UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project
Final Paper 2022

Title: Design and implementation of a usable, extensible and general database for microbial carbohydrates (Glycan3DB)

Author: Joseph Sidley

Project Abbreviation: SugarToo

Supervisor(s): Michelle Kuttel

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	20
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	0
System Development and Implementation	*0*	*20*	20
Results, Findings and Conclusions	*10*	*20*	10
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Design and implementation of an extensible and general database for microbial carbohydrates (Glycan3DB)

SugarStore 2.0

Joseph Sidley
Department of Computer Science
University of Cape town
Cape Town, South Africa
SDLJOS001@myuct.ac.za

ABSTRACT

Glycan database systems are necessary for the understanding, data storage and classification of microbial carbohydrates. Microbial carbohydrates are abundant and increasingly used in vaccine development. Current systems are either built for specific carbohydrates or are not easy to use and navigate. This project explores the design and development of a usable and extensible database system called Gycan3DB that stores data on potentially any microbial carbohydrate. This data is visualised in three different ways (string, 2D image and 3D visualisation) and other important information that is vital for deeper research is also present in the database. This project also discusses the search and filter functions and options in the database system, and the methods, techniques and testing strategies used to make the database system usable, useful and reliable.

CCS CONCEPTS

- **Applied Computing** → Life and medical sciences → Computational biology → Molecular structural biology
- **Software and its engineering** → Software creation and management → Designing software → Software design engineering
- **Information Systems** → Data Management Systems → Database Design and Models

KEYWORDS

Glycan, Carbohydrate, Database, Web Development

1 INTRODUCTION

At present, molecular database systems are incredibly important in supporting the research into microbial carbohydrates. However, many of the currently available options in glycomics are littered with issues in both their functionality, and their usability.

Microbial carbohydrates are so important to research as they are present on the shell of many disease-causing viruses and bacteria [2]. They also play a role in cancer metastasis [6]. Understanding and researching them has potential impacts in drug and vaccine research at both an academia and an industry level [2]. Having a store of information on their structure is a functional and useful way of speeding up research, through linking to more in depth scientific papers. A user can also categorise the glycans in specific ways through search and filter functions in order to find a group of glycans with similar characteristics.

This system is important specifically because of the data that is planned to be encapsulate in it. As a base, this database was filled with data on *Klebsiella pneumoniae* K (capsular) antigens. *Klebsiella pneumoniae* strains are known to cause sepsis, pneumonia, meningitis and other serious infections [16]. This pathogen has become antibiotic resistant. The current approach of treatment is antibiotics which no longer works, sparking a move to carbapenems – another type of antimicrobial agent. This too has been met with mutations resulting in carbapenem resistance, and thus a new approach is based on vaccination, specifically using these K-antigens [16][17]. Work on this is being done at UCT as well as around the world, with the Gates Foundation and GlaxoSmithKline giving funding for this. In the process of discovery of strains, researchers need to know if these fit into the currently known strains or if these are new strains that must be sequenced. Having a searchable database helps speed up this process and help researchers in their fight against this pathogen. Other data on *Escherichia coli* has started to be added, and this bacterium is similarly important in its disease-causing properties and the potential of its use of microbial carbohydrate based vaccines.

In glycomics, the structure of the molecule defines the function of the molecule, and thus having different ways to present and represent their structure in one place allows for scientists to better understand the function of the carbohydrate.

Carbohydrates can be represented and categorised in several ways. A text based "chemical" representation is widely used, but this does not allow for easy visualization and thus 2D "Symbols Nomenclature for Glycans (SNFG)" [1] visualisations are also used. Finally, 3D "ball and stick" representations can be used to understand the structure of the molecule in space, with bond angles being present in this.

Some work on this has been done. Glycan databases exist for some viruses and bacteria [10][13][14], but these do not have all the information needed for the full understanding of the glycan. Most of these databases do not host 3D representation integration, with some of them not even having 2D representations. Furthermore, the current systems have issues in both functionality and usability. Even the most well-versed scientists find these systems confusing to use and navigate or cannot use them at all due to the broken software elements present in many of the options available today. Issues with nomenclature standardisation are common, the necessary data for full understanding of the carbohydrate is often missing and redirection to other sites or other software packages make navigation a hard task as not every function needed is in one place. Some examples of databases that have one or more of these problems are PolySacDB [11] and the Carbohydrate Structure Database [12]. These databases attempt to be useful for every type of antigen from any species, but they either do not display the correct information, do not display the correct types of information, have broken software, or are extraordinarily hard to navigate. They also do not accept community input for their data, which makes the website's usefulness entirely reliant on which antigens the developers decide to add, and this base of antigens is often stagnant and not improved upon.

Another common issue is that these systems are not built with extensibility and generality in mind. New microbial carbohydrates are being continually discovered, and thus extensibility should be an important consideration of any microbial carbohydrate database. Many of these databases are focused on one specific species, or type of antigen. Examples of this include EK3D [13] and ECODAB [10] which focus on *Escherichia coli* antigens, with EK3D focusing on K-antigens and ECODAB focusing on the O-antigens. These database dashboards are useful in a small context, however need to be extensible and broader to be useful for anyone in the glycomics field. This is in vast contrast to proteomics, where the gold standard of the Protein Data Bank [4] has data on many species, is extensible through the community, and each data entry has all the data sections that a scientist might need including 3D visualisation.

Glycan3DB was developed with these issues in mind, and is an attempt to build a usable, general, simple and extensible interface that will help users to research and better understand glycans. It intends to be a webtool that can be used by academics, students and people in industry alike – and to be a base for further development and addition. The aim is for it to be the first iteration of a Protein Data Bank style database for glycans, having all the data necessary for full understanding of the carbohydrate, in the sense of visualisation, and having correct and important information that is searchable in an easy to use and easy to navigate way.

2 APPROACH

2.1 Prototyping and iterative strategy

Due to the time constraints of the project, the implementation strategies chosen were important in order to make sure that all project goals were met and fully developed to the satisfaction of the supervisor.

