Using user-centered design to unburden genetic analyses for novice genomic researchers

Harsh Vijaykumar Patel

A dissertation

submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:
David R. Crosslin, Chair
Gail P. Jarvik
Taryn O. Hall

Program Authorized to Offer Degree:
Department of Biomedical Informatics and Medical Education

University of Washington

Abstract

Using user-centered design to unburden genetic analyses for novice genomic researchers

Harsh Vijaykumar Patel

Chair of the Supervisory Committee:
David R. Crosslin

Department of Biomedical Informatics and Medical Education

Increasingly larger genomic databases have allowed for more robust genetic analyses, leading to advances in bioinformatics, translational medicine, and, ultimately, improving patient care. However, the current landscape of genetic analysis software is riddled with unintuitive and inaccessible tools and software packages. These tools often lack proper documentation, need extensive setup, fail to communicate with each other, and require painstaking debugging for even simple exploratory analyses. This creates large barriers of entry for novice genomic researchers (NGRs), individuals who are interested in conducting genetic experiments but either lack the computational experience/biological background or do not have access to extensive technological resources, such as local computational clusters. Historically, very little work has been done to address the needs of NGRs, leading to an overlooked, but keystone user base that lacks proper foundational support needed to best begin their informatics journey. User-centered design (UCD) is one solution to this problem that has been under-utilized in bioinformatics software development. In this work, we sought to better characterize the NGR user base and to apply the UCD framework during the development of a more usable bioinformatics software tool. To achieve this, we first explored the existing landscape of bioinformatics software tools via a literature review and sought to create a rubric that can be utilized to evaluate the usability of those tools within the context of NGRs. To further inform the creation of this rubric, we also performed a needs assessment of NGRs utilizing semi-structured interviews. From these two sources of knowledge, we found that the key attributes that resulted in poor adoption and sustained use of most bioinformatics tools included poor documentation, lack of context-specific instructional content, difficulty in installation and setup, and uninformative error messages (**Aim 1**). We then created user personas to help better characterize specific types of users and utilized those personas to help design a cloud-agnostic, user-friendly GWAS analysis tool (UF-GWAS). UF-GWAS utilized a Docker container to neatly package a JupyterLab instance which allowed users to run GWAS analyses quickly and easily (**Aim 2**). Next, we evaluated the usability of UF-GWAS by recruiting NGRs who performed task-based evaluations. We also tested the efficiency, accuracy, and cost of UF-GWAS against industry standard software. NGRs reported UF-GWAS as highly-usable and appreciated the following key components: clarity of the documentation, quick access to relevant background knowledge, ease of onboarding, and the shareability and reproducibility of results (**Aim 3**). Finally, we combined the many knowledge sources throughout this study to create a set of guidelines that future researchers can follow in order to create more usable informatics software. As NGRs and other researchers begin to enter

the informatics landscape, it will become increasingly important to as informaticians to create more usable analysis software. By doing so, we can encourage robust experiments from a more diverse workforce, hopefully leading to an improvement in quality of care.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

"It is important to draw wisdom from many different places. If you take it from only one place, it becomes rigid and stale." - Uncle Iroh

When I was sitting down to write this list of acknowledgements, I soon realized it would be an impossible task to thank everyone who has influenced, guided, encouraged, and inspired me throughout my graduate school journey. But as with all impossible tasks, there is always something one can do to make a dent. I want to start by thanking my dissertation committee (Dr. David Crosslin, Dr. Gail Jarvik, Dr. David Veenstra, and Dr. Taryn Hall) for their guidance and wonderful feedback over the past few years. I want to also thank David Crosslin for being a wonderful mentor and someone I know I could turn to when I needed advice. This was not an easy journey but David made it manageable with his encouragement and motivation.

Next, I want to thank all of the current and previous members of the Electronic Medical Records and Genomics (eMERGE) Network and the Clinical Sequence Evidence Generating Research (CSER) Consortium. Without the foundation you all built, I would not have been able to explore my passions and utilize the fruits of the data collection and coordination that was already done before I even joined UW. I also want to thank all of the participants who agreed to help me with my user studies, some of which are my fellow peers here at UW BIME. It was a great honor to work alongside such passionate scientists and I look forward to seeing your wonderful research in the future.

Thank you to all of my friends, peers, professors, and all of the administrative staff here at UW BIME. I was just this brazen undergraduate from Georgia and you all helped foster such a wonderful and caring work environment. For that, I will always be grateful. I want to also thank my friends Grace, Jackson, Daniel, Angelica, and the boys at the Cabin (Bryce, Jason, Tommy, and Trav) for bringing me much needed laughter and for their words of encouragement over the years. Finally, I want to thank my mom Bharti, dad Vijay, brother Dhruv, my grandparents, my extended family (all of my cousins, aunts, and uncles) for being extremely supportive and patient throughout all these years. Words cannot capture how appreciative I am and I love everyone of y’all!

DEDICATION

For my parents, Bhartiben and Vijaykumar Patel.
I’m sure it was not easy being my parents, but I hope I made you proud!

CHAPTER 1: INTRODUCTION

1.1 Background

Increasingly larger genomic databases have allowed for more robust genetic analyses, thus leading to advances in bioinformatics, translational medicine, and, ultimately, improving patient care. Over the past two decades, the amount of genetic data collected, stored, distributed, and analyzed has grown exponentially [1, 2]. It is estimated that the total amount of human genomes being sequenced doubles every seven to twelve months and, by 2025, will require storage needs that rivals astronomical data and video data stored on YouTube. Additionally, while genomic sequencing and variant discovery was historically conducted almost exclusively by large, government-sponsored research projects, such as the Human Genome Project, widespread use of next-generation sequencing (NGS) technologies have allowed smaller laboratories, researchers, and even private, consumer-facing companies such as 23andMe to generate, store, and analyze high quality genetic information [3, 4]. Although this has led to the growth of precision medicine and individualized treatment plans, many stakeholders accurately point out storage and access challenges currently being faced along with the plethora of privacy and security hurdles both researchers and private citizens will have to overcome in the future [5].

The quantity of bioinformatics tools has significantly increased to keep pace with the growth of genomics data collection and analysis. In 2016, the OMICtools database reported over 25,000 available genetic tools, with this number increasing rapidly each year [6]. Although most of these tools are developed independently in laboratories, there has been a rise in the usage and popularity of large-scale analysis platforms, such as Analysis Visualization and Informatics Lab-space (AnVIL)/Terra and the All of Us platforms [7, 8]. However, despite the availability of these tools, there are widespread adoption challenges due to usability barriers resulting from the lack of user-centered design during development and initial evaluation.

User-centered design (UCD) is relatively scarce in the bioinformatics tool landscape and is critical in developing functional, user-engaging platforms. Very often, genetic tools that do not utilize user-centered design principles suffer from “the joys of an early release that lasts a short time and the bitterness of an unusable system that lasts for years” [9]. While UCD usage in the bioinformatics space has increased slightly over the past decade, such as the development of genetic reports readable by non-specialists and creation of information portals such as the Enzyme Portal, there is still a noticeable gap of usability considerations in genetic tools tailored for knowledge discovery and risk analysis [10, 11]. Early examples of applying UCD techniques in designing genomic analysis tools include GenomIUm, where expert bioinformaticians reported reduction of complexity during knowledge extraction and easier analysis workflows, and the All Of Us Cohort Selection Pipeline, which allows for easy case-control cohort selection [8, 12]. Overall, there is a need for more usability guidelines during the development and evaluation phases of tool creation and by doing so, we can enable researchers to collect, analyze, and share information more effectively and efficiently [13].

An often overlooked keystone user group of bioinformatics tools are Novice Genomics Researchers (NGR). NGRs are individuals who are interested in conducting genetic experiments but lack the quantitative background or experience with existing bioinformatics techniques/tools or do not have access to extensive technological resources, such as local computational clusters. Very little work has been done to address the needs and challenges faced by these users as they begin to enter the genomic analysis space and their interaction with existing bioinformatics tools. A core mission of the National Human Genomic Research Institute (NHGRI) is to train a diverse and talented workforce for the future of genomics in healthcare [14]. However, as discussed earlier, the current landscape of training and analysis tools fail to acknowledge barriers of knowledge gaps and usability, leading to user frustration and experimental delays within novice genomics researchers.

1.2 Specific Aims

To address the rising clutter of bioinformatics tools lacking UCD principles and to address the complex needs of NGRs, in this dissertation, we aimed to develop a more user-friendly bioinformatics analysis tool targeted specifically towards NGRs. The following specific aims provide further detail about that process:

1.2.1 Aim 1: Rubric Development via Literature Review and Needs Assessment

In this aim, we created an evaluation rubric for bioinformatics software based on two key sources of knowledge. The first source was from the existing literature on bioinformatics tools both widely known and commonly used in practice. The second source came directly from NGRs via semi-structured interviews that extracted key barriers faced when first interacting with bioinformatics software. By combining these two knowledge sources, we were able to better characterize the current state of bioinformatics software and the challenges future software will have to overcome to address the complex needs of NGRs.

1.2.2 Aim 2: User-centered, cloud-agnostic GWAS Application

In this aim, we developed a bioinformatics analysis software called UF-GWAS that utilized UCD principles to create a more user-friendly workflow when conducting GWAS experiments. Insights from Aim 1 were utilized to create user personas that helped inform the design of UF-GWAS. To test UF-GWAS, a GWAS experiment on the *Clostridioides difficile* phenotype from the eMERGE dataset was run on a cloud-hosted analysis environment.

1.2.3 Aim 3: Evaluation of UF-GWAS and Recommendations for UCD Bioinformatics Tool Development

In this aim, we evaluated UF-GWAS by performing task-based user evaluations and provided a set of guidelines for UCD-based bioinformatics tool development. NGRs were asked to perform sample GWAS experiments and provide feedback on the usability of the system. This feedback paired with knowledge gained from Aim 1 and Aim 2 was used to create development guidelines that can be adapted for future use in a wide variety of bioinformatics systems.

1.3 UCD Theoretical Framework

To provide structure to the overall dissertations, we utilized the user-centered design (UCD) conceptual framework, proposed by Donald Norman and Stephen Draper [15] . UCD is a long-standing and proven framework for the development of both hardware and software services and systems [16]. UCD rests on the idea that a service or system will best fulfill its user needs when designers utilize an iterative design process in which users are involved in each step of the design process, leading to highly usable and accessible products.

Figure 1.1 below details the overall UCD framework and how each aim of this dissertation falls in line with the different stages of the conceptual framework.

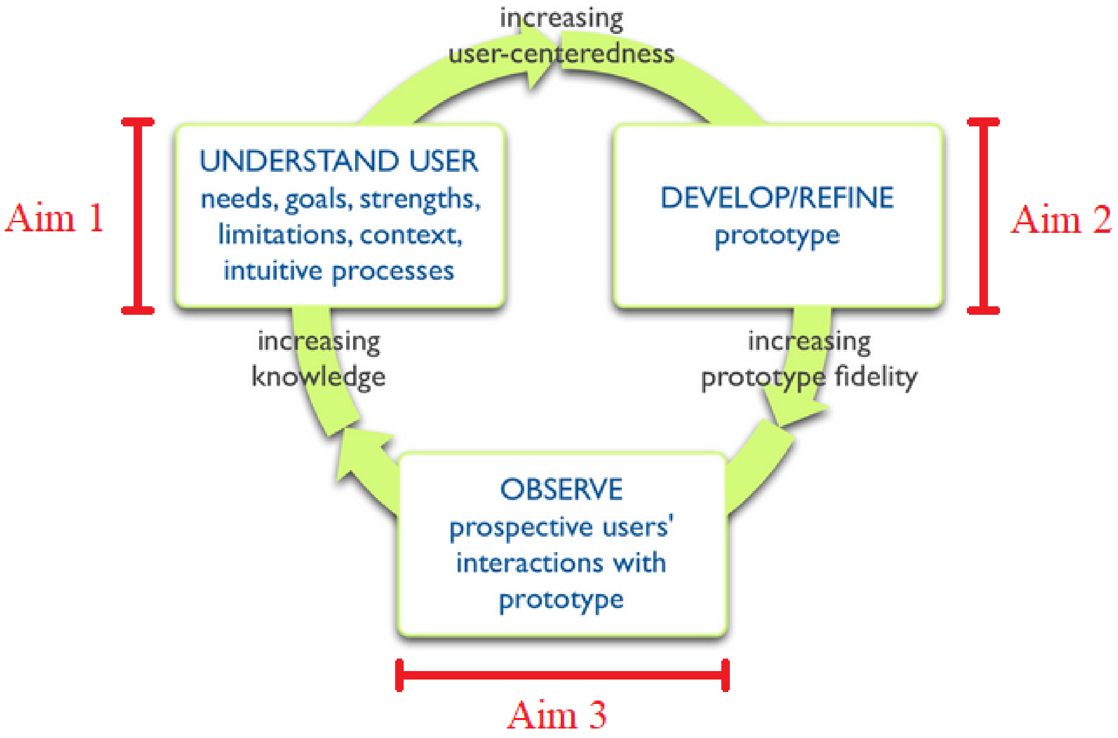


Figure 1.1. Schematic of User-Centered Design Framework. Source: Witteman et al. 2015 [16]

While the UCD framework has seen some usage in the healthcare space, it has not been thoroughly utilized in the development of user-friendly tools and technologies [17]. Specifically within the context of bioinformatics tools, there exists a great need to consider the needs of the users during the design and development process of future technologies. We therefore utilized the UCD framework to help guide the design and development of a user-friendly bioinformatics tool, tailored for NGRs. **Aim 1** centered around understanding the current user needs in order to “increase user-centeredness”, the upper-left component of the visual UCD framework from **Figure 1.1**. We then utilized this knowledge of the users’ needs and challenges to develop and refine a tool prototype in **Aim 2**. Lastly, we evaluated the novel tool developed and re-engaged with the users in **Aim 3** to understand how the users interact with the bioinformatics tool and gather recommendations/guidelines for further refinement. Although the annotations of the specific aims in **Figure 1.1** are isolated into three distinct nodes, multiple aspects of the UCD framework (understanding the user, developing/refining prototypes, and observing user interactions) were utilized throughout all aims of the study in order to best engage the target users and create a highly accessible end product.

