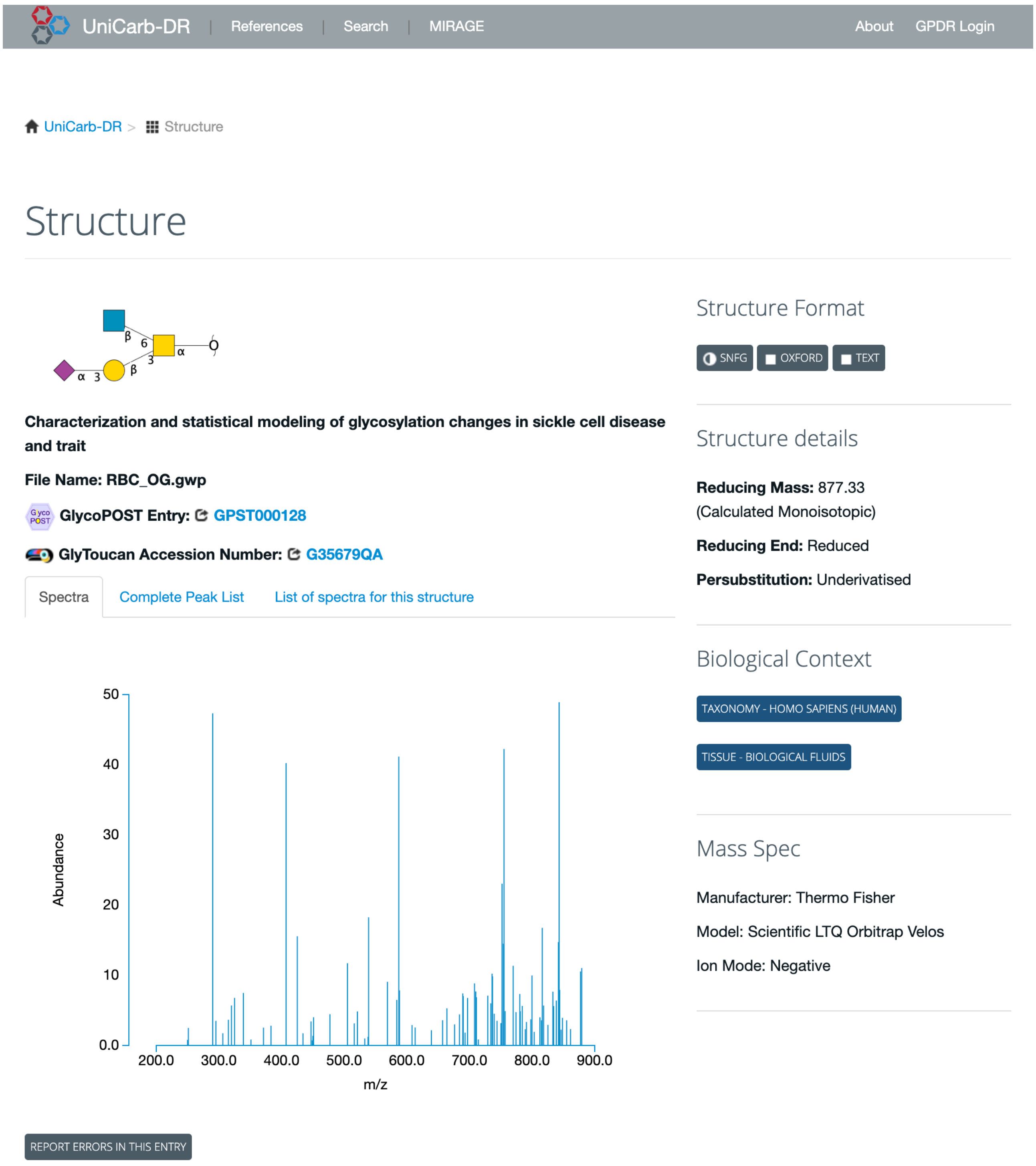
Structure Format
SNFG OXFORD TEXT

Biological Context
TAXONOMY - HOMO SAPIENS (HUMAN)
TISSUE - BIOLOGICAL FLUIDS

Mass Spec
Manufacturer: Thermo Fisher
Model: Scientific LTQ Orbitrap Velos
Ion Mode: Negative

MIRAGE
SEE REPORT

FIGURE 3.15: Cross-references between GlycoPOST and UniCarb-DR. The submission browsing screen of each data repository has a hyperlink to the corresponding entry in the other data repository, and users can freely jump between them.



Structure

Characterization and statistical modeling of glycosylation changes in sickle cell disease and trait

File Name: RBC_OG.gwp

GlycoPOST Entry: GPST000128

GlyToucan Accession Number: G35679QA

Structure Format

Structure details

Reducing Mass: 877.33 (Calculated Monoisotopic)

Reducing End: Reduced

Persubstitution: Underivatised

Biological Context

TAXONOMY - HOMO SAPIENS (HUMAN)

TISSUE - BIOLOGICAL FLUIDS

Mass Spec

Manufacturer: Thermo Fisher

Model: Scientific LTQ Orbitrap Velos

Ion Mode: Negative

FIGURE 3.16: Visualization screen of identification results obtained from GlycoPOST as of October 4, 2023. Users can now automatically visualize the identification results simply by submitting their raw experimental data with the identification results to GlycoPOST.

3.2.2 Addition of MIRAGE liquid chromatography guidelines support to GlycoPOST

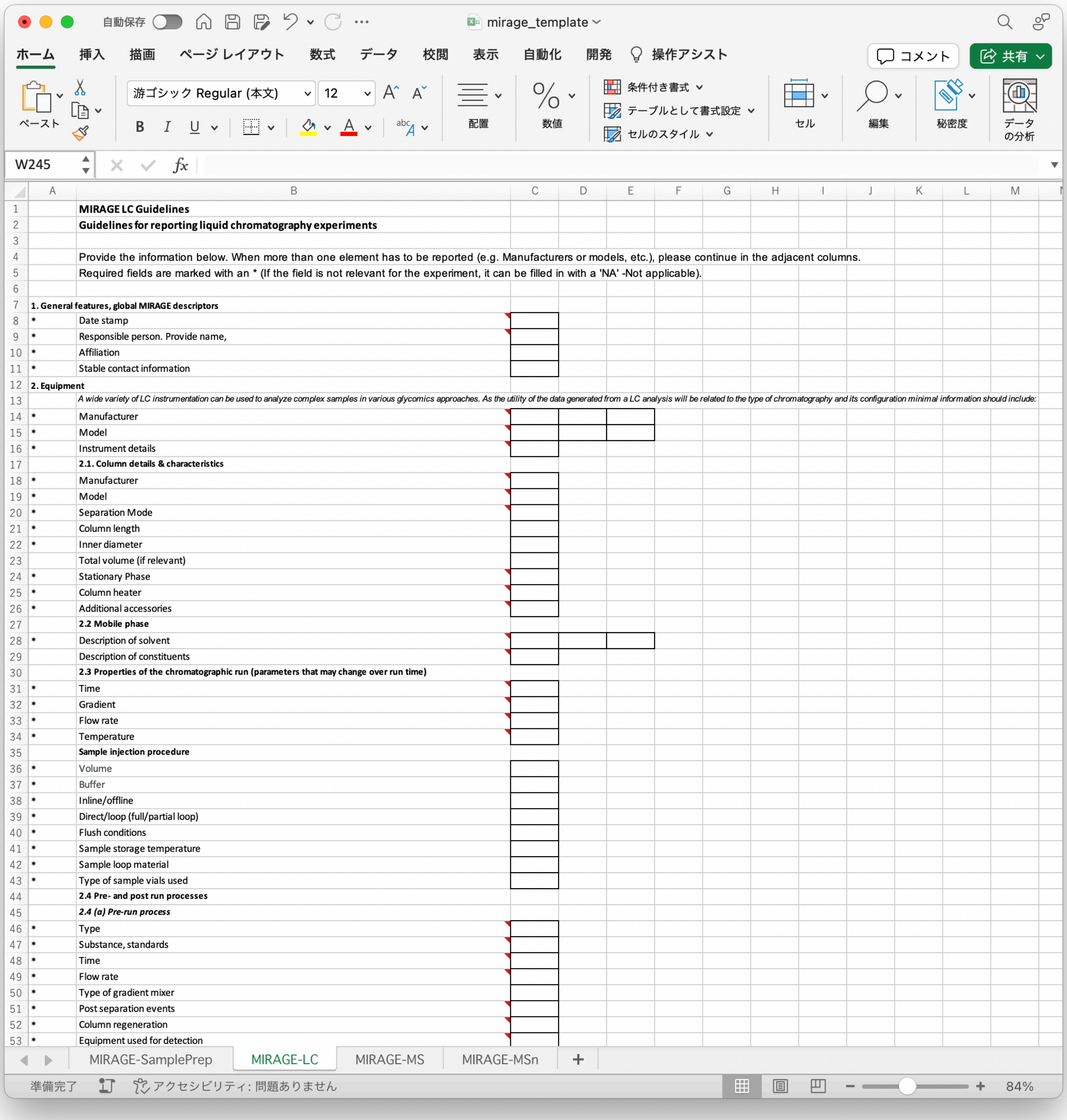
To add support for the guidelines for liquid chromatography experiments proposed in MIRAGE to GlycoPOST, we first prepare an Excel template spreadsheet that allows users to easily fill in and upload various liquid chromatography experiment information. GlycoPOST provides its own Excel template spreadsheet for inputting MIRAGE guideline information. Users can easily input experimental condition information into GlycoPOST by uploading this file to GlycoPOST after filling in it. Previously, only MIRAGE sample preparation guidelines and mass spectrometric analysis guidelines could be entered in this template file. Therefore, a new worksheet was added to this file to input liquid chromatography guidelines information. Figure 3.17 shows the worksheet for inputting MIRAGE-compliant liquid chromatography experimental guideline information. This updated Excel template file was reviewed by members of the international MIRAGE committee and adopted for input of liquid chromatography information into GlycoPOST.

Subsequently, we updated the GlycoPOST user interface, server-side program, and database information. This allows users to add liquid chromatography experiment guidelines along with the existing sample preparation guidelines and mass spectrometric analysis guidelines as metadata describing experimental information when submitting data to GlycoPOST. Figure 3.18 shows the updated GlycoPOST file submission form.

