Usability of scientific software: focus on web-tools for 3-D modelling of carbohydrates

Literature Review

Lauren Paton
University of Cape Town
Cape Town, South Africa
ptnlau002@myuct.ac.za

ABSTRACT
The usability practices of scientific software are not well-defined due to the complex and specialised nature of this software. This issue is prevalent in web-tools for the visualization of carbohydrate molecules. This review aims to explore the general issues faced by developers of scientific software and summarises recommendations for best usability practices. Various usability evaluation methods are discussed and it can be shown that user testing is the most direct approach to addressing user needs, but in combination with expert inspections, the best results are achieved. Some available web-tools for carbohydrate molecule visualization summarized and their usability briefly assessed.

CCS CONCEPTS
• HCI • UCD

KEYWORDS
Scientific Software, Usability, Visualisation, Glycoscience, Glycan, Bioinformatics, Heuristic Evaluation,

1 Introduction
Industry-standard software systems need to be usable, focused on efficiency, learnability, memorability and utility, in order to be successful [19]. Additionally, many consider the available definitions of usability broad and not well-defined [2][18]. Usability can be summarised as a product's degree of efficiency, comprehensibility, effectiveness, and satisfaction with which users interact with the product [2, 7]. Software should be attractive to the user, learnable and adhere to the standards/guidelines when used under specific conditions [6].

The specialised nature of scientific software projects has posed a problem for developers and researchers, and made it difficult to incorporate usability into its design process. Time for analysis of scientific software is often not available [14] and scientific software is inherently not-intuitive, often poorly documented and difficult to install [2]. Scientific software is often released as a prototype developed by a small team of researchers and students [2] without the budget or skills to develop usable software. Domain experts have little training in software engineering [18], specifically in UX/UI design [13].

The usability of software is most often evaluated in the late stages of design [7]. A usability evaluation method (UEM) performs usability assessments of a system or interface against a set of pre-defined metrics at any stage in the design process [11]. UEMs are done through a series of well-defined activities and include collecting data on end-user interactions [15]. Studies on the use of different UEMs with scientific software have been done. In a mapping study of UEMs it was found that most methods were applied during the implementation phase of web-application development [6] and are therefore usually performed on functional/finished software. This literature review aims to investigate methods for evaluating the usability of functioning web-based scientific tools. We aim to investigate the best practices of testing software usability, factoring in the difficulties related to testing and evaluating scientific software.

Specifically, this review will focus on software developed in the fields of glycoscience and bioinformatics. Molecules "express" their function throughout their structure [2] and thus the 3-Dimensional qualities of a molecule characterize their biological function. So, drawing/visualisation are two of the most important activities done in structural glycobiology. Additionally, knowledge of carbohydrate conformation is useful for vaccine development and drug design and improving tools for 3-D carbohydrate visualization can provide a framework for data quality validation [20]. The availability and quality of tools for carbohydrate visualisation is therefore important. There are a number of web-tools for this purpose which will be summarised in this review. We will also give a brief assessment of their usability.

2. Usability of Scientific Web-Tools
Usability practices should be integrated throughout the entire design process of scientific software [17]. Although the simplest solution to ensuring software is usable is collaboration between software developers and domain experts [18], this is an expensive approach. There are certain precautions that can be taken to incorporate usability into the design of scientific tools [2]. One of these precautions is ensuring scalability. Scalability is necessary for early scientific projects as predicting the growth of a product is essential to its robustness and computational integrity. Another import technique is hiding unnecessary complexity from users. In order to minimalize user-confusion and ultimately errors, the user must only be exposed to mandatory parameters/information [2]. However, interfaces that are too "clean" can often abstract away too much detail [17]. Using standard data formats is especially important in scientific software, including glycan sketchers, which rely on certain notations to be comprehensible [20]. Nielsen's design heuristic "consistency and standards" emphasises the importance of standardised formats throughout a system [18]. Predicting user errors is also a precaution that needs to be taken – especially with scientific software, which is complex and often unintuitive [2]. This also links directly to one of Nielsen's design heuristics of "error prevention" [18]. Providing tutorials and quick-start guides creates a more beginner-friendly interface [13].