CHAPTER 2: RUBRIC DEVELOPMENT VIA LITERATURE REVIEW AND NEEDS ASSESSMENT (AIM 1)

Abstract

The lack of user-centered design principles in the current landscape of commonly-used bioinformatics software tools poses challenges for novice genomics researchers (NGRs) entering the genomics ecosystem. Furthermore, comparing the usability of one analysis software to that of another is a non-trivial task and requires evaluation criteria that incorporates perspectives from both existing literature and a diverse, underrepresented user base of NGRs. In order to better characterize these barriers and to provide a metric for evaluation, in this aim, we utilized a two-pronged approach consisting of a literature review of existing bioinformatics tools and semi-structured interviews of the needs of NGRs. From both sources of knowledge, the key attributes that resulted in poor adoption and sustained use of most bioinformatics tools included poor documentation, lack of readily-accessible informational content, challenges with installation and dependency coordination, and inconsistent error messages/progress indicators. An evaluation rubric that considered the above components along with several others was created to help grade both existing and future bioinformatics software tools. As the NHGRI's 2020 strategic vision aims to foster a more diverse workforce, create solid foundations for exciting genomics research, and develop robust analysis tools, it is becoming increasingly important to consider multiple perspectives in the development and deployment of analysis software tools. By doing so, both novice and experienced researchers will benefit from more user-friendly tools leading to streamlined training and richer analyses for future bioinformaticians.

2.1 Introduction

Bioinformatics is a rapidly growing field that involves the application of statistical methods and computational algorithms to analyze and interpret biological data. With the increasing availability of genomics data, the development of bioinformatics tools has become essential for researchers to efficiently analyze and interpret the vast amounts of biological data generated. In order to rival this rapid growth of genomic data, the amount of bioinformatics tools being developed and published has increased dramatically [18]. While most of those tools are developed in siloed laboratories, large scale analysis platforms have also risen in usage and popularity, such as National Human Genome Research Institute's (NHGRI) Analysis Visualization and Informatics Lab-space (AnVIL)/Terra and the All of Us platforms [19, 20]. Most of these tools, however, suffer widespread adoption challenges due to inherent usability barriers from a lack of user-centered design during developed and testing phases.

Usability is a fundamental aspect that influences both the initial reception and longevity of interactive application. The International Organization for Standardization defines usability as "the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use" [21]. While related fields of study such as global health and healthcare education for medical professionals have enlisted the aid of professional design and innovation firms such as IDEO, Frog, and Dalberg to tackle challenges that arise with proper usability [22, 23], usability analysis in bioinformatics tools development is scarce and poses many barriers to proper development and sustained adoption [13, 24]. Tools and software packages improperly created without usability considerations in mind suffer from lack of adoption or sustained use [9]. Some of these considerations include easy installation and setup, a stable uniform resource locator, proper documentation, lack of onboarding tutorials and test materials [25]. Furthermore, the challenges and distribution of software tools in academia, such as assuming technical and biological background of end users, academic journals and conferences being the primary source of distribution of tools, and the push for publication resulting in dissemination of incomplete tools, pose great barriers for long-term tool stability.

To address some of the challenges listed above, several bioinformatics tools have been developed specifically with usability and end-user needs in mind. Examples of applying user-centered design (UCD) techniques to boost usability of bioinformatics analysis tools include GenomIUm, where expert bioinformaticians reported reduction of complexity during knowledge extraction and easier analysis workflows, and the All of Us Cohort Selection Pipeline, which allows for streamlined case-control cohort selection [12, 20]. Larger research institutes such as the National Human Genome Research Institute (NHGRI) and EMBL-European Bioinformatics Institute (EMBL-EBI) have also begun to push for more deliberate user-friendly tool development and dissemination citing “the user is the future” [9, 26].

Although there has recently been a push towards developing more user-centered bioinformatics tools, novice genomics researchers (NGR) are overlooked despite being keystone users of these tools and the ones who can benefit the most from platforms developed using UCD principles. NGRs are individuals who are interested in conducting genetic experiments but lack the quantitative background or experience with existing bioinformatics techniques/tools or do not have access to extensive technological resources, such as local high performance computational clusters. As a core mission of the NHGRI is to train a diverse and talented workforce for the future of genomics in healthcare, NGRs pose as a key user base for the future of bioinformatics tools adoption and sustained usage [26, 27]. Existing challenges faced by NGRs can be elucidated by examining the current perspectives on bioinformatics training. Attwood et al. reviewed global perspectives on evolving bioinformatics and data science training and found a plethora of challenges faced by novice researchers [28]. Some of these challenges included training deficits in basic programming, biostatistics, data mining, data analysis and visualization, bioinformatics tools usage, and scaling to the cloud or a high performance cluster. All of the aforementioned hurdles contributed to the NGR's overall "acquisition of expertise and confidence in data/statistical analysis and interpretation." Machado et al.'s case study on teaching and learning in bioinformatics reinforces the importance of information gathering needs of novice researchers and the hurdles currently faced in the forms of poorly designed bioinformatics tools and databases, lack of proper documentation and support, and vague or missing error handling [29]. UCD principles can be utilized to address these needs of NGRs and produce a workforce capable of interacting with, manipulating, analyzing, and sharing biological data more effectively and efficiently [2].

Albeit limited, existing work to address the needs of NGRs has already been done in several disciplines of bioinformatics. Murphy et al. studied users who regularly navigated complex databases and designed a visual interface to ease the burden for novice users trying to find research patient cohorts [30]. Britto et al. conducted scenario based usability analysis of a pediatric patient portal to study the challenges faced by novice users and reported major barriers of medical language complexity and difficulties in navigation [31]. We see barriers of information overload and lack of proper tool documentation faced by non-specialists who try to understand complex genetic reports and even by novice ontologists utilizing popular ontology creation tools to capture a relatively simple biological system (as evidenced by Zhang et al.'s study documenting user challenges creating an ontological representation of parts of a leaf) [11, 32]. The challenges faced by novice users in the above studies are not limited to just the contexts of those investigations but are universal across bioinformatics as novice researchers continue to be overlooked.

To fill the gap of usability in bioinformatics tools and to query the needs of NGRs, in this aim, we will employ a mixed-methods study design consisting of two primary parts: a literature review and a needs assessment of target users. First, a retrospective analysis of existing bioinformatics analysis tools utilized by researchers to perform genetics experiments will be conducted to better understand the current landscape of available analysis instruments. Thereafter, a prospective analysis using semi-structured interviews will be conducted to assess the needs of novice genomics researchers from different sites. The primary goal of this aim will be to identify a core rubric for the evaluation of genetics tools for usability and performance, within the context of novice genomics researchers. We hypothesize the existing literature and the initial needs of NGRs will primarily revolve around thorough documentation and tutorials, minimalistic user-interfaces, and low barriers of installation and deployment of bioinformatics tools.

2.2 Methods

2.2.1 Literature Review

The literature review was conducted using the guidelines set forth by Harris et al as a reference [33]. However, as the primary goal of the literature review was not to assess the strengths of the methodology and relative outcome(s) of each piece of literature but to rather explore and assess the corpus of commonly-utilized bioinformatics tools available, some suggested analytical procedures such as statistical meta-analysis and systematic bias assessment were not conducted. Additionally, given the scarcity of user-centered design usage when developing and evaluating bioinformatics tools [9], specifically tools focused on novice genomics researchers, the scope of the literature review was intentionally left broad in order to best capture as much literary evidence possible for the final analysis and evaluation rubric development.

2.2.1.a Scope and Research Question

The primary scope of the literature review was to survey the existing landscape of bioinformatics tools. Specifically, the review centered around the following research question: “What are the available bioinformatics analysis tools commonly utilized to conduct genetic analyses?”. Additionally, a sub-focus of the review was to focus on tools that claimed to cater towards novice genomic researchers and genome wide association studies.

2.2.1.b Data collection

Literature to populate the review was sourced primarily from the following databases: PubMed, Web of Science, Medline, Bioinformatics Software DB (https://bioinformaticshome.com/db/) and Google Scholar. Searches within these databases were restricted to articles that were peer-reviewed, written in the English language, and published since the year 2003 - completion of the Human Genome Project and year referenced by Clément et al. in their history of bioinformatics tools [18]. **Table 1** shows the search strings utilized to identify relevant literature in the surveyed databases:

Database	Search String
Pubmed	(("bioinformatic") OR ("biomedical informatic") OR ("genomic") OR (“genetic”) OR (“informatic”) OR (“gene”)) AND ((“software”) OR (“tool”) OR (“platform”) OR (“analysis”) OR (“user-friendly”) OR (“UCD”)) AND ("2003"[Date - Publication] : "3000"[Date - Publication]) AND (English[Language])
Web of Science	(("bioinformatic") OR ("biomedical informatic") OR ("genomic") OR (“genetic”) OR (“informatic”) OR (“gene”)) AND ((“software”) OR (“tool”) OR (“platform”) OR (“analysis”) OR (“user-friendly”) OR (“UCD”))
Medline	(("bioinformatic") OR ("biomedical informatic") OR ("genomic") OR (“genetic”) OR (“informatic”) OR (“gene”)) AND ((“software”) OR (“tool”) OR (“platform”) OR (“analysis”) OR (“user-friendly”) OR (“UCD”))
Bioinformatics Software DB	Individual Searches: “genomic”, “gene”, “user-friendly”, “analysis”, “GWAS”
Google Scholar	(("bioinformatic") OR ("biomedical informatic") OR ("genomic") OR (“genetic”) OR (“informatic”) OR (“gene”)) AND ((“software”) OR (“tool”) OR (“platform”) OR (“analysis”) OR (“user-friendly”) OR (“UCD”))

Table 2.1. Search criteria utilized to find relevant literature from different databases. Except for Pubmed, all other databases were manually filtered for the publication date and English language requirements.

After each search result was executed, the resulting studies’ RIS files were exported to Mendeley Web Citation Manager where duplicate articles were removed [34]. After that, the following exclusion criteria was applied to each abstract remove articles that were not relevant to the research question: “no mention of an informatics tool”, “no mention of genetic/genomic data”, “conference abstract”, “dissertation/thesis”, “no peer review”, “not in English”, “published before 2004” and “no mention of example results/analysis”. Once that was done, all remaining articles were then pruned on their full text and subsequently removed of the met one or more of the following exclusion criteria: minimal or no discussion of genomics software, no links or URLs to where the genomic tool can be found, no mention of documentation, no mention of the type of analysis that can be done, and not enough information to reasonably ascertain the types of potential users of the genomics tool.

2.2.1.c Data extraction

Data extraction was conducted to retrieve the following information about each piece of literature:

1. Name of analysis tool
2. Target audience (explicitly named or inferred)
3. Target analysis
4. User-centered Design Considerations
5. Paper/Link to Tool Documentation
6. Data Restrictions
7. True/False to the following criteria: “Cloud-based”, “GUI Present”, “Github/Open-Sourced”, “Tutorials Available”, and “Free-to-Use”

2.2.2 Needs Assessment

2.2.2.a Participant Recruitment

Prior to recruiting participants, an Institutional Review Board (IRB) application for this study was submitted to the University of Washington's Human Subjects Division on August 31st, 2022. This application was approved on September 7th, 2022 with exempt status. After obtaining approval from the IRB, a target sample of 12 study participants were recruited for interviews. This sample size falls in line with the estimate of thematic saturation typically being reached between 10-20 interviews in studies involving qualitative health research [35, 36]. The inclusion criteria for study participants were as follows:

- Graduate student currently belonging to a lab at the University of Washington and Tulane University
- Prior experience with bioinformatics analysis methodologies
- Prior experience with informatics tools and software
- Less than 3 years of experience with genomic analyses

In order to recruit participants, an initial list of potential candidates was compiled using a contact list provided by the doctoral committee chair (sharing contacts across UW and Tulane), the public UW Biomedical and Health Informatics graduate student directory, and public UW and Tulane faculty directories whose labs conducted research involving bioinformatics and genomics. As the interview study progressed, some additional participants were recruited using snowball sampling from interviewed participants after being confirmed in meeting the inclusion criteria. The final potential interview list consisted of 20 potential interviewees: 12 from the University of Washington and 8 from Tulane University. This list was stored in a password-protected file and was only shared with the doctoral committee members. Beginning in

September 2022, potential interview candidates were contacted via email to participate in the interview study, starting with the UW candidates and followed by Tulane candidates. By November 2022, all 20 candidates were invited and out of those 20 candidates, 12 agreed to participate in the study.