3.2.3 Visualization of GlycoWorkbench files during peer review period in GlycoPOST: MiniCarb-Viewer

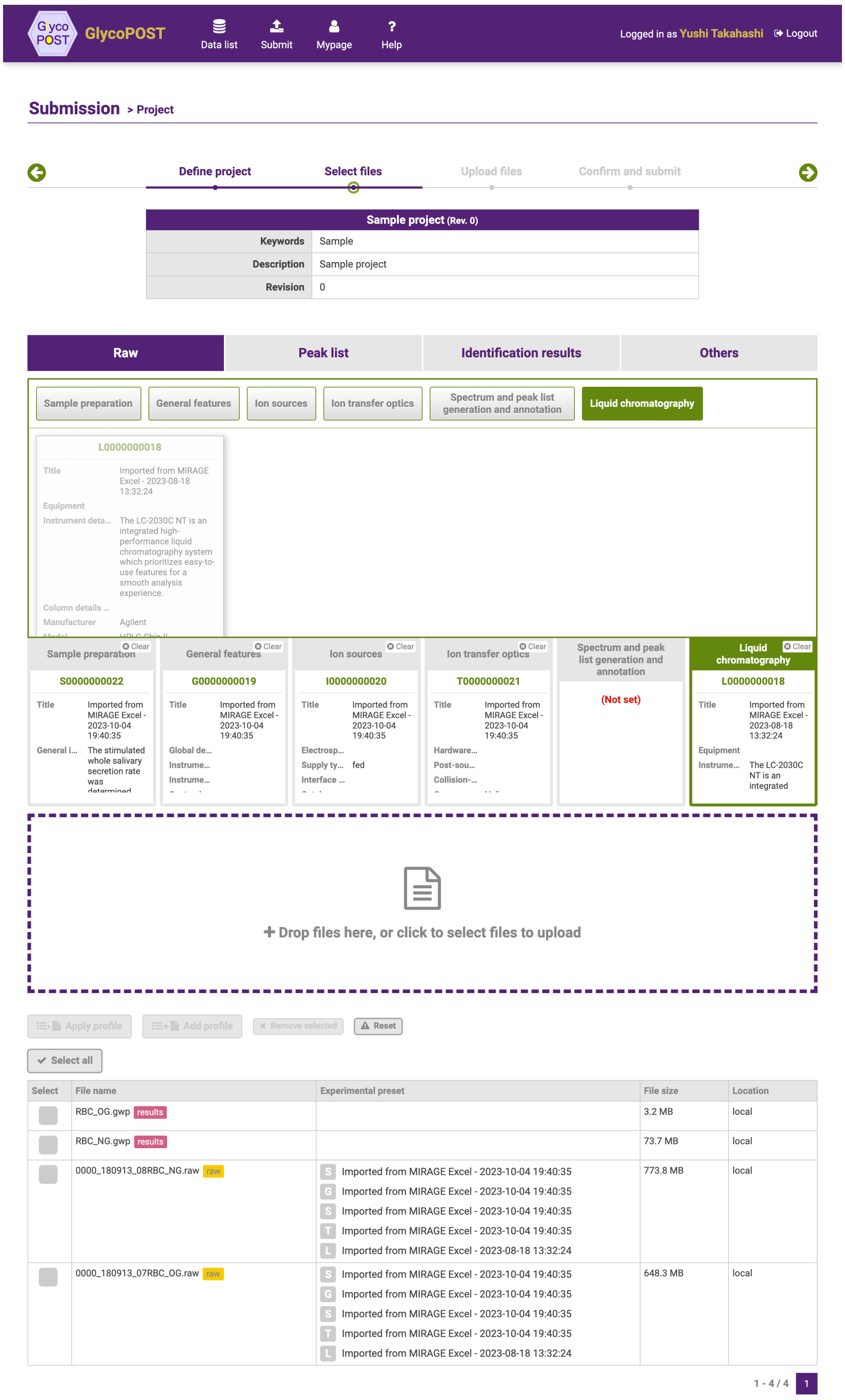
UniCarb-DR can only visualize the contents of GlycoWorkbench files included in submissions that have already been published on GlycoPOST. However, it would be useful to be able to visualize the identification results of submissions that have not yet been made publicly available on GlycoPOST for the submitter and reviewers during the peer review period. Therefore, we also addressed the visualization of identification results in GlycoPOST submissions, which are not yet publicly available.

To achieve this, we separated the visualization function of UniCarb-DR as a separate component. We then utilized this component to develop a new private API, independent of UniCarb-DR and accessible only from GlycoPOST, which we named MiniCarb-Viewer. MiniCarb-Viewer registers GlycoWorkbench files submitted to GlycoPOST in its own private database like UniCarb-DR, and performs the necessary data analysis for visualization. This provides a data visualization function



	A	B
1		**MIRAGE LC Guidelines**
2		**Guidelines for reporting liquid chromatography experiments**
3		
4		Provide the information below. When more than one element has to be reported (e.g. Manufacturers or models, etc.), please continue in the adjacent columns.
5		Required fields are marked with an * (If the field is not relevant for the experiment, it can be filled in with a 'NA' -Not applicable).
6		
7	**1. General features, global MIRAGE descriptors**	
8	*	Date stamp
9	*	Responsible person. Provide name,
10	*	Affiliation
11	*	Stable contact information
12	**2. Equipment**	
13		*A wide variety of LC instrumentation can be used to analyze complex samples in various glycomics approaches. As the utility of the data generated from a LC analysis will be related to the type of chromatography and its configuration minimal information should include:*
14	*	Manufacturer
15	*	Model
16	*	Instrument details
17		**2.1. Column details & characteristics**
18	*	Manufacturer
19	*	Model
20	*	Separation Mode
21	*	Column length
22	*	Inner diameter
23		Total volume (if relevant)
24	*	Stationary Phase
25	*	Column heater
26	*	Additional accessories
27		**2.2 Mobile phase**
28	*	Description of solvent
29		Description of constituents
30		**2.3 Properties of the chromatographic run (parameters that may change over run time)**
31	*	Time
32	*	Gradient
33	*	Flow rate
34	*	Temperature
35		**Sample injection procedure**
36	*	Volume
37	*	Buffer
38	*	Inline/offline
39	*	Direct/loop (full/partial loop)
40	*	Flush conditions
41	*	Sample storage temperature
42	*	Sample loop material
43	*	Type of sample vials used
44		**2.4 Pre- and post run processes**
45		***2.4 (a) Pre-run process***
46	*	Type
47	*	Substance, standards
48	*	Time
49	*	Flow rate
50	*	Type of gradient mixer
51	*	Post separation events
52	*	Column regeneration
53	*	Equipment used for detection

FIGURE 3.17: The template Excel spreadsheet for entering MIRAGE-compliant liquid chromatography experiment guidelines.



Sample project (Rev. 0)	
Keywords	Sample
Description	Sample project
Revision	0

L0000000018

Title	Imported from MIRAGE Excel - 2023-08-18 13:32:24
Equipment	
Instrument deta...	The LC-2030C NT is an integrated high-performance liquid chromatography system which prioritizes easy-to-use features for a smooth analysis experience.
Column details ...	
Manufacturer	Agilent

Sample preparation — Clear
S0000000022

Title	Imported from MIRAGE Excel - 2023-10-04 19:40:35
General i...	The stimulated whole salivary secretion rate was determined

General features — Clear
G0000000019

Title	Imported from MIRAGE Excel - 2023-10-04 19:40:35
Global de...	
Instrume...	
Instrume...	

Ion sources — Clear
I0000000020

Title	Imported from MIRAGE Excel - 2023-10-04 19:40:35
Electrosp...	
Supply ty...	fed
Interface ...	

Ion transfer optics — Clear
T0000000021

Title	Imported from MIRAGE Excel - 2023-10-04 19:40:35
Hardware...	
Post-sou...	
Collision-...	

Spectrum and peak list generation and annotation
(Not set)

Liquid chromatography — Clear
L0000000018

Title	Imported from MIRAGE Excel - 2023-08-18 13:32:24
Equipment	
Instrume...	The LC-2030C NT is an integrated

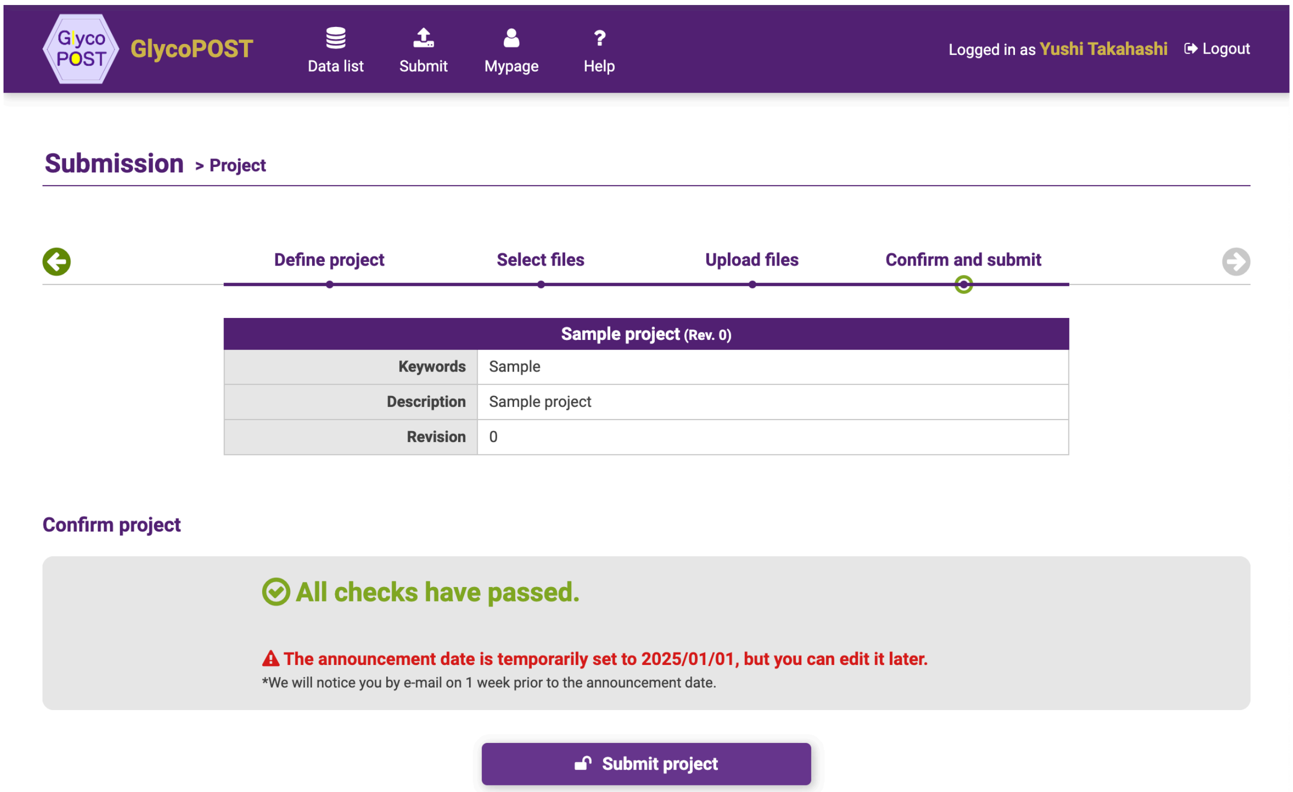
+ Drop files here, or click to select files to upload

Select	File name	Experimental preset	File size	Location
	RBC_OG.gwp results		3.2 MB	local
	RBC_NG.gwp results		73.7 MB	local
	0000_180913_08RBC_NG.raw raw	S Imported from MIRAGE Excel - 2023-10-04 19:40:35 G Imported from MIRAGE Excel - 2023-10-04 19:40:35 S Imported from MIRAGE Excel - 2023-10-04 19:40:35 T Imported from MIRAGE Excel - 2023-10-04 19:40:35 L Imported from MIRAGE Excel - 2023-08-18 13:32:24	773.8 MB	local
	0000_180913_07RBC_OG.raw raw	S Imported from MIRAGE Excel - 2023-10-04 19:40:35 G Imported from MIRAGE Excel - 2023-10-04 19:40:35 S Imported from MIRAGE Excel - 2023-10-04 19:40:35 T Imported from MIRAGE Excel - 2023-10-04 19:40:35 L Imported from MIRAGE Excel - 2023-08-18 13:32:24	648.3 MB	local

1 - 4 / 4 1

FIGURE 3.18: Updated GlycoPOST data submission form as of October 4, 2023. Users can now submit data with liquid chromatography experimental information.

similar to UniCarb-DR only for unpublished data in GlycoPOST. The request to register the GlycoWorkbench files to MiniCarb-Viewer is issued when a researcher clicks the “Submit project” button on the confirmation screen in GlycoPOST for a submission containing GlycoWorkbench files, as shown in Figure 3.19. In GlycoPOST, private submissions may be revised, which may increase or decrease the number of GlycoWorkbench files in the submission. To handle this properly, MiniCarb-Viewer registers the GlycoWorkbench files separately to the database for each revision of each submission. Researchers need to click the “Submit project” button on this same confirmation screen, even when revising a submission, so that MiniCarb-Viewer can reflect the contents of every revision of each submission.