During a heuristic evaluation of bioinformatics tools [1] the following themes arose: The tools had poor UI design. The FAQ

(frequently asked questions) feature was useful. There was a lack of support for "powerful interactions" and researchers were forced to spend excessive time retrieving data. These qualities are important in scientific web-tools and should be factored into their design.

Additionally, there are mechanisms that have been shown to improve usability of web-tools, that should also be included in the design process. An experiment performed user-testing on a web-application and gauged the effect of introducing three usability mechanisms on the application's usability [7]. Specifically, on user satisfaction, effectiveness (percentage task completion) and efficiency (speed and number of clicks). The three mechanisms were abort operation, progress feedback and preferences. The results show that abort operation and preferences had a high impact on user satisfaction and effectiveness. Users were 74.7% more likely to be satisfied with the system if the preferences mechanism is adopted and 67.4% if the abort operation mechanism is adopted. In some cases, not including these two mechanisms prevented users from completing tasks entirely. Users were also 80.9% more likely to be efficient when the abort operation mechanism was adopted. There was no evidence implying that progress feedback had any impact on these factors. Additionally, it was also the most expensive to implement.

There were constraints resulting from the experiment's design: participants were not usability nor computer science experts. However, the fact that users were not familiar with the mechanisms could also be seen as contributing to the validity of the results [7]. Adopting tested mechanisms such as those above, is another good way to ensure system usability.

After a design iteration, the usability of a system can be evaluated using a number of UEMs.

3. Usability Evaluation Methods

Web usability evaluations are becoming a growing topic of interest and so, evaluation methods are evolving to accommodate this [7]. UEMs can be divided into two categories - usability tests and usability inspections - the former requires user-involvement while the latter does not [10]. Usability inspections include heuristic evaluations and cognitive walkthroughs. Usability tests include empirical methods of testing [12].

3.1. Heuristic Evaluation

Heuristic evaluations are inexpensive and fast [10]. The major advantage of heuristic evaluation is that they do not require users. However, they may not be suitable for more complex interfaces and may not accurately gather users' needs.

In a study done comparing heuristic evaluations and user-testing, it was found that heuristic evaluations identify more usability issues than any other UEM and have the greatest efficiency (issues found per time period) [12].

In a study analysing the use of heuristic evaluation on scientific data analysis and visualization tools [21], Nielsen's usability heuristics [18] were adapted to suit a system of this nature. Two heuristics were added: "Analytical Reasoning and Vis Support" and "Customized experience". In the same study, three scoring systems were used. One devised by Nielsen Group to map the usability problem to a usability rule. Another quantified the usability of the tool based on the heuristics. The third is a severity scale ranging from 'irritant' to 'unusable'. The results of the study indicated the usefulness of this multi-step framework and the adapted heuristics. The evaluators gave the application a low-rating for customization as the tool does not allow users to easily customize the interface. The study found that individual evaluators can find 35% of usability issues, and recommends having more than one for a heuristic evaluation [21].

Another study conducted on the design and evaluation of bioinformatics tools also used a specialized set of heuristics to evaluate usability [1]. The heuristics used were divided into three categories: First Impression, Query/Results Form and Interacting with Results. The participants of the study were usability experts, bioinformatics experts or both and they inspected the system based on the sets of heuristics and provided severity ranking, comments and suggestions. They results were categorized into strengths and weaknesses with the following themes emerging:

The metrics from the study were divided into two categories: Cognitive and Computational Perspective. Cognitive Perspective metrics represents overall satisfaction. Computational Perspective metrics represent efficiency. The two can be seen as qualitative and quantitative metrics respectively and can be represented in a usability matrix [7].

Heuristic evaluations may not always detect users' needs if they differ from the needs of the evaluator [10]. Cognitive walkthroughs are an inspective method that utilise user personas to analyse a system.