2.2.2.b Interviews

The primary investigator developed a preliminary set of questions to ask during a 30-minute, semi-structured interview. These questions were then reviewed by the doctoral committee and a qualitative research expert. An initial interview guide (**Appendix A**) was developed based on feedback received. This interview guide was then used by the primary investigator to conduct two pilot interviews to assess the quality of the interview guide and the content depth being investigated by the interview questions. After the second interview, a revised, more comprehensive version of the interview guide (**Appendix B**) was created to address some of the concepts not covered by the first interview guide. This revised guide was reassessed after conducting 2 more interviews and was thorough enough to be utilized for the remainder of the interview study.

Each interview was scheduled for 45 minutes, which included time for introductions, background on the purpose of the study, informed consent, and the interview itself. Consent was acquired verbally utilizing a consent guide approved by the IRB. A semi-structured interview methodology was utilized when conducting the interviews. DeJonckheere et al. have shown that semi-structured interviews appear to be a good balance between rigor and flexibility and allow for effective extraction of qualitative themes [37]. As the primary goal of the needs assessment was to elucidate the experiences of novice genomics researchers, this methodology was effective in reaching that goal.

After each interview, participants were asked to complete a short demographic questionnaire, approved by the IRB, and this data was stored securely in a password-protected UW Google Drive. Prior to beginning each interview, all participants were asked for verbal consent to audio recording of the interview. These audio recordings were stored in a secure UW Zoom-cloud account. The output files (a .MP4a file) were then exported to an encrypted device and then subsequently uploaded to a password-protected automatic transcription service called HappyScribe [38]. After transcription, all recordings were permanently deleted from the transcription service and the transcripts were downloaded onto the same encrypted device for post-processing. Additionally, once the transcripts were collected, all audio files were permanently deleted from the UW Zoom-cloud account. Finally, the transcripts were uploaded to a local ATLAS.ti project for qualitative analysis [39].

2.2.2.c Template Analysis

In order to analyze the transcripts for emergent qualitative themes, template analysis was conducted [40]. Introduced by Crabtree et al., template analysis is a form of thematic analysis that utilizes hierarchical coding derived from a codebook based on an existing template in addition to allowing for new codes that may appear from the transcripts. In the context of this study, the template that was utilized as the foundational basis for codebook development was the interview guide itself. As the interview guide was slightly altered throughout the course of data collection, the corresponding analysis codebook also went through a series of updates and changes. **Table 2** below shows the finalized codebook that was derived from template analysis and utilized to tag relevant information from each of the transcripts:

Code	**Description**
First experiences	describes their first experiences when they got into bioinformatics/genomics research
Resources used when getting into field	describes what kinds of resources (textbooks, tutorials, courses) they used to help them get into genomics research and conduct experiments
Specific software tool(s)	describes any specific tool or software package used to conduct their experiments/workflows
Emotional reaction	describes any emotional reaction to utilizing specific tool (i.e. was satisfied, frustrated because of _______)
Cognitive reaction	describes any cognitive reaction to utilizing specific tool (i.e. was not able to understand, understood usage and setup easily)
Overall experience	describes any overall experience details that were not captured by the other codes
Benefits of specific tool	describes specific benefits in utilizing analysis tool (features, usability, etc.)
Barriers of specific tool	describes specific barriers in utilizing analysis tool (shortcomings, usability, etc.)
Impact on upkeep/cost	describes any costs on data upkeep or analysis computational costs that were incurred as a result of the tool
Overall impact	describes any overall impact details that were not captured by the other codes
Suggested improvements	describes what improvements would they like to see in the specific tool or more generally to bioinformatics tools being developed in the future

Challenges/concerns with cloud genetics	describes what challenges/concerns exist in the future of performing genomics analysis on the cloud

Table 2.2. Codebook developed throughout template analysis and subsequently utilized to code interview transcripts

In order to arrive at the finalized codebook above, the primary investigator and an independent coder (graduate student with experience in qualitative analyses) worked together to first reach high inter-coder agreement. The metric utilized to assess inter-coder reliability was the Krippendorff's Cu /cu-alpha coefficient [41, 42]. In order to calculate this value, the primary investigator and the secondary coder both independently coded two transcripts with the preliminary version of the codebook. The following methodology was used to perform the reliability analysis in ATLAS.ti:

1. The primary investigator created a clean snapshot of all transcripts (removing all comments and codes).
2. This snapshot was then exported into an ATLAS.ti project bundle file and shared in a password-protected UW Google Drive folder.
3. All coders downloaded this snapshot and coded independently two initial transcripts
4. All coders exported their coded snapshots back to the secure UW Google Drive.
5. The primary investigator downloaded the individual project snapshots and merged them into one to run the built-in ALTAS.ti inter-coder reliability metric.

After this first round of coding, the inter-coder metric was calculated and both coders discussed the current version of the codebook (if any codes needed to be added or revised and any disagreements about the meaning of specific codes). A second round of coding was performed with two more transcripts and again the inter-coder metric was calculated. This time, a Krippendorff's Cu /cu-alpha coefficient of 0.919 was achieved, corresponding to high inter-coder reliability. Once this reliability was achieved, the primary investigator and secondary coder split up the remaining transcripts and coded independently. All coding data was then merged into a single ATLAS.ti project (similar to the methodology used to merge the initial transcripts detailed above).

2.2.3 Evaluation Rubric

In order to synthesize the knowledge gained from the literature review and needs assessment, an evaluation rubric was developed. Evaluation, within the context of professional assessment, is defined as the systematic determination of the quality or value of something [43]. A good evaluation requires knowledge about the evaluand (what you are evaluating), its background, and its context [44]. For this aim, the evaluand was the effectiveness and usability of existing bioinformatics tools within the context of NGRs. A rubric is a type of evaluation framework that "sets out criteria and standards for different levels of performance and describes what performance would look like at each level" [45]. Utilizing the methodology for evaluating bioinformatics tools and training practices from Xie et al. and Tracentenberg et al., an initial set of evaluation criteria were created to begin rubric creation [46, 47]. These criteria were then adjusted to incorporate the needs of NGRs and the strengths and weaknesses of bioinformatics tools gathered from the literature review. The primary investigator then utilized Markiewicz et al.'s primer on evaluation framework development to create a scoring metric for each criteria

[48]. Finally, the evaluation rubric was assessed for quality and consistency by the dissertation committee.

2.3 Results

2.3.1 Literature Review

After searching through the aforementioned databases, a total of 385 potential studies were identified. Of these studies, 196 studies were removed due to being duplicates and 133 studies were removed based on title and abstract screening. Of the 56 remaining full-text studies, 22 were removed due to either a lack of genomic tool availability, lack of documentation, or lack of information regarding the scope and target audience of the tool. A total of 34 studies remained and were utilized in the literature review. **Figure 2.1** details the filtering process.

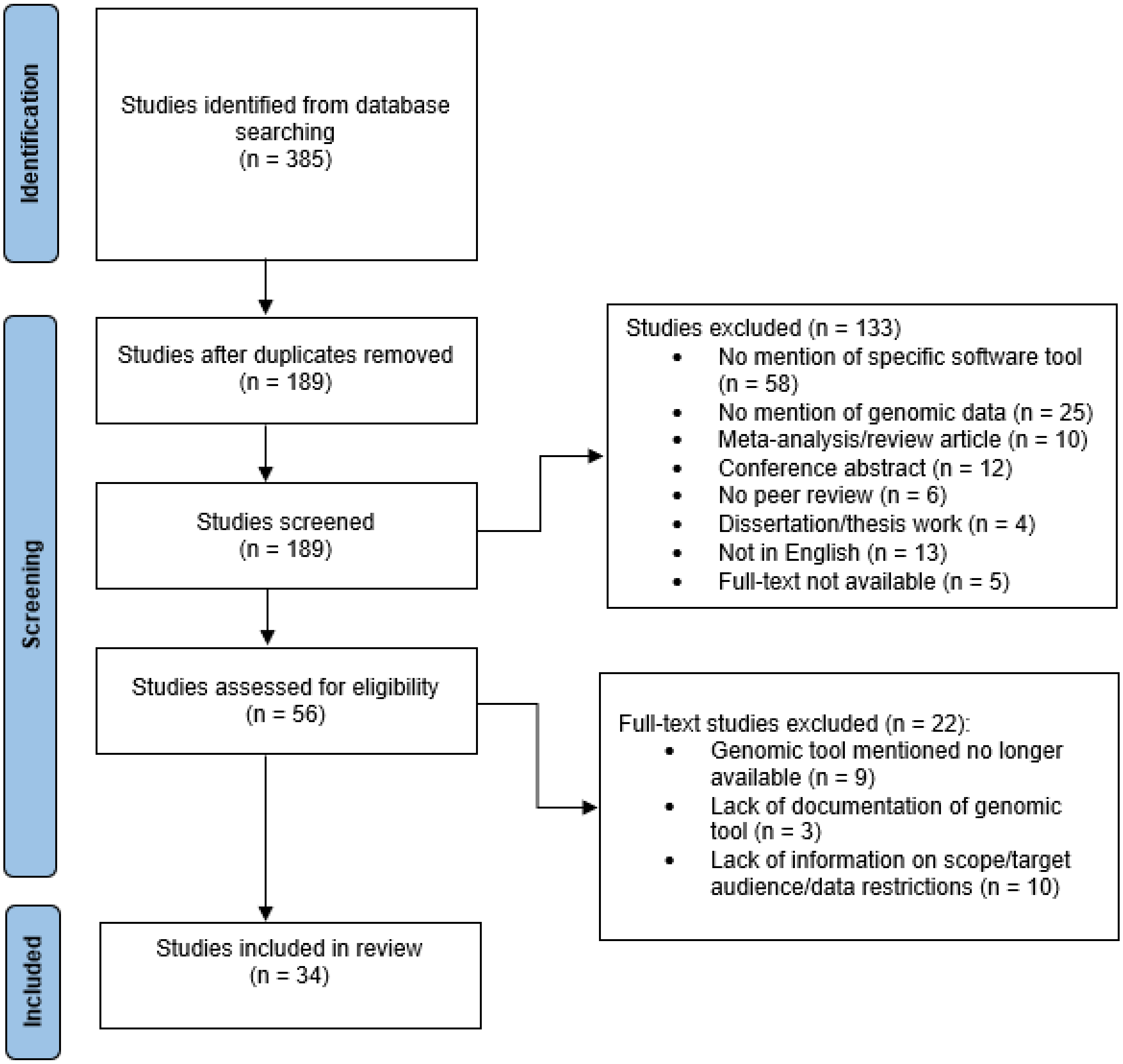


Figure 2.1. PRISMA [49] flow diagram detailing Identification, Screening, and Inclusion of studies during the literature review

Appendix C provides detailed study characteristics for the 34 studies reviewed as part of the literature review. Overall, all studies were published between 2004 and 2022, had a specific genomic analysis tool/package at the time of publication, and that analysis tool was open-source. Target analyses of the 34 studies were diverse and included: genome wide association studies, RNAseq, whole genome analyses, polygenic risk score creation and evaluation, imputation annotation of rare variants, genomic survival analysis, and transcriptomics analyses.

Of the genomics analysis tools from the 34 studies, 32 (94%) had documentation/tutorial content provided, 17 (50%) had a graphical user interface, 8 (24%) were cloud-compatible, 8 mentioned some UCD considerations (24%), and only 7 (21%) studies mentioned a target audience of novice users.

2.3.2 Needs Assessment

Table 3 displays demographic information about the participants. In summary, most participants (N = 12) were female (58%), Asian (50%), non-Hispanic/Latino (92%), had less than 2 years of experience working with genetic data (75%), ranged in age from 21 to 48 (mean = 27) years old, and had varied education and familiarity with genetics.