2.2 The choice of Rapid Application Development

Early in the project it was agreed that the Rapid Application Development (RAD) methodology [8] would be the best possible development strategy for this project. This methodology prioritises quick prototyping, an iterative approach, and a lot of feedback. The focus was on utilising the short development period as efficiently as possible, by rapidly developing features and modules in a less structured manner, while making sure that the client (this project's supervisor Professor Michelle Kuttel) was still satisfied and had the ability to change the requirements throughout the process. The feedback given by the supervisor was weekly, and quite detailed and thus these short development lifecycles suited the meetings, the frequency of changes, and the intensity of the development process.

2.3 Requirements gathering

At first, a zoom meeting took place with the supervisor and a colleague in the Chemistry department of the University of Cape Town to discuss the needs, in terms of functional and non-functional requirements, involved in the project. As two experts in this particular field, and potential users of the system, the suggestions they gave were very helpful in determining what exactly would be needed in the database which is outlined in the requirements (both functional and non-functional). They also helped in deciding the user interface aspects needed in this project, and what design philosophies to follow. This gave a base to start the programming and prototyping phase of the project. Starting a project with defining the requirements is one of the tenets of RAD.

3 SYSTEM DESIGN

3.1 Non-functional Requirements

3.1.1 Speed

The speed of the system was very important as a non-functional requirement. The main table/home page of the database loads extremely quickly as very few modules have to be loaded, and the HTML and CSS code is basic enough to not incur speed limitations, along with the PHP request to the database being fast. Some concessions had to be made with the 3D visualisation aspect, as the loading of the JSmol module is by nature quite slow. This is however on the detailed “infocard” section of the system and thus will only be loaded if specifically needed by the user.

3.1.2 Ease of use

Simplicity and usability of Glycan3DB was seen as an integral requirement. Specific user-friendly design practices, outlined in 3.3 were decided upon. The website structure is also simple, with little redirection to new webpages in order to make sure that users have a pleasant user experience, do not have trouble navigating the dashboard, and have ease in going back and fixing any possible user errors. This means that most of the important data is all hosted on the main webpage.

3.1.3 Notation standardisation

It was important, in order to minimise user error and in order to keep the same standard across the database, to make sure notation was standardised. This was achieved by using the industry standard of CASPER notation [7] for the string representation, Symbol Nomenclature for Glycans (SNFG) [1] for the 2D representation and ball and stick for the 3D representation. However, CASPER notation uses α, β and → in the notation, and thus this had to be changed to a, b and -> in the data entries of the database. This is communicated to the user above the search bar.

3.2 Functional Requirements

3.2.1 Search by substructure

The first and most obvious functional requirement for Glycan3DB was the search function. Choosing what to be able to search by was not a difficult decision, and almost immediately from conception it was decided that the search option should be a search of antigen chemical structures (or substructures) in string format. Other options were prototyped, such as search functions for every column, but it was decided that the user would only need to search by substructures of antigen chemical representation. The string data entries are in CASPER [7] notation. This allows the user to find differences and similarities between antigens easily and accurately. As discussed in the introduction, this is important as the structure of molecules defines their function, and thus working out which molecules share some substructures can assist a researcher in ascertaining similarities between how molecules function. Through testing, this search function was altered to become more powerful, and this is explained in 7.2.

3.2.2 Dropdown menu for species origin

Due to the small number of species in the database in this project's iteration – a decision was made to use a drop-down menu to filter between species. This menu would have an option for each species and an option for all. It would have to be extended when new species were added but it would be very simple to achieve this and thus it was decided this was a good way to balance ease of use, database size, and user experience – as well as differentiating the species filter with the structure search in order to not confuse users.

3.2.3 2D structure

A 2D SNFG [1] representation of the antigen was very important for the table data. Similarly, to the string based chemical representation, it is a way of visualising the structure of the carbohydrate and understood well by people in the field of glycomics. It was decided that the images, which were sourced from K-PAM [14], should be shown in the table on the home page. This slows down the website slightly as it has to load the images, however it is such important data that it should be easily accessible without having to move into the infocard section, and this load time is negligable. Sourcing these images from K-PAM is not ideal, but it was decided that as Glycan3DB will originally only be used internally this would be okay.

3.2.4 3D structure and infocards

3D structure, once again is another way of visualising the data in the database and visualising a molecule in 3D space. This visualisation can be achieved using a package called Jmol/JSmol [15]. It allows the user to zoom into and rotate the 3D structure of the molecule. These 3D visualisations were to be loaded into a separate infocard for every antigen. The JSmol package can present a molecule as long as it is provided with a .pdb (protein data bank) file. This file is the standard for providing coordinates in space for the 3D molecule. This infocard would be a dedicated html page for each antigen which hosted all data about the antigen, including the 3D visualisation and other peripheral data that could not be stored in the table. These files were sourced by me, through K-PAM [14] and EK3D [13], and are loaded into each infocard separately. The reason they are not present in the table section, is that the loading time for the 3D visualisation is relatively slow and having a large enough container for them is important in making them as readable and thus useful as possible.

3.2.5 Extensibility through suggestions

It was important for Glycan3DB to have to ability to be built upon by the users of the database. Thus, it was a requirement of the system to have a section in which users could suggest new antigens for the database. This suggestion page would then send an email with the data to the owner of the website/MySQL database to inspect and possibly integrate into the website. This means that the website and its incorporated data is developing and evolving with the community's needs, as no developer will have all the information on every possible antigen, and new antigens are being continually discovered.

3.2.6 Database column header choices

The first batch of data used in Glycan3DB was mostly collected by a Chemistry Masters student named Siwaphiwe Mfana for their thesis. It is a database of *Klebsiella Pneumoniae* K-antigens, and this is the basis of most of the data in this database. *Escherichia coli* data was then collected by me and added to the database. The *E coli* data in the database is very sparse, as *Klebsiella pneumoniae* was focused on in this iteration, and *E coli* was added as a proof of concept of having many species share the same database. The 2D and 3D visualisations for both *E coli and K pneumoniae* were collected by me. This data, hosted in an excel spreadsheet, had many data entries and some data entries did not need to be part of the Glycan3DB table. It was decided that the columns of the table should show the Antigen ID, the Chemical Representation, the 2D Representation, the number of repeating units, a link to the paper where the antigen is discussed and the species of origin of the antigen. All of these choices were seen as being significant to anyone looking to get an overview of the antigen, or to research the antigen further. The Boolean branched factor, the references of the paper where the antigen is discussed, some Nuclear Magnetic Resonance data (where NMR data is available) and the 3D visualisation were left to be details on demand in the infocard section.