Sample project (Rev. 0)	
Keywords	Sample
Description	Sample project
Revision	0

Confirm project

All checks have passed.

The announcement date is temporarily set to 2025/01/01, but you can edit it later.

*We will notice you by e-mail on 1 week prior to the announcement date.

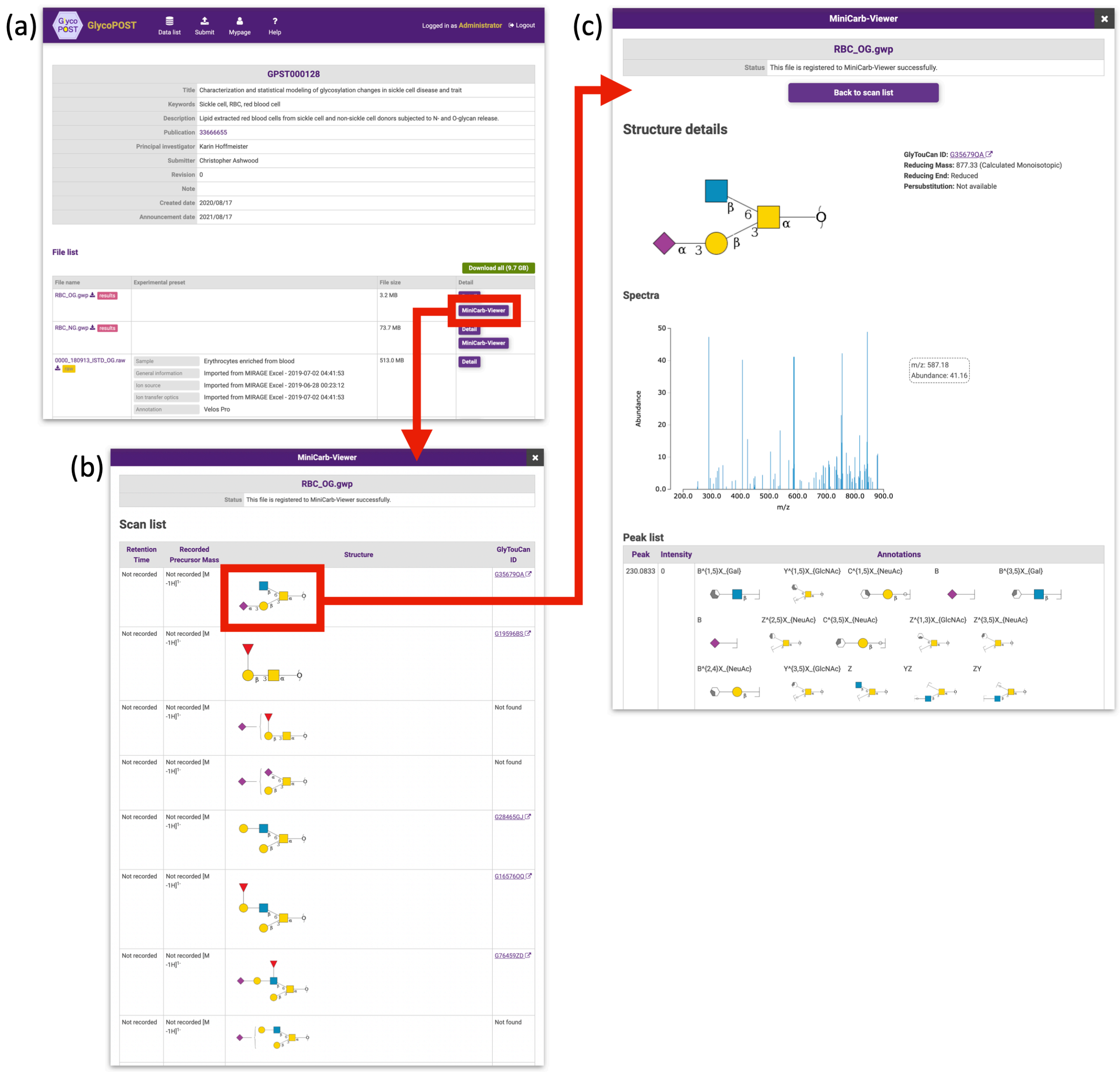
FIGURE 3.19: Screenshot of GlycoPOST submission confirmation screen as of January 1, 2024. By clicking the “Submit project” button at the bottom of this screen, the submission content is fixed. At the moment GlycoPOST receives this request, it will send an asynchronous request to MiniCarb-Viewer to register the GlycoWorkbench files contained in the submission.

Figure 3.20 shows the workflow of using MiniCarb-Viewer from the preview screen, which makes a private submission in GlycoPOST available to reviewers only. First, when the reviewer opens the preview screen, a list of the various files included in the submission will be displayed (Figure 3.20 (a)). If it contains GlycoWorkbench format files (GWP files), the “MiniCarb-Viewer” button will appear on the right hand side of the screen. By clicking this button, the identification results of the MS/MS experiment included in the GWP file are displayed as a list, similar to UniCarb-DR (Figure 3.20 (b)). By clicking the image of the glycan structure showing each identification result in the list, the chart of the peak list included in the identification result and the annotation information for each peak are displayed in detail (Figure 3.20 (c)).

3.2.4 Integration with GlyTouCan by obtaining GlyTouCan IDs from glycan structure strings in GWS format

In the GlycoWorkbench files that UniCarb-DR can recognize, the GlycoWorkbench sequence (GWS) format is adopted as a glycan structure notation method. Since GlyTouCan does not support GWS strings, there has been little progress in linking the glycan structures in the identification results registered in UniCarb-DR with the glycan structures in GlyTouCan. However, now that GlyCosmos has implemented the `gws2wurcs` API, which accepts GWS strings and returns GlyTouCan IDs, it is possible to use this API to find the GlyTouCan ID corresponding to each registered glycan structure in UniCarb-DR. In addition to UniCarb-DR, MiniCarb-Viewer also represents glycan structures in GWS format, therefore we also worked on linking UniCarb-DR and MiniCarb-Viewer with GlyTouCan by utilizing this API.

To accomplish this, we added JavaScript code to UniCarb-DR and MiniCarb-Viewer that runs in the browser. This program will be executed after the web page is loaded by the web browser and passes a GWS format glycan structure string to the GlyCosmos `gws2wurcs` API. Then, after the web browser retrieves the result from the API, if the result from the API contains a valid GlyTouCan ID, a link to the corresponding GlyTouCan entry is automatically displayed on the screen. Figure 3.21 shows a conceptual diagram of the relationship among UniCarb-DR, MiniCarb-Viewer, and GlyTouCan.



GPST000128

Title	Characterization and statistical modeling of glycosylation changes in sickle cell disease and trait
Keywords	Sickle cell, RBC, red blood cell
Description	Lipid extracted red blood cells from sickle cell and non-sickle cell donors subjected to N- and O-glycan release.
Publication	33666655
Principal investigator	Karin Hoffmeister
Submitter	Christopher Ashwood
Revision	0
Note	
Created date	2020/08/17
Announcement date	2021/08/17

File list

File name	Experimental preset	File size	Detail
RBC_OG.gwp results		3.2 MB	Detail MiniCarb-Viewer
RBC_NG.gwp results		73.7 MB	Detail MiniCarb-Viewer
0000_180913_ISTD_OG.raw raw	Sample: Erythrocytes enriched from blood; General information: Imported from MIRAGE Excel - 2019-07-02 04:41:53; Ion source: Imported from MIRAGE Excel - 2019-06-28 00:23:12; Ion transfer optics: Imported from MIRAGE Excel - 2019-07-02 04:41:53; Annotation: Velos Pro	513.0 MB	Detail

RBC_OG.gwp

Status: This file is registered to MiniCarb-Viewer successfully.

Scan list

Retention Time	Recorded Precursor Mass	Structure	GlyTouCan ID
Not recorded	Not recorded [M -1H]$^{1-}$		G35679QA
Not recorded	Not recorded [M -1H]$^{1-}$		G19596BS
Not recorded	Not recorded [M -1H]$^{1-}$		Not found
Not recorded	Not recorded [M -1H]$^{1-}$		Not found
Not recorded	Not recorded [M -1H]$^{1-}$		G28465GJ
Not recorded	Not recorded [M -1H]$^{1-}$		G16576OQ
Not recorded	Not recorded [M -1H]$^{1-}$		G76459ZD
Not recorded	Not recorded [M -1H]$^{1-}$		Not found

RBC_OG.gwp

Status: This file is registered to MiniCarb-Viewer successfully.