		N
Age	Mean Median	27 (SD 7) years 26 (range 21 - 48) years
Sex	Male Female	5 (42%) 7 (58%)
Race	White African-American Asian More than 1 race Other	4 (33%) - 6 (50%) 1 (8.5%) 1 (8.5%)
Ethnicity	Hispanic or Latino Non Hispanic or Latino	1 (8%) 11 (92%)
Education Level	Bachelor's Degree Master's Degree Doctor of Medicine	6 (50%) 5 (42%) 1 (8%)
Familiarity with genetics (out of 5)	Median	3 (range 1 - 4)

Years of experience	Less than 1 year	4 (33%)
	1 - 2 years	5 (42%)
	More than 2 years	3 (25%)

Table 2.3. Participant Demographics from Needs Assessment

Emotional and Cognitive Reactions

Participants’ emotional reactions to first utilizing bioinformatics tools varied when asked “ How did you feel when you were working with your analysis tool?”. Some participants reported feelings of satisfaction and excitement:

> *“Well, I was super excited about it, actually. This was actually my first time that I got to look at real data and trying to figure out how to use it and make it useful.” (P12)*
>
> *“Definitely felt a sense of like satisfaction that I had kind of worked through and gotten things [SVABA] to work.” (P8)*

However, the primary emotions shared by the majority of the participants were frustration, confusion, and being overwhelmed:

> *“Honestly, that was terrible for me ... a starter, the documentation and the reason you run through the pipeline [PLINK] is it's really hard, like, for a starter to figure out every component in that. So that's going to cause a lot of headache.” (P2)*

> *“Definitely overwhelmed at the beginning, I think, because mostly kind of a lack of domain expertise.” (P8)*

> *“My first experience was with bulk RNA sequencing [Galaxy] ... and it was extremely confusing until all of the preprocessing was done ... Overall, everything was really confusing and remained confusing until, honestly, maybe a year ago.” (P1)*

The emotional reactions above were mirrored in the participants’ cognitive reaction and understanding when asked “What concerns or questions did you have about utilizing (genomics software) when you first started using it?”. Most participants reported a lack of domain knowledge and technical knowledge needed to properly utilize their analysis tool:

> *“So, I think that it [Bioconductor GWAS] feels a lot of them feel like a black box, essentially, where at least in my experience, you don't know what's going on. You don't know if it's working. You put the arguments in and you pray, and then when it finishes, you'll see if it worked or not.” (P6)*

> *“It definitely caused a lot of frustrations because of their documentation and some of those error messages. It's not really clear. When you also Google that [the errors] there are a bunch of other people googling the same things you could find and the answers are actually various ... I think it's really frustrating when you're doing that. Sometimes it just doesn't work.” (P2)*

> *“And so I was digging through the Plink documentation and trying to figure out what are all these parameters, what are they doing? It was hard to figure out where I needed to change the Plink commands for my own purposes because I just didn't really understand*

> *how it was working yet. As I worked with it more, I don't know if I ever fully understood what was going on." (P4)*

There were a few participants who reported sufficient initial understanding of how their tool worked and increased understanding after following tutorials or working with their tool over the span of the project period:

> *"It [AllOfUs] didn't really have anything that I thought that they were missing. I thought they were pretty clear on how to use everything and they're very informative. And then they also add a more detailed help support system in terms of like, if you did get lost in trying to understand how to use one of their specific tasks that they had." (P5)*

> *"I feel it's [MaxQuant] fine because it's not like, too complicated software. I have some experience with similar software running process before." (P9)*

Benefits

When asked "Thinking back, what do you see as positives about utilizing _________ (their analysis software)?", most participants reported they appreciated the provided tutorials and documentation, how efficient and robust their tool was (even when the dataset was very large), and built-in privacy and security precautions:

> *"I think the biggest plus for me is that it has a very robust documentation. It still has some holes in it, I think, but at least of the software tools that I was using. And it's like, so*

> *this is what you do. These are the arguments. These are the possible parameters. This is what you should have. Sometimes it has some gaps in it. At least it's pretty simple to go from zero to being able to use it." (P6)*

> *"The positive side is I really appreciate the efficiency. I think some of the remote clusters have more cores and rams. You can finish a job really, really quick, even dealing with large sample data sets. So that was so powerful in my opinion. So I definitely appreciate their [PLINK] integrations in their system and also their running speed, efficiency." (P2)*

> *"They do have a lot of positives such as double authentication and different ways of making sure that these data sets are used for a purpose of some sort, so you can't just log in and access them like one specific person data set. So I like how it is. Very private, I guess." (P5)*

Barriers

When asked "Thinking back, what do you see as negatives about utilizing _________ (their analysis software)?", most participants reported trouble during initial setup, lack of thorough error messages, lack of user-friendly interfaces, and challenges sharing results outputted to colleagues or other pipelines:

> *" I think often times some tools are set up in a way where it's not immediately evident, like what you need to do to reconfigure them to non-test data that's included as part of the tool. Also, little frustrating, you know, just running into different errors. It was*

> *frustrating because you would submit something and you wouldn't know for a while whether or not it worked. And so then you get the error or whatever a few hours later and then you have to debug and you feel like you've lost a few hours." (P8)*

> *"I think negative status, apart from the documentation, I would say some of the error message, which is not clear for debugging ... and when you Google that, there's a lot of different answers. Some says that it's a RAM issue, some says it's something like the version doesn't work in that way." (P2)*

> *"The thing that takes the most time is like setting up all your data to get it in a format that Plink will be okay with and then outputting your results and doing all the filtering and stuff. The tough part is just wrangling all of the millions of files that come out of running something like a GWAS." (P4)*

> *"There's probably more efficient ways of doing this, for sure. This is not a workflow where I can easily package it up and give it to someone else and say, hey, you can use this and run it. It's definitely confusing for anyone who's not in it already. Yes, there is that." (P1)*

Impact on Cost and Research Experience

No participants reported any direct monetary costs they faced as a result of utilizing their bioinformatics tool when conducting experiments. However, some participants did anticipate future storage costs of big datasets.

Four participants reported they took workshops and courses to improve their technical skills while beginning to utilize their respective tools, and most participants indicated interest in taking coursework or attending future workshops to future build their biological, statistical, and programming knowledge.

Suggestions/Recommended Improvements

Despite facing diverse initial barriers to entry, most participants reported they benefited overall from learning how to utilize bioinformatics software. Several suggestions to improve both their experience and the experience of future researchers were provided upon asked the question “Do you have any recommendations for changes or additions to _________ (genomics software) that would aid in your research?”. Most participants (9/12) reported the need for better documentation and tutorials (with resources written in layman’s terms), an interactive analysis environment, and clearer error messages:

> *“I expected more documentation on the whole thing. They have some documentation, but they don't have many examples. So I think they could really improve in that regard.” (P3)*

> *“I would definitely wanted them to provide more examples of the workflow and how the tutorial would go. Like, if you were doing some genomic analysis and you need to say grab or filter specific regions, what do you do and what do you most of the frequently asked questions or errors.” (P2)*

> *“Pretty much every project I've done, having some kind of visualization is really important, so maybe that's not important for everyone. Like, some people are really,*

> *really comfortable just doing everything through the command Line. But I find that really cumbersome ...so having that real time interactive interface has been really important for me to think of other things that I've used." (P4)*

Several participants (4/12) reported need for extra information to be readily available during analysis in some sort of "information hub":

> *"For me, I prefer a big picture for my learning. Like if I can know how is this data generated and then once I got this data what is the next step? So I think making it a whole process like the tutorial that is to help students to make a big picture to understand what they are doing." (P7)*

> *"Honestly, I wish any tool's methods paper had, like, explain it like I'm five section or something. Because, man, some methods papers are just so convoluted and really difficult to understand, and some are, like, not like that. Also, I think I would have wanted the tool to like point to useful resources, like useful background reading. So I feel like a section on, whatever, three or four bullets of important background reading to help you understand this tool or something like that." (P8)*

Lastly, some participants (2/12) suggested a need for better sharing of analysis files and pipelines between lab mates and also to other researchers.

> *"I guess a pretty simple disadvantage, I think, is it's not for me, but if I was sharing my code with somebody, it's all in the command line. It's a little bit easier doing it from a data transparency in a sharing way than to send somebody a shell script that you ran with Plink in it. I think that might be harder for somebody who doesn't have command line experience." (P11)*

> *"I definitely want it [my analyses] to be more shareable." (P1)*

Thoughts on Cloud Analyses

When asked "As the NHGRI and other genomics research organizations are moving toward cloud computing, what challenges/concerns do you anticipate facing when conducting genomic research in the cloud?", most participants seemed optimistic about the notion of cloud computing for bioinformatics, but were skeptical about topics such as privacy and security, transition time and training needed, and cost:

> *"I guess one concern for the cloud is who necessarily has access to the cloud because it's like if we only are allowed to get like, certain portions of it. But is there someone who's maintaining trying to keep the cloud secure?" (P5)*

> *"From my perspective, I think there are a couple of things I mentioned already. What is the cost structure of it? What do I need to know to set it up correctly from a technical perspective? Those kinds of things, of course, like if you're using health data, you have to think about HIPAA and you know, proper access control, though I think that that's pretty*

well set up or like thought out with those cloud computing things. Like maybe better than any one random person's homegrown server." (P12)

"I think overall, it's pretty much easier when you are doing research in the cloud because then you don't deal with issues like data storage or security. But one challenge is cost. And I think even when we were doing that research with Microsoft, we realized that it's not usually very transparent how much you're spending." (P3)

"I think that the barrier to entry becomes higher. Yeah, if that makes sense. Because suddenly you're not just talking about learning how to use Plink or learning how to use Hail or learning how to use Bioconductor suddenly. Now you also have to learn Spark or Hadoop or any of these other technically programming languages to interface with the cloud. And people are going to have to get a different sort of preparation before they can actually carry out this type of analysis. (P6)

2.3.3 Evaluation Rubric

Table 4 below shows the evaluation rubric created after incorporating existing evaluation criteria for bioinformatics tools and adjusting those criteria to include knowledge gained from the literature review and needs assessment. The criteria include robustness of genetic analyses, presence of tutorials/documentation, restrictions on data usable by the tool, accessibility of the tool, reproducibility and intractability with other tools, and finally user-centered design considerations taken by the developers when creating the tool.

Evaluation Rubric for Bioinformatics Tools				
	Level 0	Level 1	Level 2	Level 3
Criteria	**Unacceptable**	**Marginal**	**Acceptable**	**Exceptional**
Robustness of Genetic Analyses	Only capable of one very specific type of genetic analysis (fixed parameters) and lacks ability to scale	Capable of one or two specific analyses but able to scale to large datasets	Supports many genetic analyses, can scale to large datasets, but lacks clear vision for use of tool	Many genetic analyses supported or able to be supported (users can create/share workflows), clear vision for tool's usage
Tutorials/Documentation	Had little/no documentation or tutorials about how to setup or utilize tool to perform analyses	Had documentation/tutorials but was difficult to understand/outdated	Had documentation and tutorials but lacked sample data or references to publications	Clear, updated documentation with examples and sample data along with links to publications that utilized the analysis tool
Data Restrictions	Supported only one or two specifically formatted data types	Supported specific formatted data types but did not allow use of new data	Supported specific formatted data types and allowed the upload of new data	Supported robust data types and allowed for the upload of new data
Accessibility (Cloud/Free-to-Use/Open-Source)	Inaccessible to the target userbase, behind paywall, proprietary code with little option of how to use (their own platform)	Somewhat accessible to the target audience, not open-sourced, may not be free-to-use	Mostly accessible to the target userbase (some wait if user needed to be validated), some open-source components, and	Very accessible tool with both local and cloud-based options, open-source and free-to-use

			free-to-use	
Reproducibility/Interaction with Other Tools	Does not have built in functionality for reproducing experiments or interacting with other software tools	Has built functionality for reproducing experiments but does not interact with other software tools	Has reproducibility functionality built-in and able to interact with other tools but in a very limited range	Excellent built in features for reproducing experiments and is able to interact with other tools with clear guidelines and input/output types
UCD Design Considerations (Setup, Usage, Evaluation)	Had little to no usability considerations during the design, development, or evaluation of genetic tool	Had some usability considerations but only during one or two phases of the developmental cycle	Had some usability considerations during most of the development phase but lacked clear interaction with the target user base	Utilized usability considerations during all stages of the development process with clear interaction with target userbase
Overall Performance	**Unacceptable**	**Marginal**	**Acceptable**	**Exceptional**
Points Required	0-6	6-11	12-16	16-18

Table 2.4. Evaluation Rubric for Bioinformatics Tools within the context of NGRs

2.4 Discussion

Using broad strokes, this aim captures many of the existing challenges present in the current landscape of bioinformatics software tools, such as missing or incomplete documentation, difficulty in setup/support, challenges in reproducibility, and lack of UCD considerations during design and deployment. However, the richness of the data collected during the literature review and user interviews offer more nuanced perspectives as to why these challenges are relevant, how the challenges relate to one another, and how to potentially overcome them in future bioinformatics tools. In the following sections, we will explore these emergent perspectives and offer a deeper understanding of the needs of NGRs and their role in the future of bioinformatics.

2.4.1 Landscape of Bioinformatics Tools

From the results of the literature review, we can easily identify common characteristics shared by most pieces of genomic software that is widely used and highly cited in the bioinformatics space over the past two decades. While almost all tools were open-source and offered some sort of documentation/tutorial content, very few tools noted their target user audience or any component of the UCD framework. On top of that, the tools that did mention UCD principles only did so briefly in the introduction and discussion sections and did not actively incorporate them during the design and deployment of the proposed tool. Studies have shown that poor usability considerations when designing instruments for shared use often leads to failure of sustained use [13]. Furthermore, as only 7 out of the 34 tools evaluated mentioned novice users, most genomics tools currently widely-used and cited fail to align with the NHGRI's 2020 Strategic Vision, specifically the sections for breaking down barriers that impede genomic research and building/improving a robust foundation for future research [26]. These are not straightforward barriers but will be crucial to overcome in order to assist NGRs and build a strong, robust workforce for the future.

2.4.2 Needs of Novice Genomic Researchers

The user interviews conducted in this study offered detailed insight into the challenges routinely faced by NGRs during setup, exploration, experimentation, and sustained use of a variety of different bioinformatics software. Almost all NGRs interviewed reported being overwhelmed and confused during the initial adoption stages and struggled to find adequate resources to help guide their learning process. A key source of this confusion was caused by either lack of domain expertise or insufficient technical knowledge needed to operate the software tool. Participant 6 sums it up nicely by saying:

> "*So, I think that it [Bioconductor GWAS] feels a lot like a black box, essentially, where at least in my experience, you don't know what's going on.* ***You don't know if it's working. You put the arguments in and you pray, and then when it finishes, you'll see if it worked or not.****" (P6)*

This sentiment of missing information is further supported by the call for more detailed documentation, easy to access tutorial content, and clearer error messages by almost all participants. While solutions regarding missing information for novice users exist already in a variety of other disciplines, bioinformatics software tools, especially those currently introduced to NGRS, are not yet thoroughly polished to support novice users [9, 28].