3.2. Cognitive Walkthrough

A cognitive walkthrough is a method in which evaluators assess an interface from the perspective of a potential user persona. They do not require access to users and can be done during any stage of development [10]. They require informed user models and thus extensive user research. The results obtained are limited to the skills of the evaluator and are often narrow aesthetic observations. A more direct approach to incorporate the usability needs of users is through user-testing.

3.3. User Testing

User-testing is an empirical test of software and the most commonly used UEM [15][7]. It is however, one of the expensive methods [10]. In study comparing heuristic and user testing, the issues identified by user-testing were of a greater severity and it took significantly less time [12]. User testing is sometimes more restricted than inspective methods, as users are limited to a list of tasks.

UEMs that involve users include benchmarking, usability questionnaires, surveys, interviews and think aloud protocols [10]. The strengths and weakness of these methods are tabulated below:

UEM	Description	Strengths	Weaknesses
Interviews	Inquiry into the opinions, attitudes and (perceived) habits/requirements of users	Good starting point (covers the requirements the user is aware of)	Time consuming Quality of results depend on many factors (attitude of interviewee, environment, quality of questions)
Benchmarking	Two interfaces compared (usually previous and new version)	Time efficient	Requires large group and repetitive sessions
Questionnaires + Surveys	Users answer questions after interacting with interface.	Can get quantitative data	Not diagnostic Data is speculatory, instead of behavioral
Think Aloud Protocols		More diagnostic Only requires a small number of people Demonstrates actual usage of software	Time consuming Rigid process Resource-intensive
Interaction data	Record data such a keystroke monitoring, eye movements, mouse movements and audio/video recorded	Results in the form of quantitative data	Expensive, requires equipment Requires finished product Requires large group of users Results often not "diagnostic"

Figure 1: Comparing UEMs involving users

Think aloud methods are useful as they only require a relatively small user group to gather sufficient data on conceptual problems and attitudinal phenomena. Some regard it an extremely valuable usability engineering method because it focuses on cognitive processes and demonstrates actual usage of software [12].

User centred design (USD) can be combined with user testing and has been shown to improve the usability of scientific software [18]. USD prioritizes the needs of end-users in the design on software. A study on USD in astronomical tools found that separating the design of the interface and the software allowed developers to focus better on user needs. It also found that introducing domain experts into the process of design allows us to clarify more complex user needs [18].

3.4. Choosing a UEM

Researchers experience difficulties when choosing evaluation methods because different attributes of software often require different evaluation methods in order to test their usability [10]. This also means there is no one method suitable for all types of web artefacts [6] and studies have suggested that combining methods yield the best results.

A study on usability of bioinformatics tools, devised a method of rating UEMs [10], rating a method as Low (L), Medium (M) or High (H) by its capacity to evaluate a usability factor. The results of the study's findings are summarised in the table (Figure 1) below:

		Usability Factor												
UEM Type	UEM Name	Understandable	Learnability	Well-structured	Effectively presented	Effectively supported	Tool-supported	Complete	Concise	Unambiguous	Operable	Navigable	Satisfaction	Tailorable
Testing	Thinking aloud protocol	H	M	M	M	M	M	L	M	M	H	H	H	L
	Remote Usability Testing	H	H	H	H	H	H	H	H	H	H	H	H	H
	Performance Measurement	M	M	L	M	M	L	L	L	L	M	H	L	L
Inspection	Heuristic Evaluation	L	L	M	M	M	M	M	L	L	L	M	L	L
	Cognitive walkthroughs	M	H	H	H	H	M	L	M	L	L	M	L	L
	Feature inspections	L	L	L	L	L	H	L	L	L	L	H	L	L
Inquiry	Field Observation	L	L	L	L	L	L	L	L	L	M	L	H	M
	Focus groups	M	M	M	M	M	M	M	M	M	M	M	H	M
	Logging Actual Use	L	L	L	L	L	L	L	L	L	M	L	L	L

Figure 2: Comparing UEMs by usability factors [10]

There are tools for choosing appropriate usability evaluations. Two such methods are the flow chart and framework designs [10]. The flow chart design is a simple design based on two decision stages. The first is whether a system requires usability test or a usability inspection. If a system/feature requires user-testing, it must then be considered if it must be tested against benchmark values. In this case, a cognitive walkthrough is recommended. If not, a customer talk-through is recommended. If no users are required, then it must be further considered if an adequate understanding of key tasks is necessary. If this is the case then a walkthrough is appropriate. If not, a heuristic evaluation is appropriate.