3.3 Design Principles and Schneiderman's Mantra

Ben Schneiderman's Mantra [9] was integral to how the project was planned and built from the very beginning. The three rules in his mantra were seen as being of utmost importance. This mantra is mostly used in the sphere of visualisation however in Glycan3DB, the mantra was followed throughout the entire development process.

3.3.1 Minimise clicking/details on demand

From the first coding stage of this project, it was decided that minimising clicking was one of the most important design principles followed. Previous attempts at database dashboards similar to Glycan3DB were cumbersome and hard to use, especially for those who had minimal experience in the field or using these systems. In order to navigate the useful and needed data, a user had to go through many confusing landing and transition pages. Making sure that the required information was mostly on the home page of the website was vital in making sure that the user experience was straightforward, easy to use and fast. Thus, it was decided to make the main database table on the home page, so a user can start searching immediately as they open Glycan3DB.

Another way that clicking was minimised in Glycan3DB was the alteration of the database to use the letters a and b and the symbol -> rather than α, β and → respectively. As users do not have these symbols on their keyboard, the only way to get these symbols would be to click on buttons to fill in the search box. Thus, the data in the table was altered and users were given instructions to search using the keyboard characters.

When more details were required for each antigen, an "infocard" webpage was designed which hosted the 3D visualisation as well as the references of the paper from where the antigen was discussed and whether an antigen was branched. These details would probably not be needed in many cases, and were slow to load and so leaving the option of choosing to see this data to the user as a "detail on demand" made more sense in Glycan3DB.

3.3.2 Zoom and filter

Giving a user the ability to decide which data they want to be displayed is another of the Schneiderman rules. This, of course, is a requirement of any database application but deciding which options the user needs most was a process which led us down the path to deciding the functional requirements explained earlier. The user will never need to search by link or by number of repeating units. They will also have few options in terms of species, which is why Glycan3DB settled on a substructure search function and a species drop down menu to filter the table. This gives the user the best filtering options without over complicating the UI of the webpage.

3.3.3 Overview

One of the main complaints of other similar database systems was the fact that on the opening of the website, a user had no immediate indication of what data was available for which antigens on which species. This, it was decided, can be fixed by following the Schneiderman "Overview" rule. Therefore, this project implemented an information carousel which would give an overview and a visualisation of the information stored within the database. This carousel would have a card for each species, and explain how many and which type of antigens of that species are contained in the database. This then gives a user a base of knowledge of what is in the database before they start searching through the database. This would have to be updated with new information cards if the number of species increased or the number of antigens per species increased.

3.4 Other Design Principles Followed

In the planning of this project, another commitment was made to following other design principles that were created for the development of scientific software. These including *Understanding User Motivations, Using Standard Notations, Iterating and Testing Often* and *Solving the Right Problem First [3].* Following these principles meant that every decision made had to be made with these in mind. Notations had to be standardised from the beginning of the project (as discussed in 2.1.3) and the end user's needs and motivations must be thought of in every stage of the development process. Finally, the use of iterations and having the constant ability to change in response to client and user needs was paramount, and strategies such as RAD were implemented to follow these principles.

4 IMPLEMENTATION

4.1.1 Paper Prototype

On arrival back to Cape Town after the mid year break, a meeting was held with the supervisor in which the use and necessity of paper prototypes was discussed in order to complete the requirements gathering phase. The chosen paper prototype is shown in *Figure 1* and other iterations are attached to the end of this document. This part of the project was to ensure that the supervisor and I were on the same page about the ideal structure of Glycan3DB both functionally and in terms of usability and aesthetic. The paper prototype's function mainly gave a base in order to focus on usability and UI structure. Once these paper prototypes were fleshed out, altered and decided upon, the first iteration of the coding phase was started, which was designed according to the paper prototype's specifications.

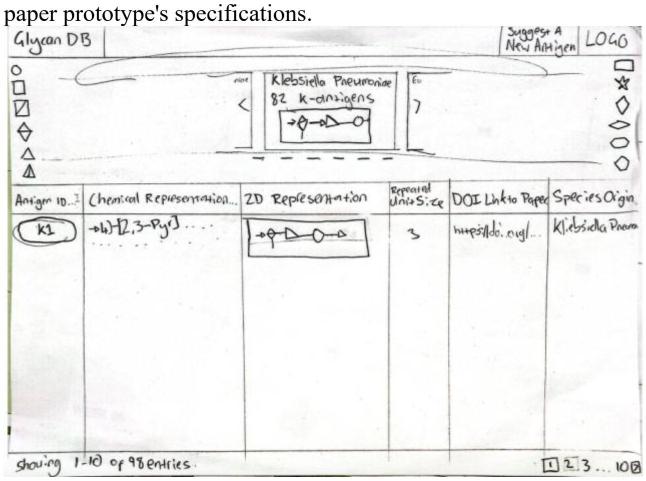
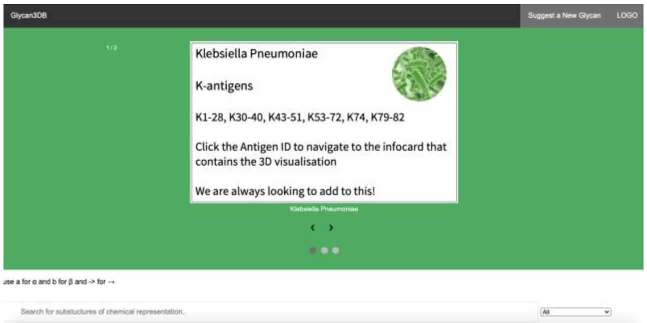


Figure 1: Paper prototype chosen. This was altered with a dropdown menu for species and the search above the table rather than in the table headers.