Additionally, most participants reported challenges in reproducibility and shareability of their findings. While research reproducibility is already seen as a crisis by many investigators, this barrier is especially critical for NGRs [50, 51]. Without proper methods to reproduce experiments, NGRs face increased frustration, delay in software adoption, and an overall deficiency in connecting theoretical concepts to practical applications. Potential solutions to this problem that future software tools can consider include integration of workflow description language (WDL), saving prior experiments in a format that can be rerun, providing guided content as part of the documentation, and having available support beyond documentation (such as support email or forums).

2.4.3 Rubric for Evaluation

Combining the findings from the literature review and the insights gained by interviewing the NGRs, an evaluation rubric was created that can be utilized to grade existing and future bioinformatics tools. This rubric acts as a summary of key components needed for software tools to cater to the diverse needs of both NGRs and experienced users. Of the key components detailed in Table 4, UCD design considerations with respect to tool setup and detailed supplementary information are the most important considerations that one should incorporate when designing usable software, as reflected by the user interviews. Existing studies that utilize these components efficiently report high throughput of results, better user understanding, and more sustained tool use [11, 30, 32]. As this is an exciting time for genomics research, it becomes increasingly important to critically evaluate existing tools and develop new ones that will help build a strong foundation for future exploration.

2.4.4 Limitations and Suggested Improvements

Despite the literature review of existing bioinformatics tools covering a wide variety of databases and potential sources of commonly used software, there are several limitations that exist which impact its accuracy. All of the final articles were in English, failing to capture any software tools that were published in non-English journals, and were selected based on the primary investigator's (H.P.) interpretation of the research question. Incorporating additional peer-reviewed sources and a different interpretation of the research question may lead to different final studies used.

Although the user interviews gathered rich qualitative data that offered a more nuanced perspective into the challenges faced by NGRs, a relatively small number of participants were interviewed. However, despite this, code groups and thematic domains did not appear to change significantly after 10 participants, indicating theoretical saturation. Furthermore, participants were only selected from the University of Washington and Tulane University and were mostly of self-identified White and Asian descent. Their training and experience may not reflect the views of NGRs from other institutions and backgrounds. For future investigations, it is important to include a more diverse participant pool in order to capture viewpoints that were not present in this study. Finally, it is important to recognize the nature of qualitative research itself and the impact the primary investigator (H.P.) played during the analysis portion. While an independent coder was involved during the analysis process, it was the primary investigator who organized and synthesized the findings. The final results only represent one possible interpretation of the available data and many other interpretations may exist depending on the investigator.

2.5 Conclusion

In this aim, we demonstrated that while the number of existing bioinformatics tools is quite large, among those that are widely-cited and commonly used, very few provide thorough documentation, easy onboarding, and actively incorporate user-centered design principles during development and deployment. Through interviews, we additionally uncovered many challenges faced by novice genomics researchers during their adoption of these bioinformatics tools and provided suggestions to better help these critical users as they begin their journey in genomics research. Lack of information to assist limited domain knowledge or computational experience was the driving factor behind most of the challenges faced by novice users. Finally, utilizing the two sources of knowledge gathered, we created an evaluation rubric on which existing and future bioinformatics software can be assessed within the context of novice users. In the next chapter, we will utilize what we learned from this aim to inform the creation of a more user-friendly bioinformatics tool. Iterative involvement of target users is critical in creating sustained usage of any tool and bioinformatics software are no exception.

CHAPTER 3: USER-CENTERED, CLOUD-AGNOSTIC GWAS APPLICATION (AIM 2)

Abstract

Novice genomics researchers (NGRs) are a diverse group of keystone individuals entering the bioinformatics space who have historically been underrepresented. Although most newly released bioinformatics tools claim to offer user support, very few actually utilized user-centered design principles during the development process. As a result, many tools published and shared today are underutilized and lack sustained use. In order to incorporate the needs and challenges faced by NGRs, in this aim, we created user personas that characterized different subgroups of NGRs and utilized those personas to inform the design and development of a user-friendly GWAS analysis tool (UF-GWAS). Three distinct types of subgroups/personas emerged: users with limited domain knowledge but reasonable computational experience, users with intermediate domain and computational experience, and users with high domain knowledge but limited computational experience. Each of these subgroups had their own strengths and weaknesses and these were repeatedly referenced during the design of UF-GWAS. The final product that was created was a cloud-agnostic GWAS analysis tool capable of being easily deployed on any major cloud computing provider. UF-GWAS utilizes a Docker container to neatly package a JupyterLab instance which allows users to run GWAS analysis and share results quickly and easily. As informatics continues to reach a more global and diverse audience, more NGRs will be trained and attempt to conduct genomics research. In order to best facilitate their transition, better and more usable bioinformatics software needs to be developed and evaluated.

3.1 Introduction

A Genome Wide Association Study (GWAS) is a genomics analysis methodology used to identify variants within the genome that are statistically associated with a particular phenotypic trait or disease [52]. Over the past two decades, GWASs have had an extraordinary impact on the discovery and characterization of the genetic architecture of many common diseases, such as breast cancer, colon cancer, diabetes, cholesterol, coronary heart disease, Alzheimer's, and many more [53, 54]. The results generated from conducting GWASs can be further utilized in a plethora of downstream analyses such as heritability estimates, calculating risk of disease onset and progression, informing drug development and targeting, and teasing apart potentially causal relationships between associated risk factors and actual health outcomes [55]. However, there are several limitations in conducting a successful GWAS such as the need for large sample sizes (challenges related to data coordination and collaboration), population stratification (challenges related to data diversity and equity), associations instead of causations (challenges related to a majority of GWAS findings are variants in non-coding regions, lack of understanding of underlying regulatory mechanisms), inaccurate phenotype definitions and measurements, intense technological demands, and slow pace in translation and incorporation into clinical care [56].

In order to offset some of these challenges, several global organizations have begun to mobilize standards and best practices to current and future research ventures. In the United States, the National Human Genome Research Institute (NHGRI), a subset of the National Institutes of Health (NIH), is one such organization. In their 2020 strategic vision on the future of improving human health through genomics, the NHGRI advocated for the need for a more robust foundation needed to progress research [57]. This includes better standard development and usage, better data sharing and collaboration in analysis pipeline development, and breaking down barriers of entry to non-researchers in hopes of stimulating more compelling research projects in the future. One concrete realization of the NHGRI's vision is the development of their Genomic Analysis, Visualization and Informatics Lab-space (AnVIL). The AnVIL is a cloud-based genomic data storage and analysis platform that aims to tackle several of the challenges listed above in addition to providing an entrypoint to FAIR (Findable, Accessible, Interoperable, Reusable) genetic analyses for both novice and experienced researchers [58, 59]. The AnVIL is not alone in this cloud-based research space. The All of Us Research Program, another NIH-sponsored research project with goals to boost diversity and scale of genetics research in the United States, with a goal of over one million highly diverse participants with deep phenotyping from electronic health records and genomic sequencing along with a variety of other data, also boasts a cloud-based data exploration and analysis platform [60]. In their own ways, both programs hope that cloud-based data storage and analysis will help address challenges in data coordination, improve data quality and accessibility, and provide a solid foundation for future research.

Cloud computing, defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources" by the US National Institute of Standards and Technology (NIST), has been around since the late 1990s and early 2000s [61]. Utilized in a wide variety of fields, such as manufacturing and distribution, management, retail, economics, engineering, big data analytics, and even farming, cloud computing's growth and rapid adoption has become ubiquitous in our daily lives [62, 63]. In genomics, cloud computing has only recently begun to be widely recognized and utilized for both data analysis and collaboration [64]. With roots in the cancer community, cloud genomics started as a way to share data and information between collaborators in the form of data portals and data clouds, such as UCSC Genome Browser and Bionimbus Protected Data Cloud developed by the University of Chicago [65, 66]. However, as the cloud began to house more and more data, there came a need for data harmonization and the formation of standardized cloud platforms called data commons [67]. Examples of such early data commons include the three NCI Cloud Pilots: the ISB Cancer Genomics Cloud, FireCloud , and the Cancer Genomics Cloud [68–70]. These programs paved the way for more recent advances in cloud-based data commons such as NCI Genomic Data Commons, utilized by over 100,000 researchers in 2018 [71]. Other examples of modern data commons and analysis platforms have already been introduced earlier via the AnVIL and All of Us platforms. The major advantages of utilizing cloud computing for genomics is in two key areas: reproducibility and global access [64]. With the ability to create customizable environments and workflows, genomics researchers are able to not only utilize powerful computation resources on a pay-by-use basis but also share their analysis pipelines with others easily for peer-review and generalization. Furthermore, by being accessible globally in the future, there are fewer restrictions on collaboration between researchers leading to more informed and diverse results. However, with any new technology, there are several disadvantages with conducting research in the cloud. Some of these disadvantages include: reluctance by researchers to switch to cloud computing, cost uncertainties when conducting cloud experiments, training and adoption time for both novice and experienced researchers, and the growing privacy and ethical considerations with housing and utilizing healthcare data [72, 73]. Although active work is being done to address these challenges, the road for widespread adoption of cloud-based genomics data collaboration and analysis is long and will require time and funding from both public and private entities.

Despite being a rapidly growing field, very little work has been done to characterize the usability of cloud genomics, especially in the context of novice genomics researchers (NGRs). Some work has been done to address usability challenges that arise from the implementation of cloud-based healthcare applications, such as cloud computing in pediatric care and mobile cloud platforms for diabetes prevention [74, 75]. However, these studies fail to properly address the learning and adoption challenges faced by novice genomics researchers revealed in Chapter 1. Novice users of cloud-based systems require careful usability considerations such as easy log-in and setup of the application/tool, clear documentation of proper usage and potential barriers that may arise, transparency in cost and time needed to utilize the system, and concrete ways to develop confidence in using the cloud system [76]. Pavelin et al. accurately warn "the joy of an early release lasts but a short time. The bitterness of an unusable system lasts for years" [9]. Lastly, as detailed in Chapter 1, NGRs play an important role in the future of genomics research and developing usable tools can help facilitate NGRs in order to democratize research from mostly well-funded organizations to more diverse institutions and environments.

To fill the gap of proper usability considerations in bioinformatics tools development, especially those present in the context of NGRs and cloud-based genomics software, in this aim, we will employ a mixed-methods approach that will consist of persona creation and the development of a cloud-agnostic, containerized application. The needs assessment and literature review results from Aim 1 will inform the key features of both the design and development of this tool. We predict proper utilization of UCD principles will enable the creation of a robust, user-friendly tool that will be compatible across cloud platforms capable of efficiently and accurately running genetic analyses such as GWAS.

3.2 Methods

3.2.1 Persona Development

Personas were created in order to focus on the specific user needs gathered from interviewing the NGRs. Personas are fictitious, specific, and concrete representations of target users [77, 78]. They allow for more focused product design and development by focusing on specific users instead of everyone leading to a more user-centered application, an example includes patient-decision making personas utilized by Holden et al. to characterize the different types of chronic heart failure patients [79]. Usability.gov provides a set of steps for effective persona creation and refinement [80].

3.2.1.a Data collection and initial persona creation

Data collection and user research needed for persona creation was conducted by the primary investigator in Chapter 2, needs assessment of NGRs. Utilizing both the demographic information of the participants and their responses to the semi-structured interview questions (**Appendix A**), two one-page personas were created by adapting the guidelines set forth by Robertson et al. [81]. These personas represented two types of potential NGRs (novice computational user but versed in biological knowledge and novice biological expert but versed in computational techniques). Personas also included fictitious names, profile pictures, demographic information, and research goals.

3.2.1.b Persona refinement

After speaking to a qualitative research expert and a few participants from the needs assessment in Chapter 1 for feedback on the developed personas, the initial personas were refined to include one additional persona (NGR who is intermediate in both biological and computational knowledge) and additional information about their research environment. The final set of three personas were then reviewed by the primary investigator for consistency and information depth before being utilized for downstream software tool design and development.

3.2.2 Cloud-agnostic software tool

In order to develop a user-friendly, cloud-agnostic GWAS application, several types of software technologies were utilized. Using broad strokes, the overall application consisted of several Jupyter Notebooks, written in Python and Markdown, hosted on a JupyterLab server. All of the data processing and data analysis used to perform the GWAS was done using the Hail Python library. The JupyterLab server was housed inside of a Docker container which was then converted into a Docker image for easy initialization on different local and cloud infrastructures.

3.2.2.a Jupyter Notebooks and JupyterLab

The Jupyter project is a non-profit organization whose goal is to provide open-source software, open standards, and services for interactive computing [82]. Two key components of the Jupyter project include JupyterNotebooks and JuptyerLab. JupyterNotebooks are widely-used interactive programming environments that allow one to conduct data analysis specifically for easy reproducibility and sharing of experiments [83]. JupyterLab extends this functionality by adding an IDE-like interface in which users can interact with multiple notebooks simultaneously along with other key aspects of a project such as the terminal and file navigation. Both JupyterNotebooks and JupyterLab have been widely used across many research domains and ease many barriers faced by novice researchers entering complex data analysis [84]. However, similar to the problem faced by bioinformatics tools in the previous chapter, JupyterNotebooks also suffer from challenges related to sustained use and lack of thorough documentation. To offer mitigation strategies for these barriers, Pimentel et al. studied over 1.4 million notebooks and reported key components present in the most popular and widely used notebooks [85]. These key components included detailed markdown cells explaining the overall goal of the analysis along with instructions throughout the notebook, easy dependency configuration/import handling, and the use of relative paths for data and images. Insights from Pimentel et al. along with those from Rule et al.'s guidelines for writing and sharing JupyterNotebooks were used in creating the JupyterNotebooks utilized in the final bioinformatics tool developed during this aim [86].