Structure details

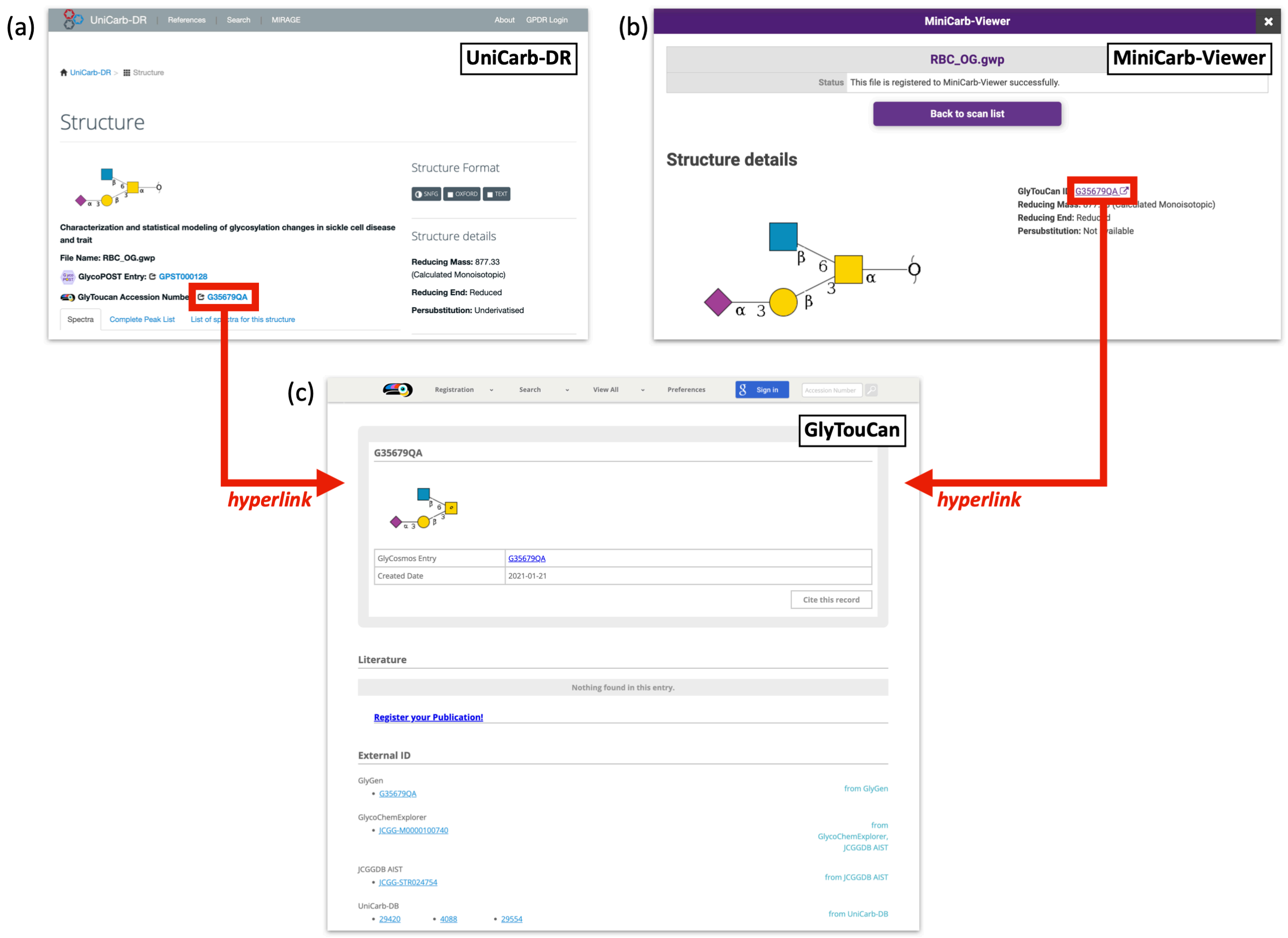
GlyTouCan ID: G35679QA
Reducing Mass: 877.33 (Calculated Monoisotopic)
Reducing End: Reduced
Persubstitution: Not available

Spectra

Peak list

Peak	Intensity	Annotations
230.0833	0	B^{1,5}X_{Gal} Y^{1,5}X_{GlcNAc} C^{1,5}X_{NeuAc} B B^{3,5}X_{Gal} B Z^{2,5}X_{NeuAc} C^{3,5}X_{NeuAc} Z^{1,3}X_{GlcNAc} Z^{3,5}X_{NeuAc} B^{2,4}X_{NeuAc} Y^{3,5}X_{GlcNAc} Z YZ ZY

FIGURE 3.20: Workflow of using MiniCarb-Viewer from the preview screen of GlycoPOST.



Structure

Characterization and statistical modeling of glycosylation changes in sickle cell disease and trait

File Name: RBC_OG.gwp

GlycoPOST Entry: GPST000128

GlyToucan Accession Numbe G35679QA

Structure Format

Structure details

Reducing Mass: 877.33 (Calculated Monoisotopic)

Reducing End: Reduced

Persubstitution: Underivatised

RBC_OG.gwp

Status This file is registered to MiniCarb-Viewer successfully.

Structure details

GlyTouCan I G35679QA

Reducing Mass: 877. (Calculated Monoisotopic)

Reducing End: Reduc d

Persubstitution: Not ailable

G35679QA

GlyCosmos Entry	G35679QA
Created Date	2021-01-21

Literature

Nothing found in this entry.

Register your Publication!

External ID

GlyGen
- G35679QA

from GlyGen

GlycoChemExplorer
- JCGG-M0000100740

from GlycoChemExplorer, JCGGDB AIST

JCGGDB AIST
- JCGG-STR024754

from JCGGDB AIST

UniCarb-DB
- 29420
- 4088
- 29554

from UniCarb-DB

FIGURE 3.21: Conceptual diagram showing the relationship between UniCarb-DR, MiniCarb-Viewer and GlyTouCan. GlyTouCan IDs can now be obtained from the GWS format glycan structures contained in the UniCarb-DR and MiniCarb-Viewer data. This allows linking from these resources to the corresponding entries in GlyTouCan.

Chapter 4

Discussion

4.1 Knowledge integration between glycomics and glycoproteomics catalyzed by GlyComb

With the development of GlyComb, the first data repository for glycoconjugate data including glycopeptide and glycoprotein data from glycoproteomics studies, we believe that glycobiology research can be accelerated by combining knowledge from both glycomics and glycoproteomics domains. For instance, in glycomics, a GlyTouCan ID has been assigned to each glycan structure as a unique identifier through the development of the glycan structure repository GlyTouCan. This has enabled glycomics researchers to precisely specify glycan structure information using the corresponding GlyTouCan IDs in their publications. Furthermore, each glycan structure entry in GlyTouCan can be associated with a publication related to that structure, making it possible to collect publications that contain references to the glycan structure with a particular GlyTouCan ID. In addition, glycoscience portal sites, such as GlyCosmos, integrate metadata such as the species in which the glycan structure was discovered, the name of the motif it contains, and various external resources describing the same structure, for each glycan structure registered in GlyTouCan. These results are available under the GlyCosmos Glycans (`https://glycosmos.org/glycans`) data resource. Recently, the collaboration between GlyCosmos and PubChem has also been enhanced (Cheng et al., 2023) largely due to these GlyTouCan identifiers.

Like GlyTouCan, GlyComb is a data repository that allows unique identifiers to be assigned to each glycoconjugate entry, which we expect will enable the integration of knowledge linking glycomics with proteomics and other omics data in the future. We have shown how this was possible through investigating the entries registered from MS-Viewer and PRIDE. As a result, we found actually overlapping entries from both repositories, which was made possible by the GlyComb IDs assigned to them. We investigated these GlyComb IDs further to see if they happened to be submitted by the same user to both repositories, and we confirmed that while none of the duplicate entries originated from the same submitters, the exact same glycopeptides were reported. Moreover, we can compare the UniProt IDs for different entries. For example, the same UniProt ID is provided for the GlyComb entry numbers GC0170AE and GC017098 shown in Table 3.3, where we can see that they have the same peptide sequence “QQLQEQSAPPSKPDGQLQFR” and the same glycan modification on the serine residue at position 11. However, GC0170AE has an additional glycan modification on the serine residue at position 7. A similar relationship holds for the GC0170B3 and GC0173AC pair, which were found in one project, PXD035445 in PRIDE, and in two projects, PPV000002, and PPV000046 in MS-Viewer. These entries are all mucin *O*-glycan modifications, which are difficult to analyze. By integrating such glycoproteomics data with accession numbers, the whole picture can be examined, providing insight into the function of mucin *O*-glycans. As this work advances, we believe that the integration of knowledge from multiple omics fields (glycomics, proteomics, and glycoproteomics) will contribute to lowering the hurdle for multi-omics research in glycoscience.

What we consider to be the key to greatly improving the potential usefulness of GlyComb is to maximize the use of Semantic Web technologies. In GlyComb, each published glycoconjugate entry assigned a GlyComb ID is sequentially converted into a format called Resource Description Framework (RDF) and stored in a database for Semantic Web technologies called a triplestore. This allows researchers to retrieve a variety of information scattered across multiple different databases at once, using a query language called SPARQL. For example, major biological pathway databases such as Reactome (Gillespie et al., 2022) and WikiPathways (Martens et al., 2021) have already published their pathway data in RDF. The pathway information in these databases could be combined with the glycoconjugate information stored in GlyComb using the SPARQL query language to find the corresponding GlyComb ID for each glycoconjugate entry in a pathway, enabling researchers to find out from GlyComb which species and which chemical pathways a particular glycoconjugate entry of interest appears in. Furthermore, it is expected to make it easier to obtain a comprehensive list of glycoconjugate entries related to a specific disease.

Moreover, subsumption relationships are defined between multiple glycan structures in glycomics, and SPARQL can be used to retrieve more specific glycan structures from glycan structures with ambiguous linkage and monosaccharide information. Owing to this, the glycan composition information contained in each glycoconjugate entry in GlyComb will provide insight into what glycan structures are actually attached to the glycoconjugate molecule. In addition to glycan structures, it may be possible to define such relationships between glycopeptide entries in GlyComb based on the peptide sequence and attached glycan structures of each entry. In LC-MS/MS experiments, some of the peptide sequences resulting from protease treatment may be a complete subpart of other peptide sequences. In other cases, the peptide sequence may be exactly the same. However, only the attached glycan structures may be different. By defining the relationships among these corresponding entries, we may be able to discover new knowledge that has not been revealed before when integrating with various information such as pathway data using SPARQL. For example, in the overlapping UniProt IDs found in the common GlyComb entries in Table 3.3, we were able to find peptide sequences where one subsumed the other. Using Semantic Web technologies, such relationships can be defined, to integrate such knowledge into a whole. Consequently, it is expected that the use of SPARQL will enable us to gain insight into glycoconjugates by integrating multiple omics data, and conversely, it will enable access to glycoconjugate information from various omics fields, which will lead to a greater understanding of biological processes in general.

At the time of writing, GlyComb does not support the input of quantitative abundance data for glycopeptides and glycoproteins. However, by expanding GlyComb to accommodate such input data in the future, and by collecting the identification results of glycoproteomics experiments further, it is expected to be made possible to compare the distribution of each glycoprotein present in different organs of each species. This will allow us to see differences in the tendency of protein glycosylation in different organs that were not visible before and may lead to a better understanding of the effects of glycosylation on protein function. This is an example of an analysis that could not have been performed previously in the bioinformatics field without the powerful infrastructure developed in this study, GlyComb.