The framework design was devised by experts at the Nielsen Norman Group. It uses a chart with 3 axes (based on the nature of a study/experiment) to select a UEM. The three axes are: Attitudinal versus Behavioural, Qualitative versus Quantitative and Context of use. The chart (Figure 3) can be seen below:

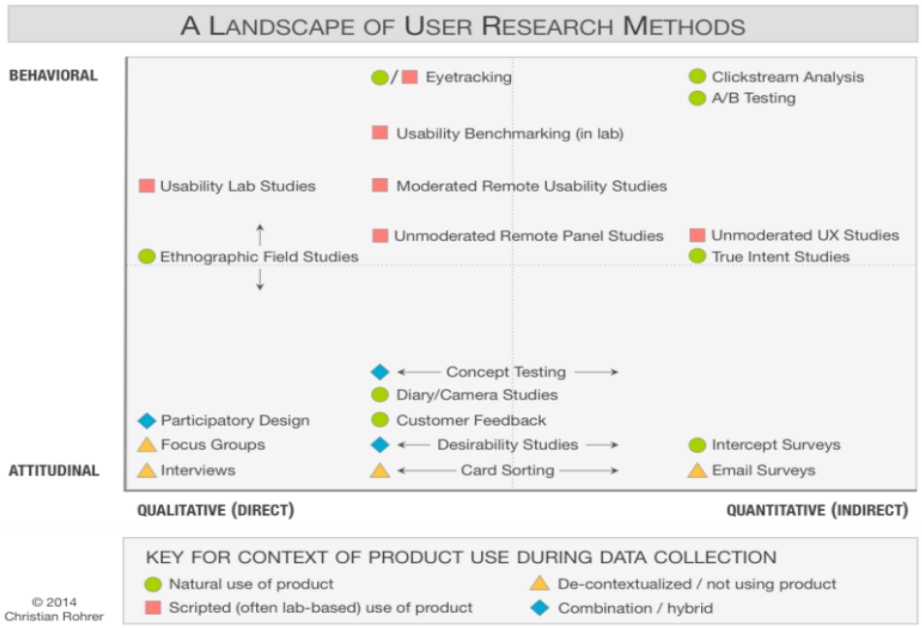

Figure 3: Framework design for choosing UEM [10]

The use of the framework design is recommended, when working on a small to medium software project and when one is unsatisfied with the current usability practice [10]. These UEMs can therefore be used to investigate the usability of available web-tools for 3-D modelling of carbohydrate molecules.

4. SOFTWARE FOR BUILDING CARBOHYDATE MODELS

Research in field of glycoscience relies heavily on tools for visualization of molecular structures [15]. There are many computational tools available - this review will focus on web-based tools for the sketching and modelling of glycans. Historically, representing the 3-D structure of carbohydrates has been a difficult task but developments in these tools have allowed users to easily draw glycans. The main characteristic of a helpful visualisation tool is adaptability (in terms of colours, size, orientation, etc.) and usability. Due to the complex nature of carbohydrates, another important requirement is accurate depiction. Encoding glycan structures in a file is necessary for collaboration between scientists and as a result – different representations have been proposed [15]. Rendering 3-D glycan structures requires recording using a notation that includes atomic coordinates and so using the PDB (protein data bank) format is beneficial [20].

Glycan sketchers allow users to draw 2-D glycans manually or by importing/exporting structure files. Some allow users to create 2D glycan structures by "dragging-and-dropping" monosaccharides onto a canvas. Glycan builders build 3-D models from 2-D sketches or directly from input.