3.2.2.b Hail

Hail is an open-source Python library for scalable genetic data exploration and analysis [87]. At its core, Hail extends the functionality of commonly used genetic analysis tools such as PLINK and Bioconductor packages by using a custom data storage primitive called a MatrixTable. By storing data in a Hail MatrixTable, downstream analyses such as GWAS can be parallelized easily and scaled depending on the available computational resources. Hail has already begun to be utilized for bioinformatics research. Studies that utilized Hail to run their backend analyses include Atkinson et al.'s use of local ancestry to enable the inclusion of admixed individuals in GWAS, Karaca et al.'s characterization of risk loci associated with asthma and lung function, and De Lillo et al.'s PheWAS of TTR and RBP4 genes to study the genetics of hereditary and wildtype transthyretin amyloidoses [88–90]. Furthermore, Krissaane et al. have shown Hail can be cost-efficient and effective on both local high performance clusters and in cloud infrastructure such as Google Cloud and Amazon Web Services [91].

For this aim, Hail was utilized as the primary backend library for data cleaning and analysis. This included functionality for loading compressed Variant Call Format (VCF) files, variant and sample quality control, merging in phenotype information, storing intermediate datatables as Hail MatrixTables, running GWASs and statistical tests, and outputting graphs such as Manhattan plots and Quantile-Quantile plots.

3.2.2.c Containerization

One of the biggest challenges in the development and deployment of any application is the extensive setup required. Allocated required computational resources, downloading and updating required libraries and prerequisites, verifying version compatibilities, and configuring networking and storage infrastructure are just some of the barriers that need to be overcome before even launching a piece of software. One solution to ease that burden is containerization. Containerization is “the packaging of software code with just the operating system (OS) libraries and dependencies required to run the code to create a single lightweight executable—called a container—that runs consistently on any infrastructure” [92]. A developer can utilize containerization to create containers which are units of software that package up application code and all its needed dependencies (including default dependency environmental variables and start-up parameters).

Docker is an open-source, industry standard platform for developing, shipping, and running applications using containers [93]. Docker can be used both locally and in the cloud. For cloud utilization, Docker containers must first be converted into Docker images which can be deployed on all large cloud service providers available today (Amazon Web Services, Google Cloud, and Microsoft Azure) [94]. These Docker images can also be shared, downloaded, and tailored by future developers to meet specific needs. In bioinformatics, Docker has been utilized for education and research. Examples include the ORCA project, which utilized containers to provide a ready-to-go bioinformatics suite for external collaborators at the University of British

Columbia, and BioPortainer, a user-friendly bioinformatics workbench capable of running both command-line and GUI-based bioinformatics analyses [95, 96].

For this aim, the Jupyter Notebooks and JuptyerLab server were packaged into a Docker container whose base image was the open-source Hail Docker image "hailgenetics/hail:0.2.98" (found at https://hub.docker.com/u/hailgenetics). This image contained all required dependencies to run Hail and was extended to also include Jupyter Notebooks and JuptyerLab.

3.2.3 GWAS on *Clostridioides difficile*

In order to test the performance and accuracy of the tool developed during this aim, a GWAS was run on the Electronic Medical Records and Genomics (eMERGE) dataset [97]. The phenotype of *Clostridioides difficile* infection was utilized for testing purposes. *Clostridioides difficile* infection (CDI) is the leading cause of nosocomial diarrhea in North America, and once acquired, secretes a toxin that causes watery diarrhea which can progress to severe pseudomembranous colitis, toxic megacolon, and sepsis [98, 99]. The phenotyping and quality control configuration was taken from the work done by Ferar et al [100]. In summary, an additive model was used to assess the disease susceptibility impact of the minor allele at each genetic position, while using the following factors as covariates: age, body mass index, sex, ancestry, nursing home status, chemotherapy, diabetes, HIV, transplant medications, corticosteroids, and medium or high-risk antibiotic exposure.

In order to perform the GWAS, a cloud configuration that utilized Hail with one driver node (with 4 CPUs and 16GB of RAM) and 10 worker nodes (with 16 CPUs and 60GB of RAM) was used, totalling 164 CPU units and 616GB of RAM). This configuration was utilized to repeat the experiment for a total of 10 GWAS runs and selected based on the work done by

Krissaane et al. during their analysis of utilizing Hail to perform GWAS on different cloud platforms [91].

3.3 Results

3.3.1 Personas

After creation and review of initial personas, a set of three fictitious representations of novice genomics researchers that would interact with the software tool were finalized. **Table 1** below briefly describes each persona and its motivation from the user data collected in Aim 1. The full personas can be found in **Appendix D**.

Name	Characteristics	Brief Description
Orlando Mccarthy	computational experience but lacks biological domain knowledge	Orlando is a motivated 1st year graduate student with a background in public health. While he possesses intermediate programming abilities, he lacks a strong biological foundation to perform complex analyses. To improve, Orlando routinely seeks out sample datasets that he can use on his local high performance computing (HPC) cluster to run training experiments.
Irene Young	balanced computational and biological knowledge	Irene is a research consultant who has experience running and leading biological experiments. She is still relatively new to cloud computing and is seeking ways to optimize her workflows, with a focus on ease of setup and shareability.
Scott Foster	biological analysis experience but lacks computational knowledge	Scott is returning to graduate school after his medical training. He possesses strong biological knowledge but struggles with some of the computational aspects of genetic analyses. To improve, he is learning how to utilize commonly used software and programming languages.

Table 3.1. Brief description of personas

From the table above, we can recognize that each persona represents user groups that have their own unique strengths and weaknesses. User-centered design principles would recommend incorporating these into the final design of the software tool. In more detail, Orlando, with his limited biological domain knowledge, would prefer access to easy-to-digest reference material in addition to background literature with sample datasets for quick experiments to reinforce any knowledge gained. Irene would value built-in reproducibility of her workflows, with some emphasis on cloud-based shareability. Lastly, Scott would prefer an analysis environment suited for novice programmers with materials provided for onboarding such as tutorial content and ways to test his understanding.

3.3.2 Cloud-agnostic software tool (UF-GWAS)

Using the knowledge gained from the persona creation and the user interviews from Aim 1, UF-GWAS (user-friendly GWAS), a cloud-agnostic GWAS analysis software tool was created. **Figure 3.1** below details the software schematic utilized to create UF-GWAS.

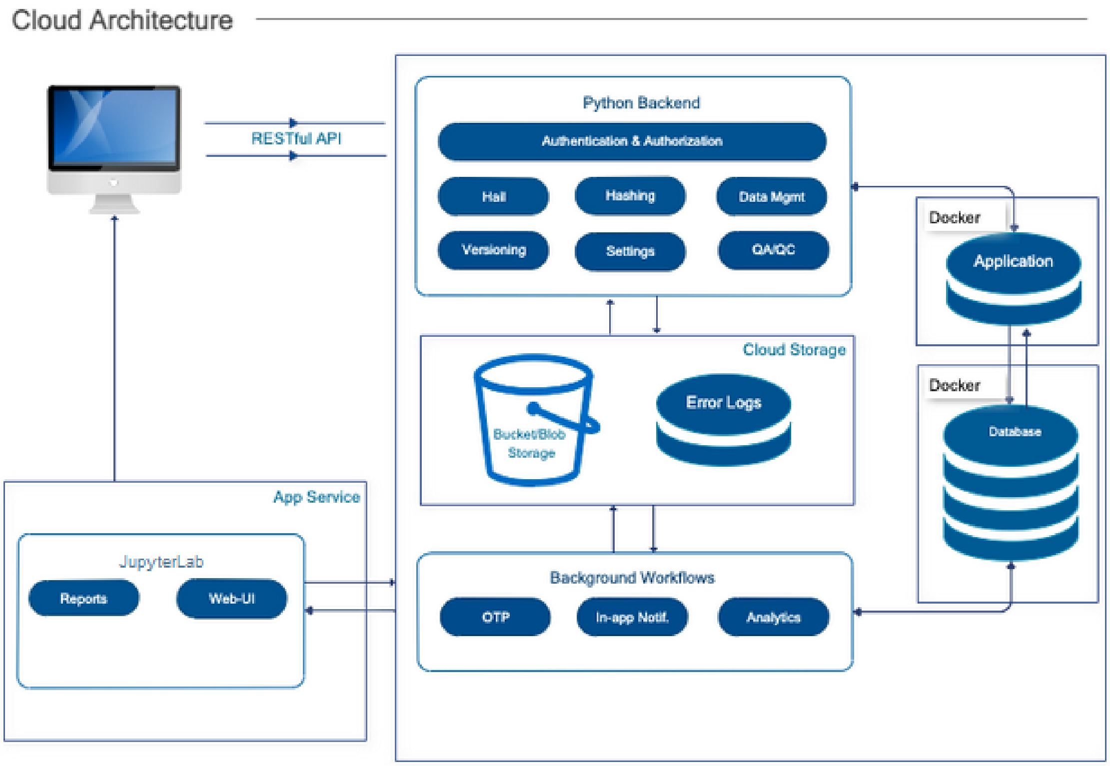


Figure 3.1. Software Schematic of UF-GWAS (software tool)

3.3.2.a Backend

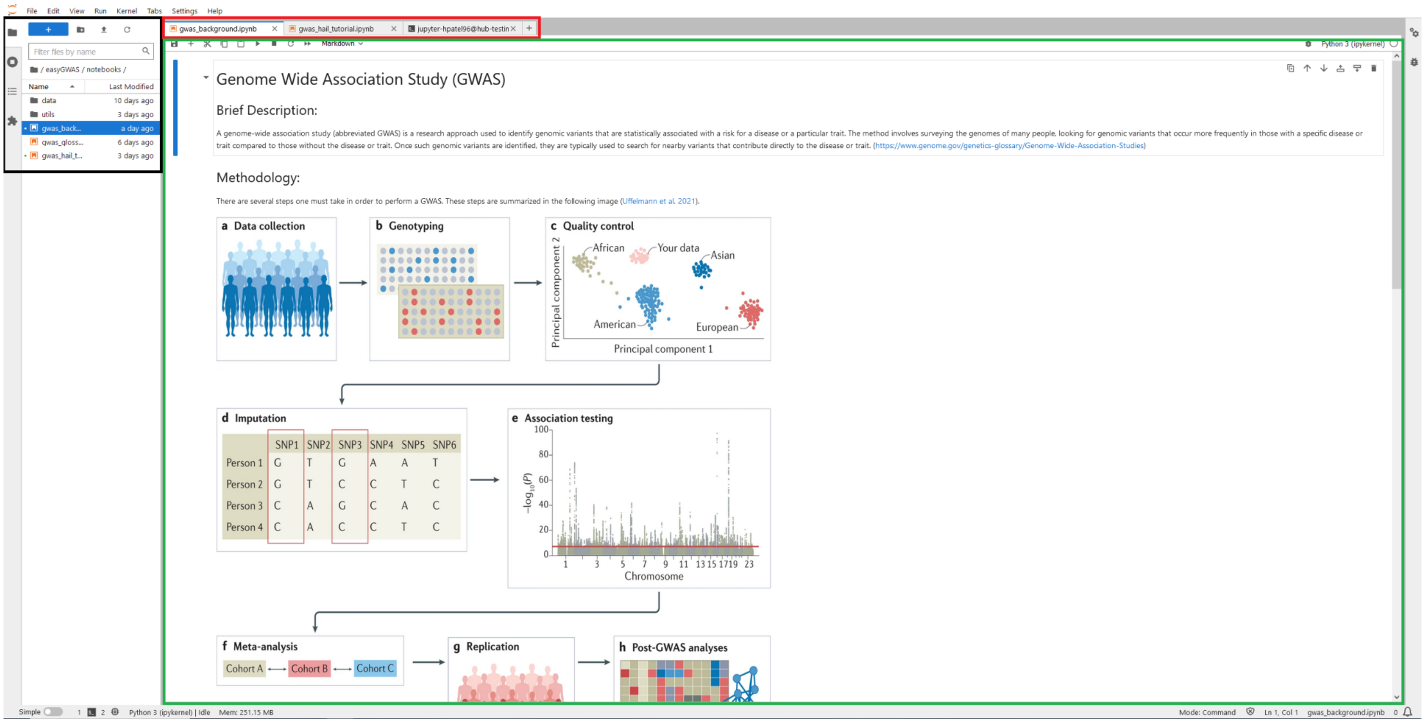
The backend of the tool consisted of a single Docker container that housed all of the necessary libraries and toolkits needed to perform a GWAS. These included Python, Hail, Jupyter, and their associated dependencies. The base Docker image was 3.09GB in size and was able to be deployed on three major cloud computing providers: Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure (Azure). The minimal required configurations to host UF-GWAS on these cloud providers varied based on the available machines sold by each provider. For AWS, a “t2.medium” EC2 instance was required that had 2 virtual CPUs, 4GB of RAM memory, and cost roughly $38.00 per month. For GCP, a “e1-medium” instance was required that had 2 virtual CPUs, 4GB of RAM memory, and cost roughly $31.00 per month. For Azure, a “B2ms” instance was retired that had 2 CPU cores, 8GB of RAM memory, and cost roughly $48.00 per month. These were baseline instances required to host just the software tool itself, not any genetic data or compute power required to run large scale analyses. Cost for additional storage (priced by GB of storage needed) varied by cloud provider and by storage location. The following Github repository hosts all the code used to create UF-GWAS: https://github.com/hvpatel96/UF-GWAS.