4.2 From integration of glycomics data repositories to integration of multi-omics knowledge

We have also strengthened the integration of the two complementary data repositories, GlycoPOST and UniCarb-DR, from various perspectives. In particular, the data submission flow to these repositories has been unified, enabling users to upload identification results along with raw data to GlycoPOST, which can then be automatically visualized in UniCarb-DR. We believe that this has greatly improved the convenience for users. Furthermore, by enabling cross referencing of corresponding entries between GlycoPOST and UniCarb-DR, these data repositories no longer operate independently, but rather function as one comprehensive data repository system in which the two data repositories work in unison. In addition, it is now possible to link to GlyTouCan from UniCarb-DR and MiniCarb-Viewer. These links are still unidirectional, and there is no link from GlyTouCan to UniCarb-DR or GlycoPOST yet. However, it is expected that these data repositories, including GlyTouCan, will be able to work consistently in the future, which will contribute to more user convenience, and that these data repositories will be able to work in unison by collaborating with each other and function as an extremely powerful information infrastructure in glycoscience research.

GlycoPOST and UniCarb-DR are extremely important data repositories to which glycomics researchers around the world submit the results of their glycomics experiments. Now that this study has enabled these data repositories to cooperate with each other while fulfilling their respective roles of storing large amounts of data and visualizing and retrieving identification results, the next step is to convert the data in these data repositories into RDF data to support the Semantic Web. This may allow the integration of glycomics experimental data accumulated in GlycoPOST and UniCarb-DR with knowledge from other omics areas by searching with the SPARQL language. For example, it would be possible to use experimental data submitted according to the MIRAGE sample preparation guidelines and the identification results in GlycoWorkbench format to search for proteins known to bind the identified glycan structures in the exact same type of cells or organs as the sample used in the experiment.

4.3 The value of reliable data repositories

In accumulating large amounts of data and extracting valuable information from it, a flawed program in a data repository can slow down the progress of the entire life science field. However, scripting languages such as Python, which bioinformaticians frequently use for its ease of use, are not effective in making developers aware that the programs they write contain problems. This is a very big obstacle in the small bioinformatics community, which cannot employ enough people and spend enough time exhaustively testing the software they create before releasing it, as companies can. In this study, to mitigate this problem a little, I utilized functional programming, a tool that the bioinformatics community has not made much use of to date, and as a result, I succeeded in developing reliable data repositories. This would have contributed to further data accumulation in the field of glycoscience.

In the future, the developed data repositories will be expanded so that they can be linked to each other. It is considered important to not only accumulate data from multiple omics fields, but also to link them to each other and create networks among them as a basis for promoting multi-omics analysis. Therefore, since it is extremely important to extract and associate data correctly, utmost care must be taken when extending an existing data repository so as not to accidentally break existing functionality. In this study, I have written programs that allow type checkers to detect potential problems in them before they are executed, rather than actually executing the programs whenever possible, and this makes future extensions of these data repositories dramatically safer than they have been in the past. Consequently, the data repositories developed in this study are expected to be useful for data integration for future multi-omics analyses.

4.4 Future outlook

First of all, we plan to enable GlyComb to store glycolipid and glycoside information, which are glycoconjugate information that GlyComb does not yet support. This will allow GlyComb to integrate lipidomics and metabolomics information in the future, providing insight into how the presence or absence of glycosylation *in vivo* affects the function of each molecule. Furthermore, I am planning to strengthen the collaboration between GlyComb and existing data repositories such as GlyTouCan, GlycoPOST, and UniCarb-DR in the glycosciences, and PRIDE, PubChem, LIPID MAPS (Sud et al., 2007), etc. in other omics fields. In order to extract the maximum amount of information from the results obtained from various studies, we plan to implement a partnership system into GlyComb. This system would allow, for example, data submitted to GlycoPOST containing the identification results of glycoconjugate data in an open file format, to be processed such that the relevant glycoconjugate results could be extracted and automatically registered in GlyComb. This would allow users to check the list of GlycoPOST entries in which a particular glycoconjugate molecule appears in the identification results via GlyComb IDs, or to check the list of GlyComb entries containing a particular glycan structure via GlyTouCan IDs. It is expected that GlyComb has the potential to unravel the hidden functions of glycoconjugate molecules by serving as a hub for integrating data from various omics fields, including glycomics, proteomics, and glycoproteomics.

In addition, the members of the international MIRAGE committee are currently discussing reporting guidelines not only for glycomics but also for glycoproteomics experiments. Accordingly, I plan to make GlycoPOST capable of supporting the input of MIRAGE guideline information for glycoproteomics experiments in the future. This will allow GlycoPOST to accumulate not only glycomics but also glycoproteomics experimental data. With this advancement, GlycoPOST and GlyComb are expected to collaborate with each other even more strongly in the future.

With these further developments, the data repositories developed in this thesis are expected to serve as a foundation for integrating knowledge from a very wide range of omics fields: glycomics, proteomics, glycoproteomics, lipidomics, and metabolomics, as shown in Figure 4.1. These data repositories that can accumulate information from multiple omics fields will be indispensable for future multi-omics analysis.

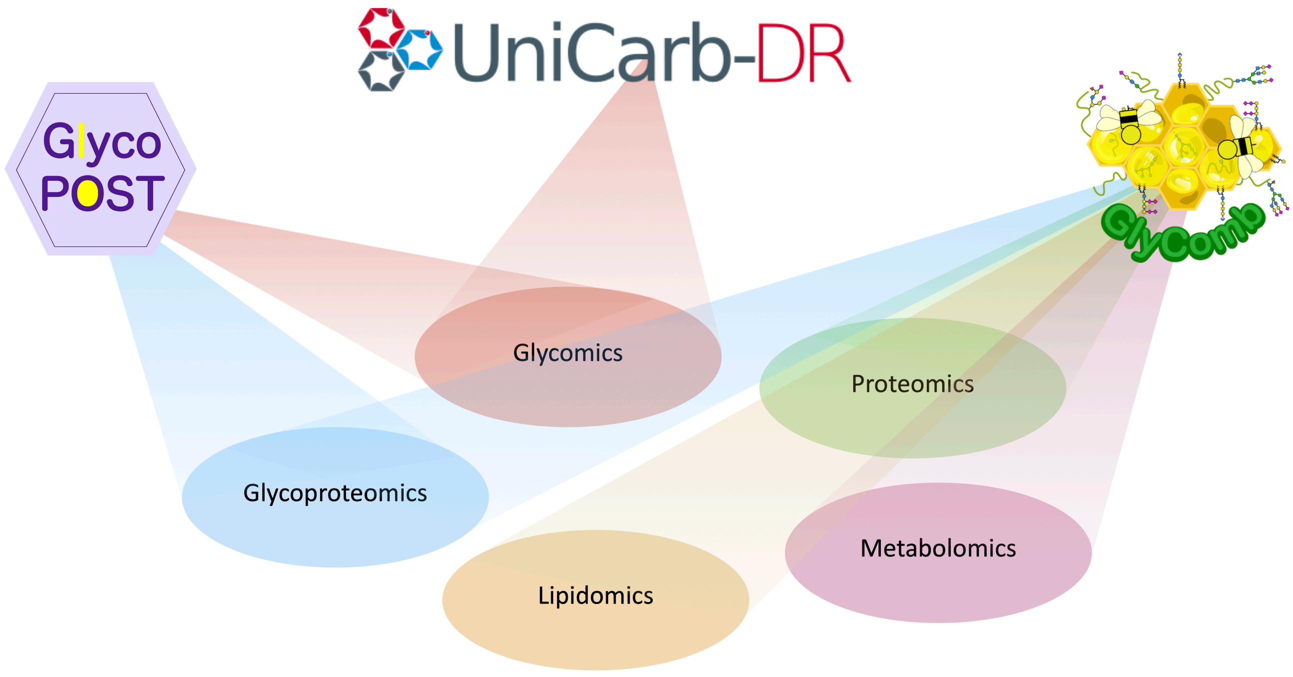


FIGURE 4.1: Expected omics fields to be covered by the data repositories developed in this thesis in the future. GlyComb currently contributes to the integration of data from the glycoproteomics and proteomics fields as it can store glycopeptide and glycoprotein entries that may contain UniProt ID data. It will further help to integrate knowledge in the lipidomics and metabolomics fields by supporting the registration of glycolipid and glycoside entries in the future. Furthermore, by supporting the registration of MIRAGE guideline information for the reporting of glycoproteomics experiments, GlycoPOST will also make a significant contribution to the accumulation of data in the field of glycoproteomics as well as glycomics.

Chapter 5

Conclusion

In glycoscience research, data obtained from various experiments, including mass spectrometry experiments, are continuously accumulated every day. Glycoinformatics is an extremely important research field due to the potential to extract the maximum amount of knowledge from such data. Furthermore, data repositories, the primary focus of this thesis, are the fundamental infrastructure for future research in the field of glycoinformatics. By accumulating submissions from researchers around the world, including even raw data from mass spectrometers, data repositories make it possible to reanalyze large-scale experimental datasets using computers. GlycoPOST is an ideal data repository for submitting such data, and now that the data submission flow is unified with UniCarb-DR, it is possible to visualize the results of mass spectrometry experiments and search for entries using peak list data obtained from those experiments. These two data repositories now form a comprehensive system that can mutually take advantage of each other's strengths.

In addition, these two data repositories are now able to represent glycan structures using GlyTouCan IDs, which has partially enhanced their collaboration with GlyTouCan, the international glycan structure repository. This is another important aspect of data repositories: they can not only accumulate submitted data but also form mutual links among them. This makes it possible, for example, to obtain a GlyTouCan ID for a glycan structure contained in the results of an experiment submitted to GlycoPOST, and then search GlyTouCan to find out in what literature the glycan structure has been reported. In this way, forming links between entries submitted to the data repositories may lead to new knowledge discovery when performing large-scale reanalysis of experimental data.

However, one of the missing pieces in the foundation of glycoscience research previously was the existence of a data repository capable of storing glycoconjugate data. The development of GlyComb has made it possible to more efficiently collect and share information on glycopeptides, glycoproteins, and other glycoconjugates, which have not been accumulated before. This has even made it possible to compare glycopeptide and glycoprotein information obtained from glycoproteomics experiments that have been submitted to multiple existing data repositories, such as PRIDE. GlyComb also uses GlyTouCan IDs to denote glycan structures bound to glycoconjugate entries. Thus, in the future, GlyTouCan IDs could be used to interlink GlycoPOST , UniCarb-DR, GlyTouCan, and GlyComb. Moreover, new linkages of information will be formed in the future using GlyComb IDs, which are the accession numbers of GlyComb. For example, it would be possible to represent each of the numerous glycoconjugate entries in the biological reaction pathway by a GlyComb ID. By combining various omics information with the glycoconjugate data stored in GlyComb using Semantic Web technologies, it is expected that the influences of glycoproteins and glycolipids on various diseases, infection, and developmental mechanisms with and without glycosylation will be further elucidated in the future.