4.1.2 First prototype phase

This programming stage involved the connection of the MySQL database to the HTML table through PHP, the integration of the 3D visualisations to the infocards, the drop down menu functionality and a general integration of all of the website pieces and modules built so far during the mid-year break. The 3D visualisation integration went through multiple stages, with JSmol, 3Dmol.js and Molstar all tested; however it was decided that JSmol was the best option, due to the ease in which a user of the package could incorporate their own .pdb files. It also involved a lot of front end coding to try and come as close as possible to the paper prototype previously agreed upon. This stage was seen as a first iteration phase as changes would have to be made after testing, however it was not one iteration on its own, but rather a phase of many smaller weekly iterations. It was added to and changed over the weeks of development, and thus followed the rules of RAD. Figures of how this first iteration looked are below. This iteration on the main page incorporated a navigation bar with a logo and the suggestion button, a table filled with all the data in the MySQL database, the dropdown filter function, a search bar for carbohydrate substructures and the overview carousel. This stage was built using a web server called XAMPP which was necessary for building a correct file structure and using languages such as PHP to connect to the database. In this iteration, the infocard design and file structure was being finalised and so the links of the antigen ID to the infocard are not functional yet.



Klebsiella Pneumoniae

K-antigens

K1-28, K30-40, K43-51, K53-72, K74, K79-82

Click the Antigen ID to navigate to the infocard that contains the 3D visualisation

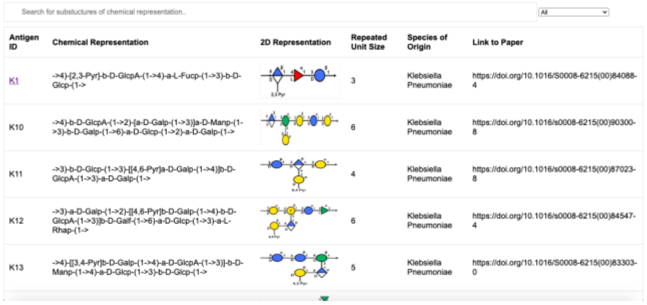
We are always looking to add to this!

Klebsiella Pneumoniae

use a for α and b for β and -> for →

Search for substuctures of chemical representation..

Figure 2: Navigation bar, information carousel, search bar and drop down filter functionality.

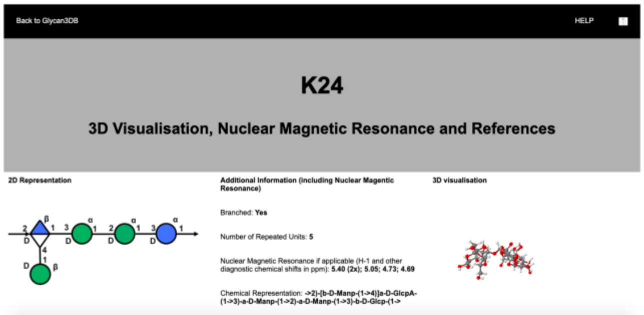

Search for substuctures of chemical representation..

Antigen ID	Chemical Representation	2D Representation	Repeated Unit Size	Species of Origin	Link to Paper
K1	->4)-[2,3-Pyr]-b-D-GlcpA-(1->4)-a-L-Fucp-(1->3)-b-D-Glcp-(1->	2,3-Pyr	3	Klebsiella Pneumoniae	https://doi.org/10.1016/S0008-6215(00)84088-4
K10	->4)-b-D-GlcpA-(1->2)-[a-D-Galp-(1->3)]a-D-Manp-(1->3)-b-D-Galp-(1->6)-a-D-Glcp-(1->2)-a-D-Galp-(1->		6	Klebsiella Pneumoniae	https://doi.org/10.1016/s0008-6215(00)90300-8
K11	->3)-b-D-Glcp-(1->3)-[[4,6-Pyr]a-D-Galp-(1->4)]b-D-GlcpA-(1->3)-a-D-Galp-(1->		4	Klebsiella Pneumoniae	https://doi.org/10.1016/s0008-6215(00)87023-8
K12	->3)-a-D-Galp-(1->2)-[[4,6-Pyr]b-D-Galp-(1->4)-b-D-GlcpA-(1->3)]b-D-Galf-(1->6)-a-D-Glcp-(1->3)-a-L-Rhap-(1->		6	Klebsiella Pneumoniae	https://doi.org/10.1016/s0008-6215(00)84547-4
K13	->4)-[[3,4-Pyr]b-D-Galp-(1->4)-a-D-GlcpA-(1->3)]-b-D-Manp-(1->4)-a-D-Glcp-(1->3)-b-D-Glcp-(1->		5	Klebsiella Pneumoniae	https://doi.org/10.1016/S0008-6215(00)83303-0

Figure 3: Database table with header column choices and search and drop down functionality. It was decided that the data should be in one large table rather than pages of 10 entries each such as in the paper prototype (Figures 2 and 3 are the same page).

4.1.3 Final database

Following the functionality and usability testing, a second iteration phase of the project was embarked upon using the information gathered from that. This stage involved creating the suggestion form and linking it to the server side with PHP. A fleshed out logo and help section were be added to the navigation bar, and colours were changed for usability optimisation. Infocard design was also finalised in this stage. Functionality errors were fixed, especially in the search function, and some usability and aesthetic changes were made. This phase, as laid out by RAD, is the phase where maintainability and optimisation are focused on, and especially, in the context of Glycan3DB, where usability is refined. This second iteration is the final iteration of this project and will be shown here. The specifics of the changes made are explained fully in the results section (Section 7). An image of the infocard design chosen is in *Figure 4* and *Figure 5*.



K24

3D Visualisation, Nuclear Magnetic Resonance and References

2D Representation

Additional Information (including Nuclear Magentic Resonance)

Branched: **Yes**

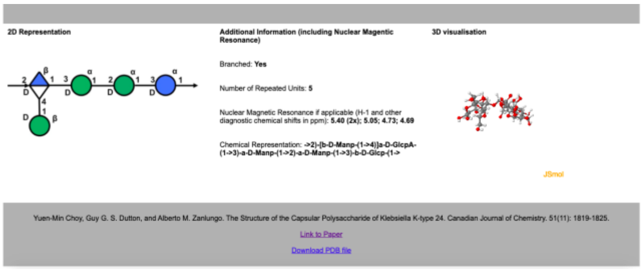
Number of Repeated Units: **5**

Nuclear Magnetic Resonance if applicable (H-1 and other diagnostic chemical shifts in ppm): **5.40 (2x); 5.05; 4.73; 4.69**

Chemical Representation: **->2)-[b-D-Manp-(1->4)]a-D-GlcpA-(1->3)-a-D-Manp-(1->2)-a-D-Manp-(1->3)-b-D-Glcp-(1->**