3.3.2.b Frontend

The frontend of the tool was created using JupyterHub. JupyterHub handled all of the authentication of users and redirected successful logins to a JupyterLab server that hosted the core UF-GWAS tool. To allow users to connect to the tool easily, a domain name and DNS server were purchased, costing $18 per year.

3.3.2.c Contents of UF-GWAS

UF-GWAS consisted mainly of customized Jupyter Notebooks. These notebooks included background information, template code, and a fully-fledged GWAS tutorial that walked novice researchers through all stages of conducting a GWAS analysis (loading data, quality control, phenotyping, merging genetic and phenotype data, controlling for ancestry, statistical testing, and visualization/storage of results). Additionally, supplementary material was provided that included a tutorial on Python/Jupyter, a tutorial on Hail, hyperlinks to peer-reviewed publications/reference material, glossary of terms used throughout the tool, and two publicly available datasets (1000 Genomes and Human Genome Diversity Project). The figures below showcase the various core aspects of UF-GWAS:



Genome Wide Association Study (GWAS)

Brief Description:

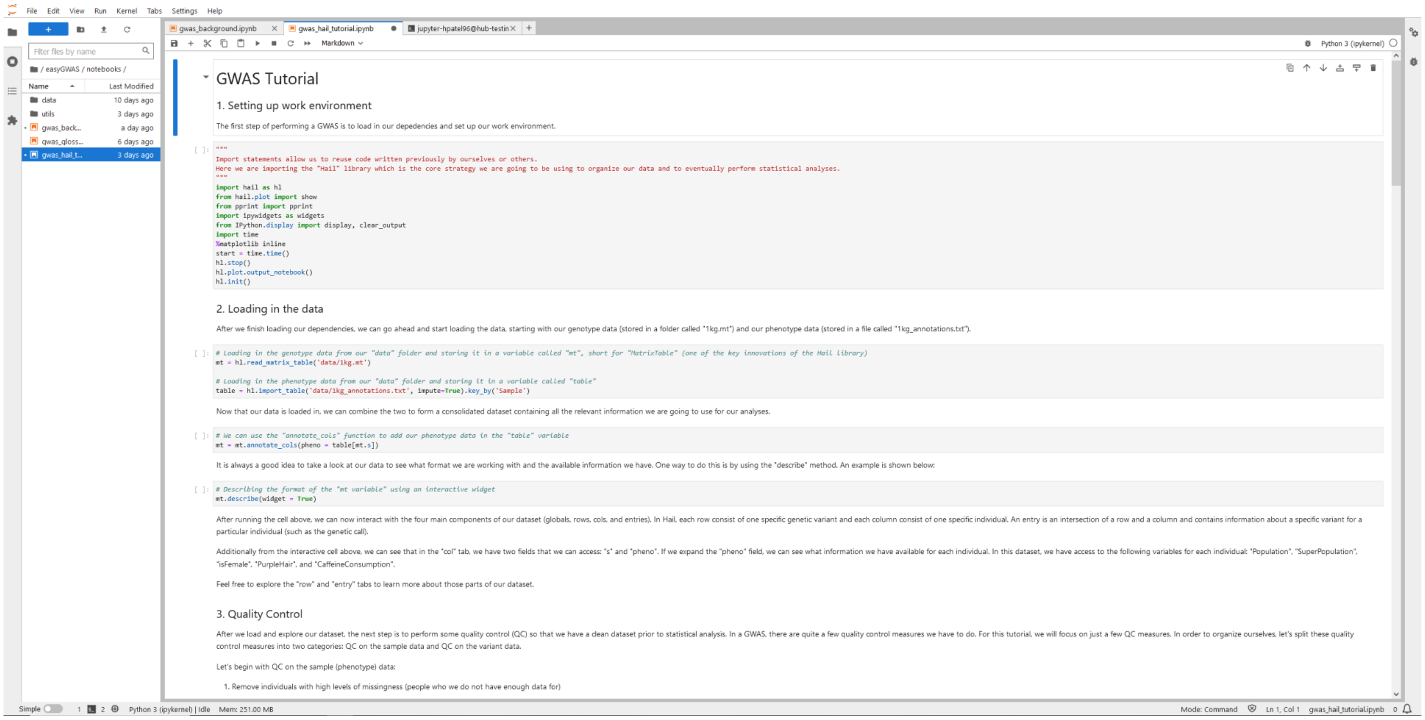
A genome-wide association study (abbreviated GWAS) is a research approach used to identify genomic variants that are statistically associated with a risk for a disease or a particular trait. The method involves surveying the genomes of many people, looking for genomic variants that occur more frequently in those with a specific disease or trait compared to those without the disease or trait. Once such genomic variants are identified, they are typically used to search for nearby variants that contribute directly to the disease or trait. (https://www.genome.gov/genetics-glossary/Genome-Wide-Association-Studies)

Methodology:

There are several steps one must take in order to perform a GWAS. These steps are summarized in the following image (Uffelmann et al. 2021).

	SNP1	SNP2	SNP3	SNP4	SNP5	SNP6
Person 1	G	T	G	A	A	T
Person 2	G	T	C	C	T	C
Person 3	C	A	G	C	A	C
Person 4	C	A	C	C	T	C

Figure 3.2. Screenshot of UF-GWAS landing page. File navigation tree is shown in the black rectangle. Ability to have multiple windows (including analyses, supporting files, and terminal window) highlighted by red rectangle. Main workspace highlighted by green rectangle.



GWAS Tutorial

1. Setting up work environment

The first step of performing a GWAS is to load in our depedencies and set up our work environment.

```
"""
Import statements allow us to reuse code written previously by ourselves or others.
Here we are importing the "Hail" library which is the core strategy we are going to be using to organize our data and to eventually perform statistical analyses.
"""
import hail as hl
from hail.plot import show
from pprint import pprint
import ipywidgets as widgets
from IPython.display import display, clear_output
import time
%matplotlib inline
start = time.time()
hl.stop()
hl.plot.output_notebook()
hl.init()
```

2. Loading in the data

After we finish loading our dependencies, we can go ahead and start loading the data, starting with our genotype data (stored in a folder called "1kg.mt") and our phenotype data (stored in a file called "1kg_annotations.txt").

```
# Loading in the genotype data from our "data" folder and storing it in a variable called "mt", short for "MatrixTable" (one of the key innovations of the Hail library)
mt = hl.read_matrix_table('data/1kg.mt')

# Loading in the phenotype data from our "data" folder and storing it in a variable called "table"
table = hl.import_table('data/1kg_annotations.txt', impute=True).key_by('Sample')
```

Now that our data is loaded in, we can combine the two to form a consolidated dataset containing all the relevant information we are going to use for our analyses.

```
# We can use the "annotate_cols" function to add our phenotype data in the "table" variable
mt = mt.annotate_cols(pheno = table[mt.s])
```

It is always a good idea to take a look at our data to see what format we are working with and the available information we have. One way to do this is by using the "describe" method. An example is shown below:

```
# Describing the format of the "mt variable" using an interactive widget
mt.describe(widget = True)
```

After running the cell above, we can now interact with the four main components of our dataset (globals, rows, cols, and entries). In Hail, each row consist of one specific genetic variant and each column consist of one specific individual. An entry is an intersection of a row and a column and contains information about a specific variant for a particular individual (such as the genetic call).

Additionally from the interactive cell above, we can see that in the "col" tab, we have two fields that we can access: "s" and "pheno". If we expand the "pheno" field, we can see what information we have available for each individual. In this dataset, we have access to the following variables for each individual: "Population", "SuperPopulation", "isFemale", "PurpleHair", and "CaffeineConsumption".

Feel free to explore the "row" and "entry" tabs to learn more about those parts of our dataset.

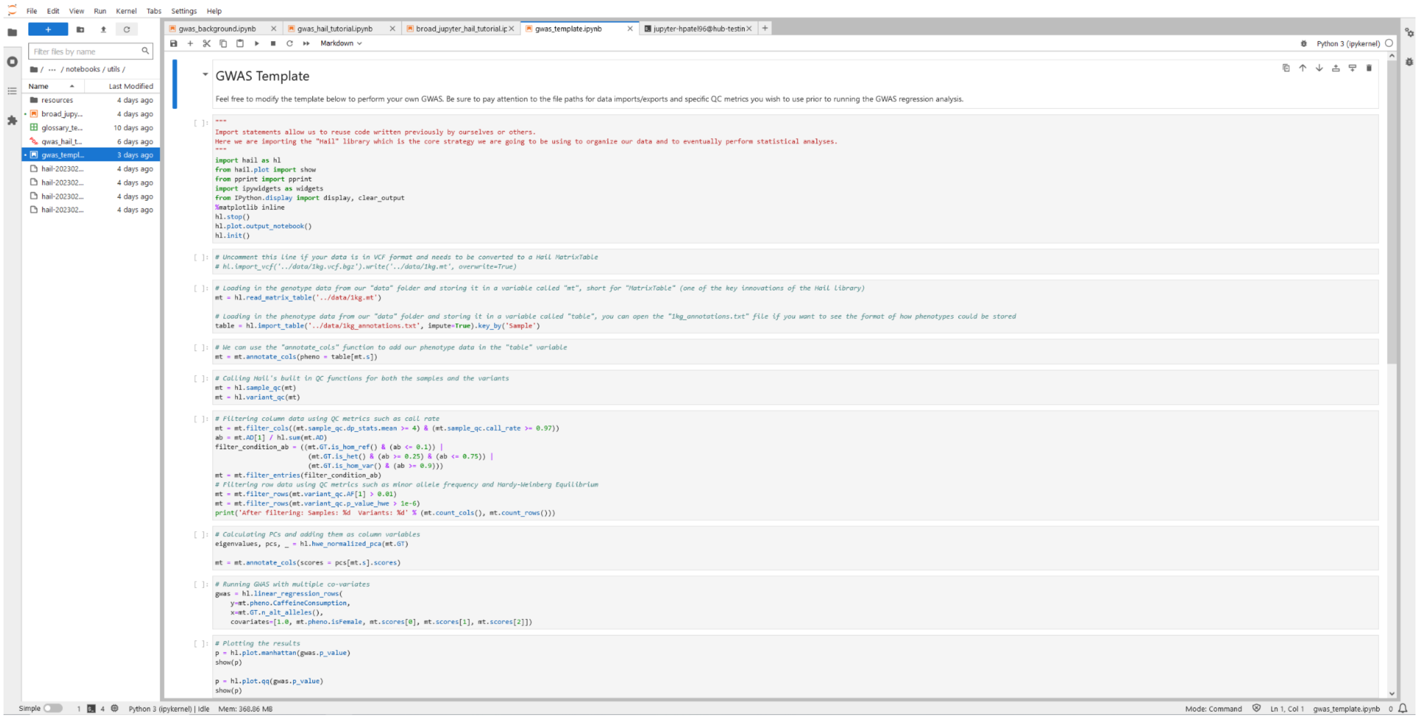
3. Quality Control

After we load and explore our dataset, the next step is to perform some quality control (QC) so that we have a clean dataset prior to statistical analysis. In a GWAS, there are quite a few quality control measures we have to do. For this tutorial, we will focus on just a few QC measures. In order to organize ourselves, let's split these quality control measures into two categories: QC on the sample data and QC on the variant data.

Let's begin with QC on the sample (phenotype) data:

1. Remove individuals with high levels of missingness (people who we do not have enough data for)

Figure 3.3. Screenshot of GWAS tutorial JupyterNotebook that allows users to complete a full GWAS (data import, adding phenotype information, quality control, correction for ancestry, regression analysis, and visualization) using the provided 1000 Genomes dataset.