Below various glycan sketchers and builders are summarised [15] and comparisons of their usability are tabulated.

4.1. DrawGlycan-SNFG

DrawGlycan-SNFG is a tool presented by the Virtual Glycome and developed at the University of Buffalo, New York [23]. It is both a web interface and open-source software for 2-D sketching of glycans. The interface takes in a glycan string as input and the resulting sketch can be downloaded as a .jpg. This file can also be modified in terms of text, line and symbol and their size and spacing. DrawGlycan-SNFG is not appropriate for obtaining 3-D models as it lacks sufficient data conversion functionality [20].

4.2. Glyco.ME (Sugar Builder)

Glyco.ME is an online interface for rapid glycan sketching. It is still in development by the *in silico* glycomics group in Copenhagen, part of the Center for Glycomics [9]. It uses a template system in which monosaccharides can be added to a

structure template selected by the user – it uses a drag-and-drop design. Sketching is limited to a set of "rules" as the user interacts – these can be deactivated. The final image can be in .png or.svg format.

4.3. GlyCam-Web (Carbohydrate Builder)

GLYCAM-Web is a tool that generates sketches and subsequently 3-D models of glycans. It was developed in the Complex Carbohydrate Research Centre at the University of Georgia [9]. There are three options for glycan building: manual building using the "Carbohydrate Builder" button, using a template library or direct text input (used often when a glycan is especially complex and/or not in the template library). GlyCam-Web allows minimisation [15]. The 3-D outputs can be compressed into a file containing PDB (.pdb) files and 2-D sketches in GIF format.

4.6 PolysGlycanBuilder

PolysGlycanBuilder is an interactive web-tool developed at the Centre National de la Recherche Scientifique (French National Centre for Scientific Research) [16]. It translates a glycan sequence or polysaccharide repeat unit into the coordinates of the final structure which are then outputted in PDB format. PolysGlycanBuilder can take in files in INP, IUPAC and GlycoCT formats. It also has a "drag-and-drop" option for its 2-D modelling – the image can also be downloaded in SVG format. Glycosidic linkages and dihedral angles can be edited easily by the use. These sketches can be further converted into a PDB file containing the 3-D coordinates required [15].

Figure 4 (below) is an overview of the findings of a brief usability assessment of each of the above tools:

Tool	Description	Strengths	Weaknesses
DrawGlycan-SNFG	Glycan Sketcher	Good instructions Hides advanced options Simple interface Demonstration videos available Instant output with option to save as image immediately Builds around errors	Only textual input
Glyco.me (SugarBuilder)	Glycan Sketcher	Templates allow for quick start Simple interface	Unclear instructions No demonstrations or examples Cannot build from blank
GlyCam-Web	Glycan Builder	Clear system status Icons are well labelled Format is more structured than others	Not drag and drop No feedback or call to action
CHARMM-GUI	Glycan Builder	Thorough demos	Only takes in PDB format file (no drag and drop) Not beginner- friendly Build takes very long
Sweet II	Glycan Builder	Beginner and expert options Tutorials/examples, multiple download options	Text-based, older interface No back option from rendered page Outdated interface
PolysGlycan Builder	Glycan Builder	Good call to action Drag and Drop Buttons have explanations, interface is well documented Feedback for impossile construction	Dragging animation poor Slow to indicate system status

Figure 4: Results of a usability assessment of web-tools

4.4. CHARMM-GUI (Glycan Reader and Modeler)

CHARMM-GUI's Polymer Builder is a tool that generates 3-D models. It was developed by a team at Lehigh University, Bethlehem, Pennsylvania [3]. It reads in files in certain formats (PDB, PDBx/mmCIF and CHARMM) and then detect carbohydrate molecules and linkage information or can read a glycan sequence (GRS format) [15]. The image of the carbohydrate is rendered as monosaccharides are added.