However, extending existing data resources without unintentionally breaking existing functionality can easily become more challenging than expected without thoughtful consideration. This would be a major obstacle to promoting collaboration among multiple data repositories. This is exactly the same problem that is occurring in the worldwide IT industry. In this study, I applied some of the functional programming techniques that have been developed in the field of computer science as a means of reducing this difficulty and developing data repositories stably and continuously, which has been overlooked in bioinformatics field. This allowed us to increase as much as possible the number of problems hidden in the program that the type checker could find before executing the program, and to introduce a mechanism by which the developers of the data repositories could notice the problems early and continue to enhance the functionality of the program efficiently. The data repository developed in this study is considered to be highly reliable in the bioinformatics field, since the use of results from the computer science research field has not been active in the bioinformatics field, which generally mainly uses scripting languages such as Python. Therefore, these data repositories are considered to be extremely useful and unique resources that can withstand use by researchers worldwide.

As a result, the development of these data repositories, which are the infrastructure for glycoinformatics research, will contribute to the future advancement of glycoscience research. All of these data repositories are linked directly from GlyCosmos, one of the knowledge bases in glycoscience research. Although the data stored in the data repositories are not curated, such stored data will then be curated and annotated, and combined with knowledge in knowledge bases such as GlyCosmos. This would allow the data repositories to contribute to new knowledge discovery.

Acknowledgements

First of all, I would like to acknowledge my supervisor, Prof. Kiyoko F. Aoki-Kinoshita, for her numerous advices and guidance in carrying out this research project. I am also very grateful to my numerous collaborators, including Masaaki Shiota (Soka University), Dr. Akihiro Fujita (Soka University), Dr. Issaku Yamada (The Noguchi Institute), Prof. Shujiro Okuda (Niigata University), and Prof. Niclas Karlsson (Oslo Metropolitan University University, Norway). In addition, I would like to acknowledge the insightful discussions and advice on GlyComb by GlyCosmos project members, including Daisuke Shinmachi (National Institute of Advanced Industrial Science and Technology (AIST)), Chiaki Nagai-Okatani (AIST), Atsushi Kuno (AIST), Shujiro Okuda, and Masaaki Matsubara (The Noguchi Institute). Finally, I would like to thank all the members of the Kinoshita Lab and my family for their support.

Bibliography

Aoki-Kinoshita, K. F., Lisacek, F., Mazumder, R., York, W. S., and Packer, N. H. "The GlySpace Alliance: toward a collaborative global glycoinformatics community". In: *Glycobiology* 30.2 (2020), pp. 70–71.

Bagdonaite, I., Malaker, S. A., Polasky, D. A., Riley, N. M., Schjoldager, K., Vakhrushev, S. Y., Halim, A., Aoki-Kinoshita, K. F., Nesvizhskii, A. I., Bertozzi, C. R., Wandall, H. H., Parker, B. L., Thaysen-Andersen, M., and Scott, N. E. "Glycoproteomics". In: *Nature Reviews Methods Primers* 2.1 (2022), p. 48.

Baker, P. R. and Chalkley, R. J. "MS-viewer: a web-based spectral viewer for proteomics results". In: *Molecular & Cellular Proteomics* 13.5 (2014), pp. 1392–1396.

Banin, E., Neuberger, Y., Altshuler, Y., Halevi, A., Inbar, O., Nir, D., and Dukler, A. "A novel linear code® nomenclature for complex carbohydrates". In: *Trends in Glycoscience and Glycotechnology* 14.77 (2002), pp. 127–137.

Bao, X., Kobayashi, M., Hatakeyama, S., Angata, K., Gullberg, D., Nakayama, J., Fukuda, M. N., and Fukuda, M. "Tumor suppressor function of laminin-binding α-dystroglycan requires a distinct β3-*N*-acetylglucosaminyltransferase". In: *Proceedings of the National Academy of Sciences* 106.29 (2009), pp. 12109–12114.

Beckett, D., Berners-Lee, T., Prud'hommeaux, E., and Carothers, G. "RDF 1.1 Turtle". In: *World Wide Web Consortium* (2014), pp. 18–31.

Bern, M., Kil, Y. J., and Becker, C. "Byonic: advanced peptide and protein identification software". In: *Current protocols in bioinformatics* 40.1 (2012), pp. 13–20.

Berners-Lee, T., Hendler, J., and Lassila, O. "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities". In: *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web*. 2023, pp. 91–103.

Bloch, J. *Effective Java*. Addison-Wesley Professional, 2008.

Bradshaw, S., Brazil, E., and Chodorow, K. *MongoDB: the definitive guide: powerful and scalable data storage*. O'Reilly Media, 2019.

Bray, T. *The JavaScript Object Notation (JSON) Data Interchange Format.* Tech. rep. 2014.

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., and Vingron, M. "Minimum information about a microarray experiment (MIAME)—toward standards for microarray data". In: *Nature Genetics* 29.4 (2001), pp. 365–371.

Campbell, M. P., Abrahams, J. L., Rapp, E., Struwe, W. B., Costello, C. E., Novotny, M., Ranzinger, R., York, W. S., Kolarich, D., Rudd, P. M., and Kettner, C. "The minimum information required for a glycomics experiment (MIRAGE) project: LC guidelines". In: *Glycobiology* 29.5 (2019), pp. 349–354.

Carlson, J. *Redis in action.* Simon and Schuster, 2013.

Ceroni, A., Maass, K., Geyer, H., Geyer, R., Dell, A., and Haslam, S. M. "GlycoWorkbench: a tool for the computer-assisted annotation of mass spectra of glycans". In: *Journal of proteome research* 7.4 (2008), pp. 1650–1659.

Chen, Z., Yu, Q., Yu, Q., Johnson, J., Shipman, R., Zhong, X., Huang, J., Asthana, S., Carlsson, C., Okonkwo, O., and Li, L. "In-depth site-specific analysis of N-glycoproteome in human cerebrospinal fluid and glycosylation landscape changes in Alzheimer's disease". In: *Molecular & Cellular Proteomics* 20 (2021).

Cheng, T., Ono, T., Shiota, M., Yamada, I., Aoki-Kinoshita, K. F., and Bolton, E. E. "Bridging glycoinformatics and cheminformatics: integration efforts between GlyCosmos and PubChem". In: *Glycobiology* 33.6 (2023), pp. 454–463.

Cherny, B. *Programming TypeScript: making your JavaScript applications scale.* O'Reilly Media, 2019.

Choi, M., Carver, J., Chiva, C., Tzouros, M., Huang, T., Tsai, T.-H., Pullman, B., Bernhardt, O. M., Hüttenhain, R., Teo, G. C., Perez-Riverol, Y., Muntel, J., Müller, M., Goetze, S., Pavlou, M., Verschueren, E., Wollscheid, B., Nesvizhskii, A. I., Reiter, L., Dunkley, T., Sabidó, E., Bandeira, N., and Vitek, O. "MassIVE. quant: a community resource of quantitative mass spectrometry–based proteomics datasets". In: *Nature methods* 17.10 (2020), pp. 981–984.

Church, A. "An unsolvable problem of elementary number theory". In: *American Journal of Mathematics* 58 (1936), pp. 354—363.

— *The Calculi of Lambda-Conversion.* Princeton University Press, 1941.

Codd, E. F. “A relational model of data for large shared data banks”. In: *Communications of the ACM* 13.6 (1970), pp. 377–387.

Coelho, L. P., Alves, R., Monteiro, P., Huerta-Cepas, J., Freitas, A. T., and Bork, P. “NG-meta-profiler: fast processing of metagenomes using NGLess, a domain-specific language”. In: *Microbiome* 7.1 (2019), pp. 1–10.

Damerell, D., Ceroni, A., Maass, K., Ranzinger, R., Dell, A., and Haslam, S. M. “The GlycanBuilder and GlycoWorkbench glycoinformatics tools: updates and new developments”. In: *Biological chemistry* 393.11 (2012), pp. 1357–1362.

Darula, Z., Chalkley, R. J., Lynn, A., Baker, P. R., and Medzihradszky, K. F. “Improved identification of *O*-linked glycopeptides from ETD data with optimized scoring for different charge states and cleavage specificities”. In: *Amino Acids* 41 (2011), pp. 321–328.

Day, C. J., Tran, E. N., Semchenko, E. A., Tram, G., Hartley-Tassell, L. E., Ng, P. S., King, R. M., Ulanovsky, R., McAtamney, S., Apicella, M. A., Tiralongo, J., Morona, R., Korolik, V., and Jennings, M. P. “Glycan: glycan interactions: High affinity biomolecular interactions that can mediate binding of pathogenic bacteria to host cells”. In: *Proceedings of the National Academy of Sciences* 112.52 (2015), E7266–E7275.

Deutsch, E. W., Bandeira, N., Sharma, V., Perez-Riverol, Y., Carver, J. J., Kundu, D. J., García-Seisdedos, D., Jarnuczak, A. F., Hewapathirana, S., Pullman, B. S., Wertz, J., Sun, Z., Kawano, S., Okuda, S., Watanabe, Y., Hermjakob, H., MacLean, B., MacCoss, M. J., Zhu, Y., Ishihama, Y., and Vizcaíno, J. A. “The ProteomeXchange consortium in 2020: enabling ‘big data’ approaches in proteomics”. In: *Nucleic Acids Research* 48.D1 (2020), pp. D1145–D1152.

Dijkstra, E. W. “Letters to the editor: go to statement considered harmful”. In: *Communications of the ACM* 11.3 (1968), pp. 147–148.

Doeraene, S. *Scala.js: Type-directed interoperability with dynamically typed languages.* Tech. rep. 2013.

Douglas, K. and Douglas, S. *PostgreSQL: a comprehensive guide to building, programming, and administering PostgresSQL databases.* SAMS publishing, 2003.

DuBois, P. *MySQL.* Addison-Wesley, 2013.

Fitzpatrick, B. “Distributed caching with memcached”. In: *Linux journal* 2004.124 (2004), p. 5.

Fujita, A., Aoki, N. P., Shinmachi, D., Matsubara, M., Tsuchiya, S., Shiota, M., Ono, T., Yamada, I., and Aoki-Kinoshita, K. F. “The international glycan repository GlyTouCan version 3.0”. In: *Nucleic Acids Research* 49.D1 (2021), pp. D1529–D1533.

Gillespie, M., Jassal, B., Stephan, R., Milacic, M., Rothfels, K., Senff-Ribeiro, A., Griss, J., Sevilla, C., Matthews, L., Gong, C., Deng, C., Varusai, T., Ragueneau, E., Haider, Y., May, B., Shamovsky, V., Weiser, J., Brunson, T., Sanati, N., Beckman, L., Shao, X., Fabregat, A., Sidiropoulos, K., Murillo, J., Viteri, G., Cook, J., Shorser, S., Bader, G., Demir, E., Sander, C., Haw, R., Wu, G., Stein, L., Hermjakob, H., and D'Eustachio, P. "The reactome pathway knowledgebase 2022". In: *Nucleic Acids Research* 50.D1 (2022), pp. D687–D692.

Haak, L. L., Fenner, M., Paglione, L., Pentz, E., and Ratner, H. "ORCID: a system to uniquely identify researchers". In: *Learned publishing* 25.4 (2012), pp. 259–264.