3D visualisation

Figure 4: The graphical user interface of the infocard for K24 showing the navigation bar, and information stored (including the 3D visualisation)



2D Representation

Additional Information (including Nuclear Magentic Resonance)

Branched: **Yes**

Number of Repeated Units: **5**

Nuclear Magnetic Resonance if applicable (H-1 and other diagnostic chemical shifts in ppm): **5.40 (2x); 5.05; 4.73; 4.69**

Chemical Representation: **->2)-[b-D-Manp-(1->4)]a-D-GlcpA-(1->3)-a-D-Manp-(1->2)-a-D-Manp-(1->3)-b-D-Glcp-(1->**

3D visualisation

Yuen-Min Choy, Guy G. S. Dutton, and Alberto M. Zanlungo. The Structure of the Capsular Polysaccharide of Klebsiella K-type 24. Canadian Journal of Chemistry. 51(11): 1819-1825.

Link to Paper

Download PDB file

Figure 5: This shows the footer with the link to the paper, and the download PDB file options which were added after testing (Figure 4 and 5 are the same page)

5 Code stack

5.1 Front end

5.1.1 CSS, HTML and the lack of UI frameworks

The front-end UI was built almost entirely using basic CSS and HTML to make sure that the code base was scalable and maintainable. It also allowed for the project to be customisable as it was not restricted by some a specific UI framework. In the early development stage of the project some UI frameworks were tested such as DataTables, ChocUI and MaterialUI but it was decided that this increased complexity and loading time and decreased customisability too much to be worth the ease in the development process. Finally, it was discussed that UI frameworks look very similar to many applications available today and thus would give the webtool a prescribed look, and make further developers follow a very specific design philosophy, which focuses on more mobile centred design. For this reason, UI frameworks were discarded from the project.

5.1.1 React and Bootstrap5

The Bootstrap5 CSS framework was originally used in the styling of the database application in order to enable some cleaner looking components such as the carousel element and table. It was later decided that these carousel elements and the table could look clean and run faster if created purely using CSS and HTML and thus Bootstrap5 was removed from the project. The choice to not use React was made in order to make sure that the code base was fast, scalable and did not use too much memory. If this project is to be taken further, it would make sense to stay away from React as different programmers use and organize React in vastly different ways. It was also important to me to use HTML and JavaScript separately rather than using JSX which I find to be a cumbersome language with weak documentation.

5.1.2 JavaScript

JavaScript was used as the client-side scripting language for the operation of the table "search by substructure" function, the by-species drop down menu and the control for the information carousel. Of course, JavaScript is used as the client-side scripting language in all websites, and it worked well in the development of Glycan3DB.

5.1.3 JavaScript (JSmol package)

JavaScript was used as the scripting language in order to load the 3D visualisations of each carbohydrate in the infocard section. This was programmed using Jmol/JSmol, which loads and facilitates the 3D visualisation integration. As mentioned earlier, this is quite a slow loading package, but it is very powerful, easy to integrate and is absolutely necessary for what Glycan3DB is trying to achieve.

5.2 Back end

5.2.1 MySQL and PostgreSQL

In the initial stage of the project and in the project plan, PostgreSQL was the preferred relational database management system due to its support for more complex data structures, however through the development process it was decided that that support was not needed in this implementation of Glycan3DB and the speed and ease of use of MySQL was a more important consideration for this project. MySQL has availability and scalability as core goals and having high runtime and an easily scalable backend on this application was important to us, which was one of the reasons MySQL was chosen. Its simple set up and fiercely open-source nature were also a consideration. MySQL is also a faster language than PostgreSQL, with a lot of community application support and thus for the purposes of this project was a perfect relational database management system.

5.2.2 PHP

PHP was used for the server side processing of Glycan3DB. As this project is predicated on its database connection, it was important that a suitable language was chosen. Using Node.JS in conjunction with Express was an option for this database connection and was prototyped but ultimately it was decided that the community support and long running nature of PHP meant that it was the most sustainable, scalable and safe option for Glycan3DB. PHP was also chosen due to it being completely platform independent, having a good reputation of speed, having easy integration and being supported by most major web servers [18].

5.2.2.1 PHP table connection

A PHP script is used in the HTML code in order to fetch the data from the MySQL database and display the data in the HTML table using the echo function in PHP. The SQL database is scrolled through with <tr>, </tr> (table row) and <td>, </td> (table data) tags added, and this is then compiled into an HTML string. PHP works incredibly well with databases so using PHP for this use case was the obvious choice.

5.2.2.2 PHP form connection

A PHP script is used in order to send the content of the suggestion forms to the developers or managers of the application through the use of an email. The script takes the user data, sanitizes it, and finally sends it to the email of the manager of the site (currently the developer). The current iteration of the project uses the mail() PHP function but when this project becomes live on UCT servers, this will be changed to run through the Secure Mail Transfer Protocol using the PHPMailer package.

6 EVALUATION

6.1 Developer code robustness testing

Testing the robustness of the code was mostly a test by inspection endeavour. A few tests such as SQL injection attacks in the form and search functions, and more basic functionality tests in all other functions were attempted. These included testing long strings in the search bar, using the dropdown menu and search function at once and sending forms with empty data inputs. This testing was done by me as the developer. On the other hand, I am not an expert in the chemistry sphere and thus I cannot test all functions necessary for the system. Furthermore, I am biased by my development of Glycan3DB and cannot fully test the usability of the project and thus recruiting usability testers was important to examine usability and maximise it. All ethical clearance considerations were understood and followed, as FRSEC and DSA clearance was granted.

6.2 Testing Strategy

6.2.1 Computational chemistry expert testing

As a beginner in the field of computational chemistry, it was imperative that some experts in the field were consulted in order to test if the functionality of the code was correct and that the data in the database was accurate. Professor Michelle Kuttel of the Computer Science department (the supervisor of this project), her colleague, a Chemistry professor, and a Computational Chemist PhD were asked to do this testing. The testers made sure that the data was accurate, the searching was done in the correct manner and that the 2D and 3D functionality and notations were correct. They also gave insight on usability and structure. This testing was relatively unstructured to give the testers full use of an almost complete system.