4.5 Sweet II

Sweet II is a tool used for generating 3-D models from a glycan sequence and part of the GLYCOSCIENCES.de project – developed at the German Cancer Research Centre [22].
Input can come from a library of relevant oligosaccharides, in standard glycan format or via manual entry, and the 3-D model output is in PDB file format. It is considered a versatile tool for glycan modelling [15].

5. Discussion

The UEMs discussed above can be applied to web-tools for 3D glycan visualization. The framework method can be used to choose a suitable UEM [10] as these tools fall into the category of small to medium projects. As suggested previously, a combination of evaluations by experts and user tests (conducted with individuals with some level of domain or usability knowledge) may be the most suitable approach to evaluating and improving functioning projects [18][12]. Heuristic evaluations for specialized software - such as glycan visualization tools, can be helpful, especially if heuristics are adapted to suit the requirements of the specific software and intended users [1][21]. Certain mechanisms that have been shown to improve usability, such as an abort operation and preferences options [7], should be included in the usability requirements of projects. Other mechanisms specific to the project could also be adopted and

tested, bearing in mind that the effectiveness of a mechanism is not proportional to the expense of adopting it [7]. Various methods can be used but a successful method for refactoring the usability of existing tools would be benchmarking – in which previous and new versions of an interface are compared [10].

The difficulties faced by developers of scientific software are prevalent in web-based glycan building tools. Many tools are not beginner-friendly and rely on sufficient domain knowledge to operate. From the table above, there is a correlation between interfaces that are graphics-based and usability – building tools with a drag-and-drop style were easier to understand. Some tools provide instantaneous rendering of images which allow the user instant feedback on any errors made, including DrawGlycan-SNFG [5]. Tools with a clear call to action, instructions and demonstrations were much easier to use. The three figures to follow display the interfaces of GlyCam-Web, Polys Glycan Builder and DrawGlycan-SNFG. All of which have qualities which, when combined, could produce a wholistically more usable interface.

Below, Figure 5 shows part of the GlyCam Carbohydrate Builder interface [8]. The arrangement of the monosaccharides is very structured. Other tools lack this structure and/or do not emphasize the available monosaccharides as much as the GlyCam interface. This structure could aid the user in the design of carbohydrates, especially if monosaccharides are arranged by some chemical properties.

Figure 6 below shows part of the Polys Glycan Builder Interface [16]. Polys uses a drag-and-drop design which is helpful for beginner users. Other tools, which are text-based, are less intuitive. However, the animation of the dragging is disjointed and confusing.

Figure 5: Part of the GlyCam-Web Carbohydrate Builder Interface [8]

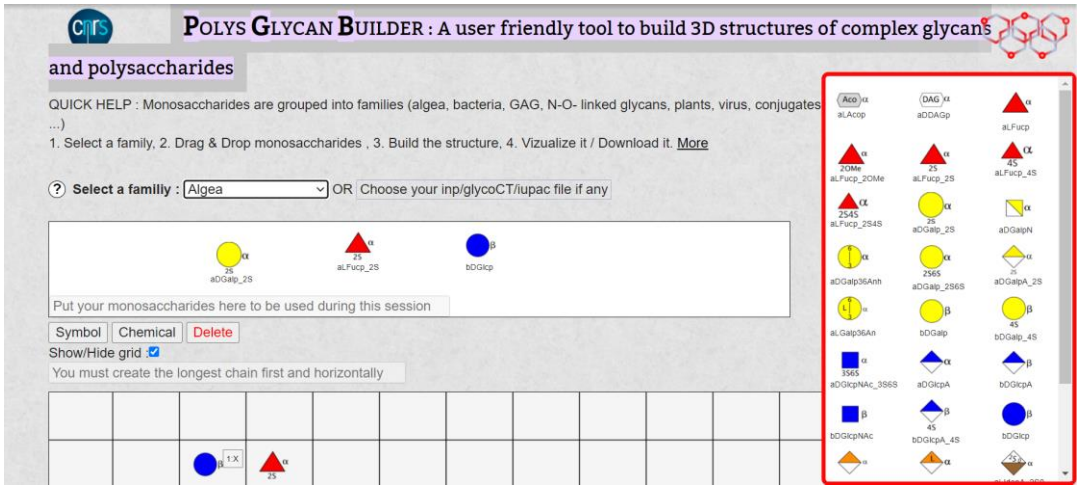

Figure 6: Part of the Polys Glycan Builder Interface [16]