Herget, S, Ranzinger, R, Maass, K, and Lieth, C.-W. "GlycoCT—a unifying sequence format for carbohydrates". In: *Carbohydrate research* 343.12 (2008), pp. 2162–2171.

Hogan, A. "SPARQL query language". In: *The Web of Data* (2020), pp. 323–448.

Hudak, P. "Conception, evolution, and application of functional programming languages". In: *ACM Comput. Surv.* 21.3 (1989), pp. 359–411. ISSN: 0360-0300. DOI: 10.1145/72551.72554. URL: https://doi.org/10.1145/72551.72554.

Hudak, P., Hughes, J., Peyton Jones, S., and Wadler, P. "A history of Haskell: being lazy with class". In: *Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages*. HOPL III. San Diego, California: Association for Computing Machinery, 2007, pp. 12–1–12–55. ISBN: 9781595937667. DOI: 10.1145/1238844.1238856. URL: https://doi.org/10.1145/1238844.1238856.

Jarnuczak, A. F. and Vizcaíno, J. A. "Using the PRIDE database and ProteomeXchange for submitting and accessing public proteomics datasets". In: *Current protocols in bioinformatics* 59.1 (2017), pp. 13–31.

Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M. "The KEGG resource for deciphering the genome". In: *Nucleic Acids Research* 32.suppl_1 (2004), pp. D277–D280.

Katayama, T. and Kawashima, S. "SPARQList: Markdown-Based Highly Configurable REST API Hosting Server for SPARQL." In: *SWAT4LS*. 2017.

Kenler, E. and Razzoli, F. *MariaDB Essentials*. Packt Publishing Ltd, 2015.

Kim, S., Thiessen, P. A., Bolton, E. E., Chen, J., Fu, G., Gindulyte, A., Han, L., He, J., He, S., Shoemaker, B. A., Wang, J., Yu, B., Zhang, J., and Bryant, S. H. "PubChem Substance and Compound databases". In: *Nucleic Acids Research* 44.D1 (2016), pp. D1202–D1213.

Kleene, S. C. "λ-definability and recursiveness". In: *Duke Mathematical Journal* 2.2 (1936), pp. 340–353.

Kolarich, D., Rapp, E., Struwe, W. B., Haslam, S. M., Zaia, J., McBride, R., Agravat, S., Campbell, M. P., Kato, M., Ranzinger, R., Kettner, C., and York, W. S. “The minimum information required for a glycomics experiment (MIRAGE) project: improving the standards for reporting mass-spectrometry-based glycoanalytic data”. In: *Molecular & cellular proteomics* 12.4 (2013), pp. 991–995.

Lageveen-Kammeijer, G. S., Rapp, E., Chang, D., Rudd, P. M., Kettner, C., and Zaia, J. “The minimum information required for a glycomics experiment (MIRAGE): reporting guidelines for capillary electrophoresis”. In: *Glycobiology* 32.7 (2022), pp. 580–587.

Lee, L. Y., Moh, E. S., Parker, B. L., Bern, M., Packer, N. H., and Thaysen-Andersen, M. “Toward automated *N*-glycopeptide identification in glycoproteomics”. In: *Journal of proteome research* 15.10 (2016), pp. 3904–3915.

Liu, Y., McBride, R., Stoll, M., Palma, A. S., Silva, L., Agravat, S., Aoki-Kinoshita, K. F., Campbell, M. P., Costello, C. E., Dell, A., Haslam, S. M., Karlsson, N. G., Khoo, K.-H., Kolarich, D., Novotny, M. V., Packer, N. H., Ranzinger, R., Rapp, E., Rudd, P. M., Struwe, W. B., Tiemeyer, M., Wells, L., York, W. S., Zaia, J., Kettner, C., Paulson, J. C., Feizi, T., and Smith, D. F. “The minimum information required for a glycomics experiment (MIRAGE) project: improving the standards for reporting glycan microarray-based data”. In: *Glycobiology* 27.4 (Dec. 2016), pp. 280–284. ISSN: 0959-6658. DOI: 10.1093/glycob/cww118. eprint: https://academic.oup.com/glycob/article-pdf/27/4/280/10863110/cww118.pdf. URL: https://doi.org/10.1093/glycob/cww118.

Loeliger, J. and McCullough, M. *Version Control with Git: Powerful tools and techniques for collaborative software development.* ” O’Reilly Media, Inc.”, 2012.

Ma, J., Chen, T., Wu, S., Yang, C., Bai, M., Shu, K., Li, K., Zhang, G., Jin, Z., He, F., Hermjakob, H., and Zhu, Y. “iProX: an integrated proteome resource”. In: *Nucleic Acids Research* 47.D1 (2019), pp. D1211–D1217.

MacQueen, D., Harper, R., and Reppy, J. “The history of Standard ML”. In: *Proc. ACM Program. Lang.* 4.HOPL (2020). DOI: 10.1145/3386336. URL: https://doi.org/10.1145/3386336.

Manola, F., Miller, E., and McBride, B. “RDF primer”. In: *W3C recommendation* 10.1-107 (2004), p. 6.

Mariethoz, J., Alocci, D., Gastaldello, A., Horlacher, O., Gasteiger, E., Rojas-Macias, M., Karlsson, N. G., Packer, N. H., and Lisacek, F. “Glycomics@ ExPASy: bridging the gap”. In: *Molecular & Cellular Proteomics* 17.11 (2018), pp. 2164–2176.

Marlow, S. et al. “Haskell 2010 language report”. In: (2010).

Martens, M., Ammar, A., Riutta, A., Waagmeester, A., Slenter, D., Hanspers, K., A. Miller, R., Digles, D., Lopes, E., Ehrhart, F., Dupuis, L. J., Winckers, L. A., Coort, S., Willighagen, E. L., Evelo, C. T., Pico, A. R., and Kutmon, M. "WikiPathways: connecting communities". In: *Nucleic Acids Research* 49.D1 (2021), pp. D613–D621.

Matsubara, M., Aoki-Kinoshita, K. F., Aoki, N. P., Yamada, I., and Narimatsu, H. "WURCS 2.0 update to encapsulate ambiguous carbohydrate structures". In: *Journal of chemical information and modeling* 57.4 (2017), pp. 632–637.

Maxwell, E., Tan, Y., Tan, Y., Hu, H., Benson, G., Aizikov, K., Conley, S., Staples, G. O., Slysz, G. W., Smith, R. D., and Zaia, J. "GlycReSoft: A Software Package for Automated Recognition of Glycans from LC/MS Data". In: *PLOS ONE* 7.9 (Sept. 2012), pp. 1–11. DOI: 10.1371/journal.pone.0045474. URL: https://doi.org/10.1371/journal.pone.0045474.

McCarthy, J. "Recursive functions of symbolic expressions and their computation by machine, part I". In: *Communications of the ACM* 3.4 (1960), pp. 184–195.

McCracken, D. D. and Reilly, E. D. "Backus-Naur form (BNF)". In: *Encyclopedia of Computer Science.* 2003, pp. 129–131.

Merkel, D. "Docker: lightweight linux containers for consistent development and deployment". In: *Linux J* 239.2 (2014), p. 2.

Meyers, S. *Effective modern C++: 42 specific ways to improve your use of C++ 11 and C++ 14.* "O'Reilly Media, Inc.", 2014.

Michele, D. E., Barresi, R., Kanagawa, M., Saito, F., Cohn, R. D., Satz, J. S., Dollar, J., Nishino, I., Kelley, R. I., Somer, H., Straub, V., Mathews, K. D., Moore, S. A., and Campbell, K. P. "Post-translational disruption of dystroglycan–ligand interactions in congenital muscular dystrophies". In: *Nature* 418.6896 (2002), pp. 417–421.

Milner, R. "Implementation and applications of Scott's logic for computable functions". In: *ACM sigplan notices* 7.1 (1972), pp. 1–6.

Odersky, M., Spoon, L., and Venners, B. *Programming in Scala.* Artima Inc, 2008.

O'grady, A. *GitLab Quick Start Guide: Migrate to GitLab for all your repository management solutions.* Packt Publishing Ltd, 2018.

Okuda, S., Watanabe, Y., Moriya, Y., Kawano, S., Yamamoto, T., Matsumoto, M., Takami, T., Kobayashi, D., Araki, N., Yoshizawa, A. C., Tabata, T., Sugiyama, N., Goto, S., and Ishihama, Y. "jPOSTrepo: an international standard data repository for proteomes". In: *Nucleic Acids Research* 45.D1 (2017), pp. D1107–D1111.

Perez-Riverol, Y., Bai, J., Bandla, C., García-Seisdedos, D., Hewapathirana, S., Kamatchinathan, S., Kundu, D., Prakash, A., Frericks-Zipper, A., Eisenacher, M., Walzer, M., Wang, S., Brazma, A., and Vizcaíno, J. "The PRIDE database resources in 2022: a hub for mass spectrometry-based proteomics evidences". In: *Nucleic Acids Research* 50.D1 (2022), pp. D543–D552.

Perkins, D. N., Pappin, D. J., Creasy, D. M., and Cottrell, J. S. "Probability-based protein identification by searching sequence databases using mass spectrometry data". In: *ELECTROPHORESIS: An International Journal* 20.18 (1999), pp. 3551–3567.

Pierce, B. C. *Types and Programming Languages.* MIT press, 2002.

Ranzinger, R., Aoki-Kinoshita, K. F., Campbell, M. P., Kawano, S., Lütteke, T., Okuda, S., Shinmachi, D., Shikanai, T., Sawaki, H., Toukach, P., Matsubara, M., Yamada, I., and Narimatsu, H. "GlycoRDF: an ontology to standardize glycomics data in RDF". In: *Bioinformatics* 31.6 (2015), pp. 919–925.

Reese, W. "Nginx: the high-performance web server and reverse proxy". In: *Linux Journal* 2008.173 (2008), p. 2.

Richard-Foy, J. *Play framework essentials.* Packt Publishing Ltd, 2014.

Roestenburg, R., Williams, R., and Bakker, R. *Akka in action.* Simon and Schuster, 2016.

Rohloff, K., Dean, M., Emmons, I., Ryder, D., and Sumner, J. "An evaluation of triple-store technologies for large data stores". In: *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops: OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, OTM Academy Doctoral Consortium, MONET, OnToContent, ORM, PerSys, PPN, RDDS, SSWS, and SWWS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings, Part II.* Springer. 2007, pp. 1105–1114.

Rojas-Macias, M. A., Mariethoz, J., Andersson, P., Jin, C., Venkatakrishnan, V., Aoki, N. P., Shinmachi, D., Ashwood, C., Madunic, K., Zhang, T., Miller, R. L., Horlacher, O., Struwe, W. B., Watanabe, Y., Okuda, S., Levander, F., Kolarich, D., Rudd, P. M., Wuhrer, M., Kettner, C., Packer, N. H., Aoki-Kinoshita, K. F., Lisacek, F., and Karlsson, N. G. "Towards a standardized bioinformatics infrastructure for *N*-and *O*-glycomics". In: *Nature communications* 10.1 (2019), p. 3275.