6.2.2 Student usability testing

For testing of usability, one round of user tests was conducted with students who study or have studied computer science. These were students who have knowledge of computer science usability best practices as they have done the Human Computer Interaction course or a similar course and some of them also have chemistry knowledge. Their names will remain anonymous due to the conditions set in the consent form. These user tests were also relatively unstructured, but they were moderated to ensure all functions are tested. This means that the students were asked to do specific tasks while the developer took notes on the issues that they encountered along the way. It is done this way so that the users can try the entire system, rather than testing specific aspects. Only two students were used in these tests due to the time pressure in development, and finding students with knowledge of these practices and this topic was a challenge. Recruitment went through the supervisor. In future iterations of this project, I hope that more testers can be found, and though this that the user interface can become more refined.

7 RESULTS AND DISCUSSION

7.1 Developer functionality testing

Testing done by me did not reveal any major exploits or issue in the code. All references pointing to a new page did their job correctly, the search function worked correctly as did the dropdown filter function. One error which was noted and then fixed was that while using the dropdown filter, if a user used the search function, then the search function did not keep the filter of the dropdown menu and searched the entire database. This was only an issue when first using the dropdown and then searching but not the other way around. The 3D visualisations, although slow, all rendered and kept to the size of the container which was a worry with the package when the molecules became larger. Finally, all of the elements needed to link to each other. At the point of the testing the file structure had not been finalized and thus moving from the main page to an infocard page and back had bugs in some infocards, as well as moving to and from the form page and the main page.

7.2 Expert Functionality Testing

Expert testing was performed on the 31st of August, and this testing was fruitful in understanding how to improve and finalise the project. Four main changes were requested. The most important of these was a changing of the search function. Where the original search function had only been able been able to search for one substring of the final structure, the experts said it would be extremely useful to be able to search for multiple strings at once. For example, a user may want to find every antigen with the monosaccharides "Galp" and "Manp", but not directly next to each other. Thus, functionality was added to separate search terms by a comma and make it possible to do this. The next change requested was the standardisation of image dimensions. Currently the 2D SNFG images were stretch to fit a 180px by 80px resolution, but for thinner, longer antigens, these dimensions compressed the image's width and for shorter, taller antigens these dimensions compressed the image's height. This decreased readability and thus the images had to keep their original dimensions. This was done by changing CSS settings to keep the image's original aspect ratios in this 180px by 80px container, as having standard heights for the table rows was also important for readability. Usability and professionalism aspects were focused on and noted. The infocard section was seen to be childish in its use of colours, and its formatting on the page and thus was changed to a grey and white colour scheme, and elements were shifted for a more minimalistic aesthetic and maximising usability. Finally, it was requested that the infocard section had a "download PDB file" option which was added. The home/table page was also slightly adjusted on the suggestion of the testers, with exclamation marks in the carousel removed and additions to the carousel UI made to make sure that a user knew there was more data in the carousel.

7.3 Student Usability Testing

Student testing was performed on the 1st of September and was extremely helpful in understanding how a user may use this system in the real world. The testers made mention of some similar usability aspects to the experts in the infocard section in terms of colour, layout and professionalism. They also mentioned that in the home/table page, figuring out where to search was slightly confusing. They suggested adding a search icon to make this a more obvious search bar. They also said that the "link to paper" column data needed to be clickable to take you to the paper, which was already being worked on. This was then completed. Other than this, the students said that they found the system easy to understand, easy to use, functionally correct and they could see themselves using this system in the future.

7.4 Portability

This code is relatively portable in that it will work on any recent browser and any recent operating system on a PC. This however was not particularly part of the system design as it is true of all websites built with only HTML, CSS and Javascript (with Javascript packages). It also has low load and thus will work on almost all processor hardware. It will also work equally on different screen sizes due to the <meta> attribute, meaning that the content will shift to the devices size. It will not however work as well on mobile, as it has not been developed for mobile and is more useful in a desktop or laptop scenario.

7.5 Maintainability

An evaluation of the systems maintainability shows that the system has the ability to be changed and upgraded with ease – due to the project's focus on extensibility. Incorporation of new data was focused on from the beginning. Any new antigens sent through the suggestions form, have to go through some cleaning and alterations and then can be added to the servers library of images and .pdb files, and added to the MySQL database. Then an infocard can be easily created for it by just changing the image file, .pdb file and other peripheral data into the prebuilt HTML and CSS template used for every other infocard. These modules are very reusable. Addition of a new antigen, including creation of the infocard, would take less than ten minutes. If another species is added, a new carousel element can be added very quickly and easily in the index2.php codebase. As far as UI upgrades go, the relative simplicity of the code base means that further additions to the codebase of Glycan3DB are not restrained by current UI packages and having to follow a predetermined design style. Using just CSS means that upgrades can be more specific, or a further developer could choose a separate design style and UI philosophy if needed. A user also will not be handicapped by having to use JSX if they do not want to use React.

8 CONCLUSIONS

8.1 Did the functionality meet the necessity goals?

Glycan3DB was originally built as a system to be used in real world applications. This project achieves that goal. When it is made live on the UCT servers, it will be available for all to use and will be useful for anyone wanting to have a searchable, usable, general database system. This project will be live on UCT servers at https://glycan3db.cs.uct.ac.za/ within 10 days of the project's completion and will be available for broader use. The goals lined out in the introduction and system design sections were all met to a satisfactory level. All functionality originally needed in this system has been implemented, and more has been added through the iterations of this project. The "suggest an antigen" feature was seen as extremely important in the conceptualisation of this project, and that works well which shows that the aim of extensibility was achieved. Having the input of professors and researchers as expert testers in this area of science was extraordinarily helpful in making sure that Glycan3DB was built in the way it needed to be built in order to help in the fight against diseases caused by *Klebsiella pneumonia, Escherichia coli* and other bacteria. The main extension in terms of functionality would be to add more species and more antigens in order to make this database even more useful for researchers. This should be an easier task with the focus on maintainability.