Finally, Figure 7 below displays part of the DrawGlycan-SNFG interface [5]. This interface hides advanced options from users, making it appear simple and clean. It also automatically renders the 2-D image of the glycan which can be immediately downloaded as a .png file. Unfortunately, it is text-based, but it also provides examples for users to view.

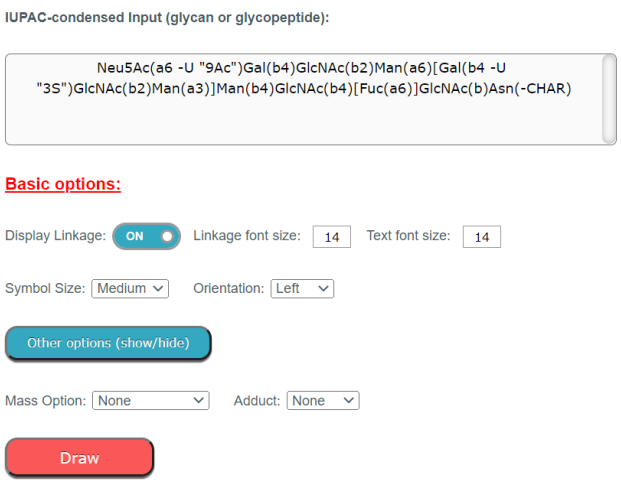

Figure 7: Part of the DrawGlycan-SNFG Interface [5]

6. Conclusions

Developers have struggled and neglected to incorporate usability into the design process of scientific software. The specialized nature of scientific software means careful precautions need to be taken to ensure usability and promote collaboration amongst researchers. The issues faced by developers of scientific software are prevalent in web-tools for visualization of carbohydrate molecules. It is possible to improve the usability of these functioning software projects and the UEMs outlined above can be used to do so. Although user testing is the most direct approach to addressing user needs, a combination of expert evaluations/inspections and user testing is recommended – especially when working with complex interfaces used by domain experts. Improving the usability practices of these web-tools could help researchers understand the structure and function of carbohydrate molecules, ergo contributing to drug design and vaccine developments.

REFERENCES

[1] Naelah Al-Ageel, Areej Al-Wabil, Noura AlOmar and Ghada Badr. (2015). Human factors in the design and evaluation of bioinformatics tools. *Procedia Manufacturing, Volume 3, 2003-2010*. https://doi.org/10.1016/j.promfg.2015.07.247.

[2] 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.

[3] CHARMM-GUI Glycan Reader & Modeler. Retrieved from http://www.charmm-gui.org/?doc=input/glycan (accessed May 2022).

[4] Kai Cheng, Yusen Zhou, Sriram Neelamegham, DrawGlycan-SNFG: a robust tool to render glycans and glycopeptides with fragmentation information, *Glycobiology, Volume 27, Issue 3, March 2017, Pages 200–205*, https://doi.org/10.1093/glycob/cww115

[5] DrawGlycan-SNFG. 2020. *Render glycans and glycopeptides with fragmentation info. using the Symbolic Nomenclature for Glycans* [SNFG]. Retrieved from http://www.virtualglycome.org/DrawGlycan/. Accessed May 2022.

[6] Adrian Fernandez, Emilio Insfran and Silvia Abrahrão. 2011. Usability evaluation methods for the web: a systematic mapping study. *Information and Software Technology. Volume 53(8), 789-817*. https://doi.org/10.1016/j.infsof.2011.02.007.