Sharma, V., Eckels, J., Schilling, B., Ludwig, C., Jaffe, J. D., MacCoss, M. J., and MacLean, B. "Panorama public: a public repository for quantitative data sets processed in skyline". In: *Molecular & Cellular Proteomics* 17.6 (2018), pp. 1239–1244.

Sinha, A., Hussain, A., Ignatchenko, V., Ignatchenko, A., Tang, K. H., Ho, V. W., Neel, B. G., Clarke, B., Bernardini, M. Q., Ailles, L., and Kislinger, T. “N-Glycoproteomics of patient-derived xenografts: a strategy to discover tumor-associated proteins in high-grade serous ovarian cancer”. In: *Cell Systems* 8.4 (2019), pp. 345–351.

Skehel, J. J. and Wiley, D. C. “Receptor binding and membrane fusion in virus entry: the influenza hemagglutinin”. In: *Annual review of biochemistry* 69.1 (2000), pp. 531–569.

Struwe, W. B., Agravat, S., Aoki-Kinoshita, K. F., Campbell, M. P., Costello, C. E., Dell, A., Feizi, T., Haslam, S. M., Karlsson, N. G., Khoo, K.-H., Kolarich, D., Liu, Y., McBride, R., Novotny, M. V., Packer, N. H., Paulson, J. C., Rapp, E., Ranzinger, R., Rudd, P. M., Smith, D. F., Tiemeyer, M., Wells, L., York, W. S., Zaia, J., and Kettner, C. “The minimum information required for a glycomics experiment (MIRAGE) project: sample preparation guidelines for reliable reporting of glycomics datasets”. In: *Glycobiology* 26.9 (2016), pp. 907–910.

Sud, M., Fahy, E., Cotter, D., Brown, A., Dennis, E. A., Glass, C. K., Merrill Alfred H., J., Murphy, R. C., Raetz, C. R. H., Russell, D. W., and Subramaniam, S. “LMSD: LIPID MAPS structure database”. In: *Nucleic Acids Research* 35.suppl_1 (2007), pp. D527–D532.

Taylor, C. F., Paton, N. W., Lilley, K. S., Binz, P.-A., Julian Jr, R. K., Jones, A. R., Zhu, W., Apweiler, R., Aebersold, R., Deutsch, E. W., Dunn, M. J., Heck, A. J., Leitner, A., Macht, M., Mann, M., Martens, L., Neubert, T. A., Patterson, S. D., Ping, P., Seymour, S. L., Souda, P., Tsugita, A., Vandekerckhove, J., Vondriska, T. M., Whitelegge, J. P., Wilkins, M. R., Xenarios, I., Yates, J. R., and Hermjakob, H. “The minimum information about a proteomics experiment (MIAPE)”. In: *Nature biotechnology* 25.8 (2007), pp. 887–893.

Thaysen-Andersen, M. and Packer, N. H. “Advances in LC–MS/MS-based glycoproteomics: Getting closer to system-wide site-specific mapping of the *N*-and *O*-glycoproteome”. In: *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics* 1844.9 (2014), pp. 1437–1452.

The UniProt Consortium. “UniProt: the universal protein knowledgebase in 2023”. In: *Nucleic Acids Research* 51.D1 (2023), pp. D523–D531.

Tilkov, S. and Vinoski, S. “Node. js: Using JavaScript to build high-performance network programs”. In: *IEEE Internet Computing* 14.6 (2010), pp. 80–83.

Tipton, K. F., Armstrong, R. N., Bakker, B. M., Bairoch, A., Cornish-Bowden, A., Halling, P. J., Hofmeyr, J.-H., Leyh, T. S., Kettner, C., Raushel, F. M., Rohwer, J., Schomburg, D., and Steinbeck, C. “Standards for Reporting Enzyme Data: The STRENDA Consortium: What it aims to do and why it should be helpful”. In: *Perspectives in Science* 1.1-6 (2014), pp. 131–137.

Turing, A. M. “Computability and λ-definability”. In: *The Journal of Symbolic Logic* 2.4 (1937), pp. 153–163.

— “On computable numbers, with an application to the Entscheidungsproblem”. In: *Proceedings of the London Mathematical Society* s2-42 (1936), pp. 230–265.

Tyanova, S., Temu, T., and Cox, J. “The MaxQuant computational platform for mass spectrometry-based shotgun proteomics”. In: *Nature protocols* 11.12 (2016), pp. 2301–2319.

Urma, R.-G., Fusco, M., and Mycroft, A. *Java 8 in Action: Lambdas, Streams, and functional-style programming*. Manning Publications Co., 2014.

Van Wijk, K. J., Leppert, T., Sun, Q., Boguraev, S. S., Sun, Z., Mendoza, L., and Deutsch, E. W. “The Arabidopsis PeptideAtlas: harnessing worldwide proteomics data to create a comprehensive community proteomics resource”. In: *The Plant Cell* 33.11 (2021), pp. 3421–3453.

Varki, A., Cummings, R. D., Aebi, M., Packer, N. H., Seeberger, P. H., Esko, J. D., Stanley, P., Hart, G., Darvill, A., Kinoshita, T., Prestegard, J. J., Schnaar, R. L., Freeze, H. H., Marth, J. D., Bertozzi, C. R., Etzler, M. E., Frank, M., Vliegenthart, J. F., Lütteke, T., Perez, S., Bolton, E., Rudd, P., Paulson, J., Kanehisa, M., Toukach, P., Aoki-Kinoshita, K. F., Dell, A., Narimatsu, H., York, W., Taniguchi, N., and Kornfeld, S. “Symbol Nomenclature for Graphical Representations of Glycans”. In: *Glycobiology* 25.12 (2015), pp. 1323–1324.

Varki, A., Cummings, R. D., Esko, J. D., Stanley, P., Hart, G. W., Aebi, M., Mohnen, D., Kinoshita, T., Packer, N. H., Prestegard, J. H., Schnaar, R. L., and Seeberger, P. H. “Essentials of Glycobiology [internet]”. In: (2022). URL: https://pubmed.ncbi.nlm.nih.gov/35536922/.

Watanabe, Y., Aoki-Kinoshita, K. F., Ishihama, Y., and Okuda, S. “GlycoPOST realizes FAIR principles for glycomics mass spectrometry data”. In: *Nucleic Acids Research* 49.D1 (2021), pp. D1523–D1528.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J., Groth, P., Goble, C., Grethe, J. S., Heringa, J., Hoen, P. A. ’t, Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., Schaik, R. van, Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., Lei, J. van der, Mulligen, E. van, Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., and Mons, B. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific data* 3.1 (2016), pp. 1–9.

Yamada, I., Campbell, M. P., Edwards, N., Castro, L. J., Lisacek, F., Mariethoz, J., Ono, T., Ranzinger, R., Shinmachi, D., and Aoki-Kinoshita, K. F. “The glycoconjugate ontology (GlycoCoO) for standardizing the annotation of glycoconjugate data and its application”. In: *Glycobiology* 31.7 (2021), pp. 741–750.

Yamada, I., Shiota, M., Shinmachi, D., Ono, T., Tsuchiya, S., Hosoda, M., Fujita, A., Aoki, N. P., Watanabe, Y., Fujita, N., Angata, K., Kaji, H., Narimatsu, H., Okuda, S., and Aoki-Kinoshita, K. F. “The GlyCosmos Portal: a unified and comprehensive web resource for the glycosciences”. In: *Nature Methods* 17.7 (2020), pp. 649–650.

York, W. S., Agravat, S., Aoki-Kinoshita, K. F., McBride, R., Campbell, M. P., Costello, C. E., Dell, A., Feizi, T., Haslam, S. M., Karlsson, N., Khoo, K.-H., Kolarich, D., Liu, Y., Novotny, M., Packer, N. H., Paulson, J. C., Rapp, E., Ranzinger, R., Rudd, P. M., Smith, D. F., Struwe, W. B., Tiemeyer, M., Wells, L., Zaia, J., and Kettner, C. “MIRAGE: the minimum information required for a glycomics experiment”. In: *Glycobiology* 24.5 (2014), pp. 402–406.

York, W. S., Mazumder, R., Ranzinger, R., Edwards, N., Kahsay, R., Aoki-Kinoshita, K. F., Campbell, M. P., Cummings, R. D., Feizi, T., Martin, M., Natale, D. A., Packer, N. H., Woods, R. J., Agarwal, G., Arpinar, S., Bhat, S., Blake, J., Castro, L. J. G., Fochtman, B., Gildersleeve, J., Goldman, R., Holmes, X., Jain, V., Kulkarni, S., Mahadik, R., Mehta, A., Mousavi, R., Nakarakommula, S., Navelkar, R., Pattabiraman, N., Pierce, M. J., Ross, K., Vasudev, P., Vora, J., Williamson, T., and Zhang, W. “GlyGen: Computational and Informatics Resources for Glycoscience”. In: *Glycobiology* 30.2 (2020), pp. 72–73.

Zammetti, F. *Modern Full-Stack Development: Using TypeScript, React, Node. js, Webpack, and Docker*. Springer, 2020.

Zhang, Q., Ma, C., Chin, L.-S., and Li, L. “Integrative glycoproteomics reveals protein N-glycosylation aberrations and glycoproteomic network alterations in Alzheimer’s disease”. In: *Science Advances* 6.40 (2020), eabc5802.

Zhang, W. and Edwards, N. J. “GNOme–Glycan Naming and Subsumption Ontology”. In: *Proceedings http://ceur-ws. org ISSN* 1613 (2021), p. 0073.

List of Figures

List of Tables

List of Listings

Appendix A

A.1 Source code

The source code related to each data repository developed in this thesis was uploaded onto GitLab Community Edition (O'grady, 2018) running on an on-premises server at the Kinoshita Lab (https://gitlab.glyco.info). The following sections include the URL(s) of the source code for each data repository.

A.1.1 GlyComb

The major components constituting GitComb, including the user interface, the public and private APIs running on the server, the database configuration files, and all utility software, are collectively maintained at the following URL:

https://gitlab.glyco.info/glycosmos/glycombgroup

A.1.2 UniCarb-DR

All source code for UniCarb-DR, including the user interface, server-side logic, database configuration files, etc., is maintained at the following URL:

https://gitlab.glyco.info/glycosmos/unicarb-app-mirage

A.1.3 GlycoPOST

The main source code for GlycoPOST, including the user interface, APIs running on the server, and database configuration files, excluding the user login system, is maintained at the following URL:

https://gitlab.glyco.info/glycosmos/glycopost/glycopost

The GlycoPOST user login system, including the user interface, APIs running on the server, and database configuration files, is maintained at the following URL:

`https://gitlab.glyco.info/glycosmos/glycopost/gpdr-user`

In addition, the source code for the MiniCarb-Viewer component for visualizing GlycoWorkbench files on GlycoPOST is maintained at the following URL:

`https://gitlab.glyco.info/glycosmos/unicarb-dr/minicarb-viewer`