8.2 Did the usability meet the necessity goals?

Usability in scientific software is often hard to define, as it is such a subjective concept, and due to so many clashes between functionality and usability in these systems. The only way of truly understanding whether usability goals are met is through extensive testing. Testing was performed on Glycan3DB, however testing was not the primary focus of this iteration. Although the system may be seen as simple aesthetically, this style of UI was preferred for a system required mostly for its functionality. Systems similar to Glycan3DB were plagued with usability issues due to their complexity both in interface and in structure. The focus on well-known and proven usability practices throughout the design and implementation of Glycan3DB makes this project have a good base for usability before testing. Testing with potential users (both students and professionals) revealed that the simpler UI was usable and understandable for users of all levels, especially after usability changes were made. In the future, this usability testing should be extended, and should further maximise the cleanliness and usability of the UI. This would be a good starting point for any future iterations of Glycan3DB. Generally, this is seen as a usable, easily navigable system by the people tested, who would be potential users, and thus usability goals are seen to have been reached. This project has thus met both its usability and its functionality goals.

REFERENCES

[1] Ajit Varki, Richard D Cummings et al. 2015. Symbol Nomenclature for Graphical Representations of Glycans. *Glycobiology*, Volume 25, Issue 12 (Dec 2015), 1323–1324.DOI: https://doi.org/10.1093/glycob/cwv091

[2] Rena Astronomo, Dennis Burton. 2010. Carbohydrate vaccines: developing sweet solutions to sticky situations?. *Nat Rev Drug Discov*, Volume 9**,** 308–324 (Aprin 2010). DOI: https://doi.org/10.1038/nrd3012

[3] Felipe Albrecht, Peter Ebert and Markus List. (2017). Ten Simple Rules for Developing Usable Software in Computational Biology. *PLOS Computational Biology. Volume 13(1), 1-5.* https://doi.org/10.1021/acs.jctc.1c00169

[4] Helen Berman, John Westbrook, Zukang Feng et al. 2000. The Protein Data Bank, *Nucleic Acids Research*, Volume 28, Issue 1, 1 January 2000, 235–242 DOI: https://doi.org/10.1093/nar/28.1.235

[5] Lavanya Ramakrishnan and Daniel Gunter. (2017). Ten principles for Creating Usable Scientific Software. *IEEE 13TH International Conference on eScience* Auckland, New Zealand, 210-218. (Oct 2017). DOI: https://doi.org/10.1109/eScience.2017.34

[6] Campbell, M., Ranzinger, R. et al. 2014. Toolboxes For A Standardised And Systematic Study Of Glycans. *BMC Bioinformatics*, 15 (Suppl 1), S9. DOI: https://doi.org/10.1186/1471-2105-15-S1-S9

[7] A. Furevi, A. Ruda, et al. 2022. Complete ^{1}H and ^{13}C NMR chemical shift assignments of mono- to tetrasaccharides as basis for NMR chemical shift predictions of oligo- and polysaccharides using the computer program CASPER. *Carbohydrate Research*. Vol 513 (Mar 2022) DOI: https://doi.org/10.1016/j.carres.2022.108528

[8] Kerr, James M.; Hunter, Richard (1993). Inside RAD: How to Build a Fully Functional System in 90 Days or Less. McGraw-Hill. ISBN 0-07-034223-7.

[9] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations,” *Proceedings 1996 IEEE Symposium on Visual Languages*, Boulder, CO, USA, 1996, pp. 336-343, DOI: https://doi.org/10.1109/VL.1996.545307

[10] Miguel Rojas-Macias, Jonas Ståhle, Thomas Lütteke, Göran Widmalm. 2015. Development of the ECODAB into a relational database for *Escherichia coli* O-antigens and other bacterial polysaccharides, *Glycobiology*, Volume 25, Issue 3, March 2015, 341–345. DOI: https://doi.org/10.1093/glycob/cwu116

[11] Abhijit Aithal, Arun Sharma , Shilpy Joshi et al. 2012. PolysacDB: A Database of Microbial Polysaccharide Antigens and Their Antibodies. *PLOS ONE* Volume 7, Issue 4, April 2012, 1-4. DOI: https://doi.org/10.1371/journal.pone.0034613

[12] Ksenia Egorova, Anna Kondakova, Phyl Toukach. 2015. Carbohydrate Structure Database: tools for statistical analysis of bacterial, plant and fungal glycomes. *Database (Oxford)*. Volume 2015, September 2015. DOI: https://doi.org/10.1093/database/bav073

[13] Bharathi Reddy Kunduru, Sanjana Anilkumar Nair, Thenmalarchelvi Rathinavelan. 2015. EK3D: an *E. coli* K antigen 3-dimensional structure database. *Nucleic Acids Research*, Volume 44, Issue D1, January 2016, D675–D681. DOI: https://doi.org/10.1093/nar/gkv1313

[14] Patro, L.P.P., Sudhakar, K.U. & Rathinavelan, T. 2020. K-PAM: a unified platform to distinguish *Klebsiella* species K- and O-antigen types, model antigen structures and identify hypervirulent strains. *Sci Rep* **10**, 16732 (2020). DOI: https://doi.org/10.1038/s41598-020-73360-1

[15] Jmol: an open-source Java viewer for chemical structures in 3D. http://www.jmol.org/

[16] Rello, J., Kalwaje Eshwara, V., Lagunes, L. *et al.* 2019. A global priority list of the TOp TEn resistant Microorganisms (TOTEM) study at intensive care: a prioritization exercise based on multi-criteria decision analysis. *Eur J Clin Microbiol Infect Dis* **38**, 319–323 (2019). DOI: https://doi.org/10.1007/s10096-018-3428-y

[17] Assoni, L., Girardello, R., Converso, T.R. *et al.* 2021. Current Stage in the Development of *Klebsiella pneumoniae* Vaccines. *Infect Dis Ther* **10**, 2157–2175 (2021). DOI: https://doi.org/10.1007/s40121-021-00533-4

[18] Majida Laaziri, Khaoula Benmoussa, et al. 2019. A Comparative study of PHP frameworks performance, *Procedia Manufacturing,Volume* 32,2019,Pages 864-871,ISSN 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2019.02.295

APPENDIX:

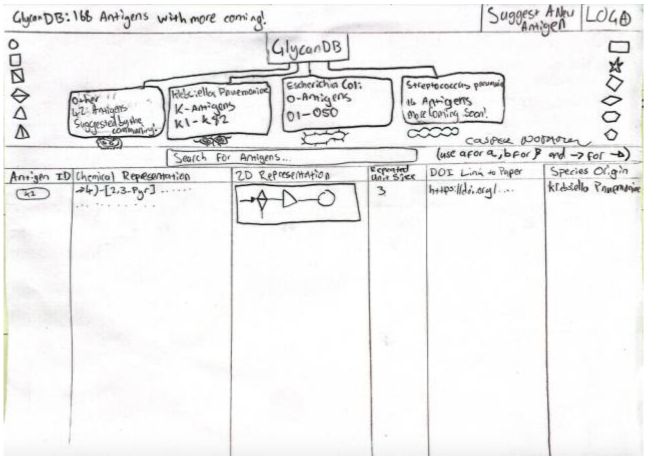

GlycanDB: 166 Antigens with more coming!
Suggest A New Antigen
LOGO
GlycanDB
Other 12 Antigens Suggested by the community
Klebsiella Pneumoniae K-Antigens K1-K82
Escherichia Coli: O-Antigens O1-O50
Streptococcus pneumoniae 16 Antigens More coming soon!
Search For Antigens...
(use a for α, b for β and -> for →)
Antigen ID
Chemical Representation
2D Representation
Repeated Unit Size
DOI Link to Paper
Species Origin
K1
→4)-[2,3-Pyr]......
3
https://doi.org/...
Klebsiella Pneumoniae

Figure 6: Another attempt at a paper prototype, with a different overview visualisation which was not chosen.

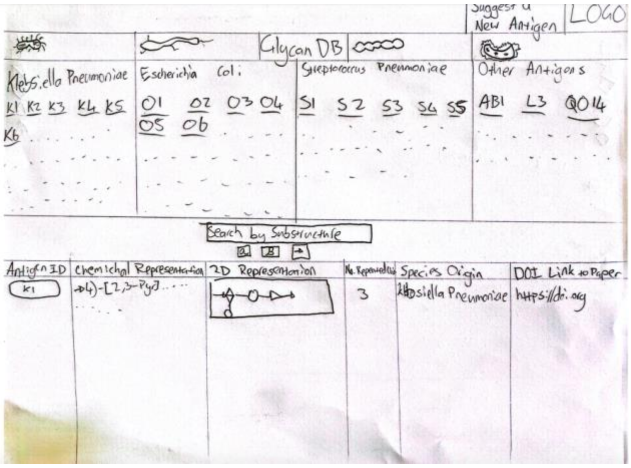
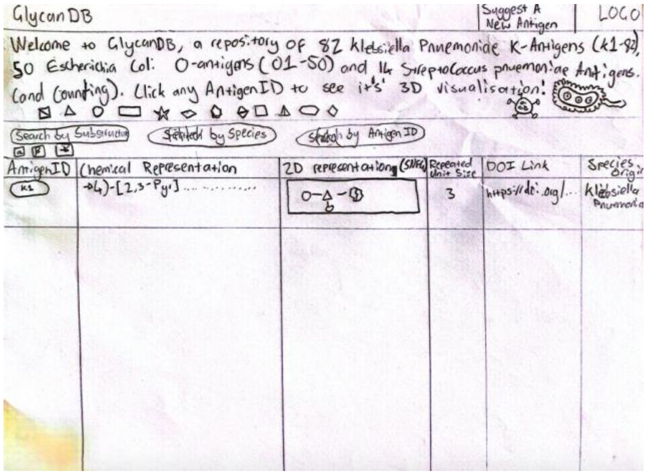
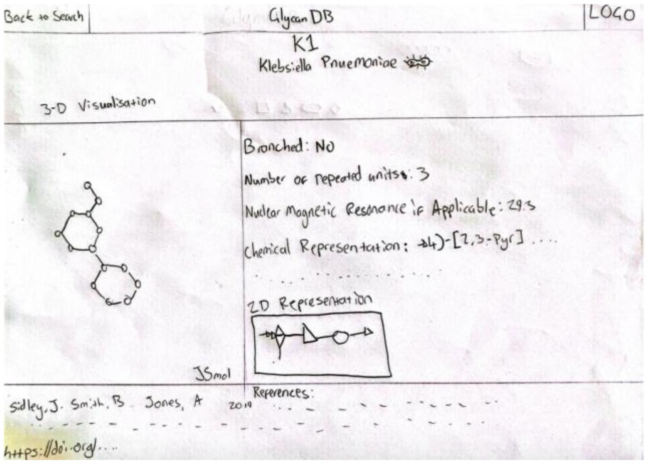


Figure 7: Unused paper prototype, visualisation overview and search function not preferred



Welcome to GlycanDB, a repository of 82 Klebsiella Pnuemonide K-Antigens (k1-82), 50 Escherichia Coli O-antigens (O1-O50) and 16 Streptococcus pnuemoniae Antigens. (and Counting). Click any AntigenID to see it's' 3D Visualisation!

Figure 8: Unused paper prototype, overview too clunky. Supervisor notes seen on search function for search by species saying "filter", meaning change to a dropdown function



Branched: No

Number of repeated units: 3

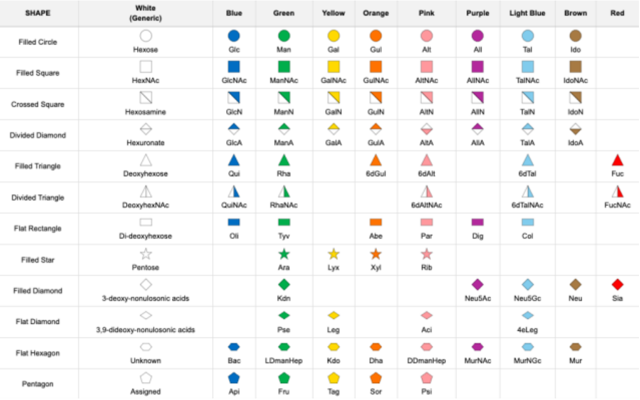
Nuclear Magnetic Resonance if Applicable: 29.3

Chemical Representation: →4)-[2,3-Pyr] . . .

Sidley, J. Smith, B. Jones, A 2019

https://doi.org/...

Figure 9: Unused infocard paper prototype. 2D representation moved to own column.



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 10: Symbol Nomenclature for Glycans [1] structure sheet used in 2D representation.