[7] Juan M. Ferreira, Silvia T. Acuña, Oscar Dieste et al. 2019. Impact of usability mechanisms: an experiment on efficient, effectiveness and user satisfaction. *Information and Software Technology, Volume 117 (2020).* https://doi.org/10.1016/j.infsof.2019.106195.

[8] GLYCAM Web. 2020 Complex Carbohydrate Research Center, University of Georgia, Athens, GA. Retrieved from http://glycam.org. Accessed May 2022

[9] Glyco.me SugarBuilder. Retrieved from https://beta.glyco.me/sugarbuilder. Accessed May 2022.

[10] Hanan Hayat. 2020. Evaluating Software Usability from Different Perspective: A Framework for Encouraging Usability Evaluations by Focusing on Software Developing Projects. Master's Thesis. Loughborough University, Loughborough, England.

[11] Hanan Hayat, Russell Lock, Ian Murray. 2019. Measuring software usability. *BCS Software Quality Management Conference, Loughborough University, 30 March 2015.*

[12] Martin Maguire and Paul Isherwood. 2018. A comparison of user testing and heuristic evaluation methods for identifying website usability problems. Retrieved May 2022 from https://hdl.handle.net/2134/36466.

[13] Serghei Mangul, Lana S. Martin, Eleazar Eskin and Ran Blekhman. 2019. Improving the usability and archival stability of bioinformatics software. *Genome Biology, Volume 20, Article 47 (2019), 3 pages*. https://doi.org/10.1186/s13059-019-1649-8.

[14] Reed Milewicz, Paige Rodeghero et al. 2019. Position Paper: Towards Usability as a First-Class Quality of HPC Scientific Software. *In Proceedings of the IEEE/ACM 14th International Workshop on Software Engineering for Science. May 28, 2019, Montreal, QC, Canada. 41-42*. doi: 10.1109/SE4Science.2019.00012.

[15] Kanhaya Lal, Rafael Bermeo and Serge Perez. (2020). Computational tools for drawing, building and displaying carbohydrates: a visual guide. *Beilstein J Org Chem, Volume 16, Pages 2448-2468.*

[16] PolysGlycanBuilder. *A user friendly tool for to build 3D structures of complex glycan and polysaccharides.* http://glycan-builder.cermav.cnrs.fr/. Accessed May 2022.

[17] Lavanya Ramakrishnan and Daniel Gunter. Ten principles for Creating Usable Scientific Software. (2017*). IEEE 13TH International Conference on eScience. October 24-27, 2017, Auckland, New Zealand, 210-218*. doi: 10.1109/eScience.2017.34.

[18] Laurisha Rampersaad, Sarah Blyth, Ed Elson and Michelle Kuttel, M. 2017. Improving the Usability of Scientific Software with Participatory Design: a new Interface Design for Radio Astronomy Visualisation Software. *In Proceedings of SAICSIT'17, September 26-28, Thaba Nchu, South Africa, 1-9.* https://doi.org/10.1145/3129416.3129899

[19] Helen Sharp, Jenny Preece and Yvonne Rogers. 2019. Chapter 1: What is interaction design? Interaction Design: Beyond Human-Computer Interaction (5th edition). Wiley, New York, NY.

[20] Sofya I. Scherbinina and Philip V. Toukach. 2020. Three-Dimensional Structures of Carbohydrates and Where to Find Them. *Int J Mol Sci, Volume 21(20), 1-46*. Doi: https://dx.doi.org/10.3390%2Fijms21207702.

[21] Samar Swaid, Mnsa Maat, Hary Krishnan et al. 2018. Usability Heuristic Evaluation of Scientific Data Analysis and Visualization Tools. *International Conference on Applied Human Factors and Ergonomics, Volume 60. 472-478.* http://dx.doi.org/10.1007/978-3-319-60492-345.

[22] Sweet. 2021. Glycosciences.de. Retrieved from http://www.glycosciences.de/modeling/sweet2/doc/index.php. Accessed May 2022.

[23] Virtual Glycome. 2022. Retrieved from https://virtualglycome.org/. Accessed May 2022.