GWAS Template

Feel free to modify the template below to perform your own GWAS. Be sure to pay attention to the file paths for data imports/exports and specific QC metrics you wish to use prior to running the GWAS regression analysis.

```
"""
Import statements allow us to reuse code written previously by ourselves or others.
Here we are importing the "Hail" library which is the core strategy we are going to be using to organize our data and to eventually perform statistical analyses.
"""
import hail as hl
from hail.plot import show
from pprint import pprint
import ipywidgets as widgets
from IPython.display import display, clear_output
%matplotlib inline
hl.stop()
hl.plot.output_notebook()
hl.init()
```

```
# Uncomment this line if your data is in VCF format and needs to be converted to a Hail MatrixTable
# hl.import_vcf('../data/1kg.vcf.bgz').write('../data/1kg.mt', overwrite=True)
```

```
# Loading in the genotype data from our "data" folder and storing it in a variable called "mt", short for "MatrixTable" (one of the key innovations of the Hail library)
mt = hl.read_matrix_table('../data/1kg.mt')

# Loading in the phenotype data from our "data" folder and storing it in a variable called "table", you can open the "1kg_annotations.txt" file if you want to see the format of how phenotypes could be stored
table = hl.import_table('../data/1kg_annotations.txt', impute=True).key_by('Sample')
```

```
# We can use the "annotate_cols" function to add our phenotype data in the "table" variable
mt = mt.annotate_cols(pheno = table[mt.s])
```

```
# Calling Hail's built in QC functions for both the samples and the variants
mt = hl.sample_qc(mt)
mt = hl.variant_qc(mt)
```

```
# Filtering column data using QC metrics such as call rate
mt = mt.filter_cols((mt.sample_qc.dp_stats.mean >= 4) & (mt.sample_qc.call_rate >= 0.97))
ab = mt.AD[1] / hl.sum(mt.AD)
filter_condition_ab = ((mt.GT.is_hom_ref() & (ab <= 0.1)) |
                       (mt.GT.is_het() & (ab >= 0.25) & (ab <= 0.75)) |
                       (mt.GT.is_hom_var() & (ab >= 0.9)))
mt = mt.filter_entries(filter_condition_ab)
# Filtering row data using QC metrics such as minor allele frequency and Hardy-Weinberg Equilibrium
mt = mt.filter_rows(mt.variant_qc.AF[1] > 0.01)
mt = mt.filter_rows(mt.variant_qc.p_value_hwe > 1e-6)
print('After filtering: Samples: %d  Variants: %d' % (mt.count_cols(), mt.count_rows()))
```

```
# Calculating PCs and adding them as column variables
eigenvalues, pcs, _ = hl.hwe_normalized_pca(mt.GT)

mt = mt.annotate_cols(scores = pcs[mt.s].scores)
```

```
# Running GWAS with multiple co-variates
gwas = hl.linear_regression_rows(
    y=mt.pheno.CaffeineConsumption,
    x=mt.GT.n_alt_alleles(),
    covariates=[1.0, mt.pheno.isFemale, mt.scores[0], mt.scores[1], mt.scores[2]])
```

```
# Plotting the results
p = hl.plot.manhattan(gwas.p_value)
show(p)

p = hl.plot.qq(gwas.p_value)
show(p)
```

Figure 3.4. Screenshot of GWAS template code that users can modify to input their own dataset/phenotypes

3.3.3 *Clostridioides difficile* GWAS

UF-GWAS was used to conduct a GWAS on the *Clostridioides difficile* phenotype, specifically in the European ancestry subgroup of the eMERGE III dataset. **Table 2** below describes summary statistics of the overall study sample after filtering for the inclusion and exclusion criteria defined by Ferar et al [100].

N	**Cases (n=879)**	**Controls (n=13,359)**	**Total (n=14, 238)**
Sex (female) **Median BMI (kg/m^2)** **Median Age**	51% (454) 25.8 62	52% (6994) 27.9 65.6	52% (7448) 27.7 65.4
Antibiotic exposure (Within	30% (260)	18% (2377)	19% (2637)

7-62 days prior to index date) High risk Moderate risk Low risk No exposure	10% (92) 2% (21) 58% (506)	19% (2552) 2% (215) 61% (8217)	19% (2644) 2% (236) 60% (8723)
Cancer (First record to index date + 7 days) **Chemotherapy** (Before 180 days prior to index date, after 7 days following index date) **Diabetes Mellitus** (Ever) **HIV** (Ever) **Nursing Home Status** (Within 90 days prior to index date) **Corticosteroid medications** (Within 21 days prior to index date) **Transplant medications** (First record to index date + 7 days)	24% (207) 20% (179) 22% (194) 0.9% (8) 13% (118) 18% (162) 17% (147)	15% (2002) 13% (1754) 22% (2968) 0.5% (60) 2% (303) 9% (1255) 6% (778)	16% (2209) 14% (1933) 22% (3162) 0.4% (68) 3% (421) 10% (1417) 7% (935)

Table 3.2. Summary statistics of the study sample used in the *Clostridioides difficile* GWAS

The *Clostridioides difficile* GWAS was able to be completed using UF-GWAS on average in 1hr 49min, costing roughly $17.37 in compute cost. The Manhattan plot showcasing the significant variants at a threshold of 5x10^{-8} is shown below in **Figure 3.5**.

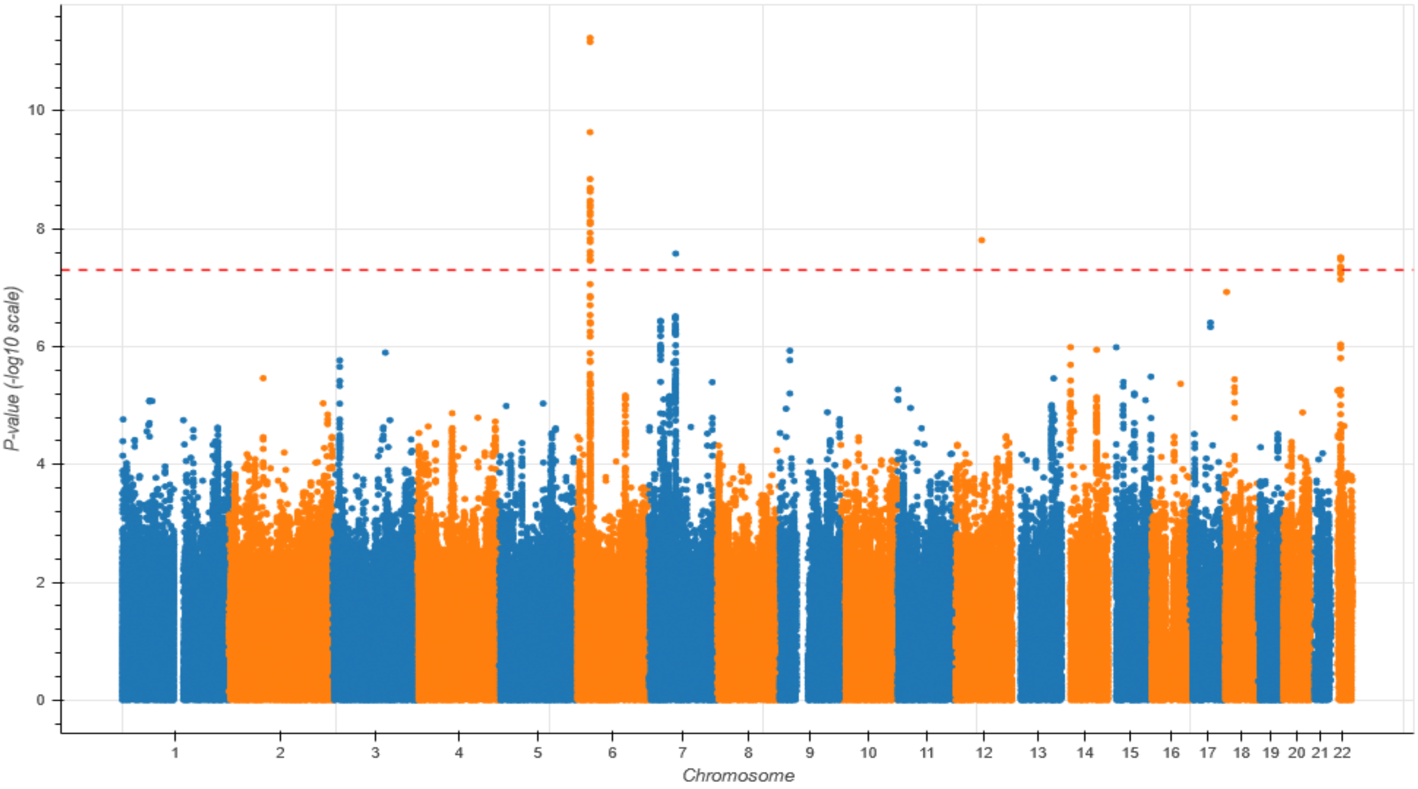
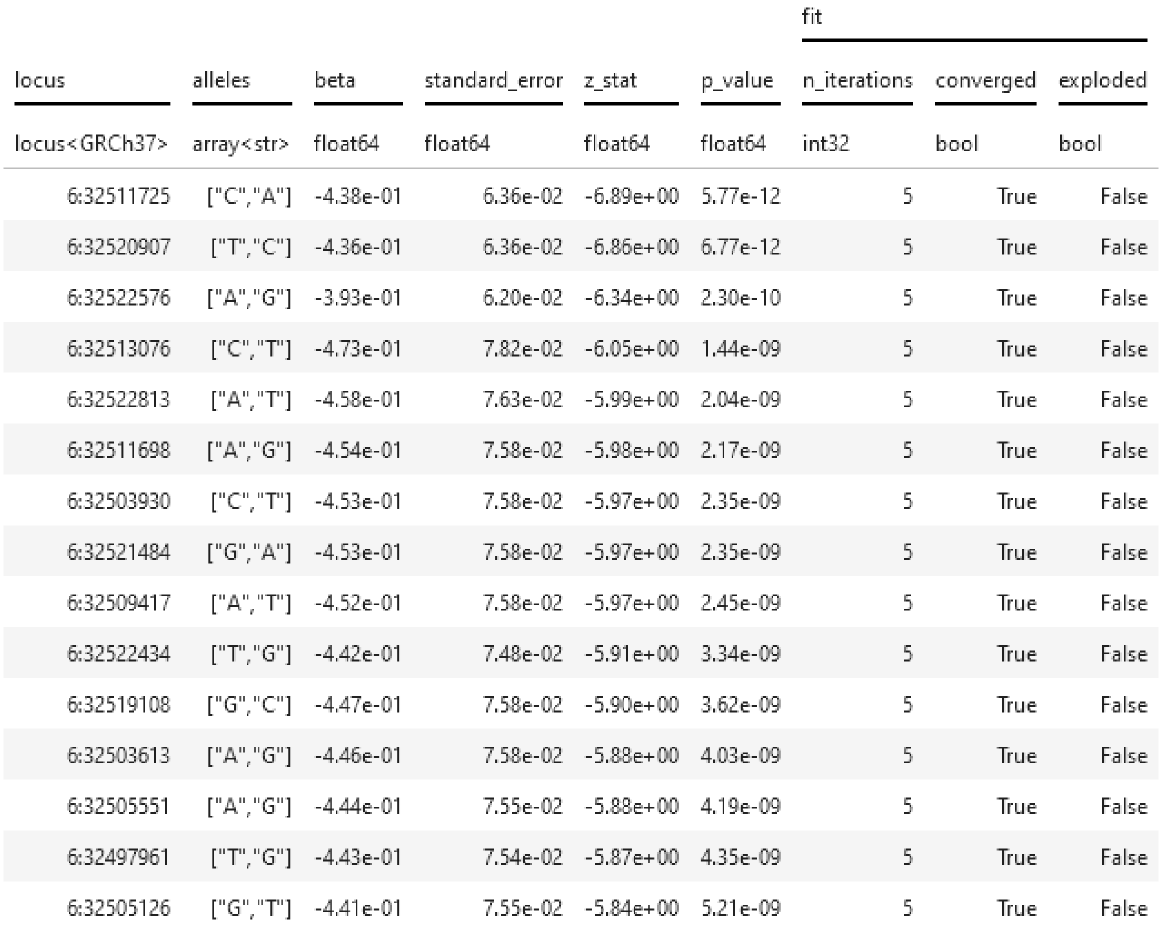


Figure 3.5. Manhattan plot of *Clostridioides difficile* GWAS

Below are the top 15 variants sorted by p-value (**Figure 3.6**). All of the top variants are on chromosome 6, and the top 5 most significant variants (rs68148149, P=5.77 x 10^{-12} ; rs3828840, P=6.77 x 10^{-12} ; rs35882239, P=2.30 x 10^{-10} ; rs35882239, P=1.44 x 10^{-9} ; rs35222480, P=2.04 x 10^{-9}) map to the intergenic region between the HLA-DRB5 and HLA-DRB1 genes in the beta block of the MHC Class II region. These findings are consistent with those of Ferar et al.'s *Clostridioides difficile* GWAS.



						fit		
locus	alleles	beta	standard_error	z_stat	p_value	n_iterations	converged	exploded
locus<GRCh37>	array<str>	float64	float64	float64	float64	int32	bool	bool
6:32511725	["C","A"]	-4.38e-01	6.36e-02	-6.89e+00	5.77e-12	5	True	False
6:32520907	["T","C"]	-4.36e-01	6.36e-02	-6.86e+00	6.77e-12	5	True	False
6:32522576	["A","G"]	-3.93e-01	6.20e-02	-6.34e+00	2.30e-10	5	True	False
6:32513076	["C","T"]	-4.73e-01	7.82e-02	-6.05e+00	1.44e-09	5	True	False
6:32522813	["A","T"]	-4.58e-01	7.63e-02	-5.99e+00	2.04e-09	5	True	False
6:32511698	["A","G"]	-4.54e-01	7.58e-02	-5.98e+00	2.17e-09	5	True	False
6:32503930	["C","T"]	-4.53e-01	7.58e-02	-5.97e+00	2.35e-09	5	True	False
6:32521484	["G","A"]	-4.53e-01	7.58e-02	-5.97e+00	2.35e-09	5	True	False
6:32509417	["A","T"]	-4.52e-01	7.58e-02	-5.97e+00	2.45e-09	5	True	False
6:32522434	["T","G"]	-4.42e-01	7.48e-02	-5.91e+00	3.34e-09	5	True	False
6:32519108	["G","C"]	-4.47e-01	7.58e-02	-5.90e+00	3.62e-09	5	True	False
6:32503613	["A","G"]	-4.46e-01	7.58e-02	-5.88e+00	4.03e-09	5	True	False
6:32505551	["A","G"]	-4.44e-01	7.55e-02	-5.88e+00	4.19e-09	5	True	False
6:32497961	["T","G"]	-4.43e-01	7.54e-02	-5.87e+00	4.35e-09	5	True	False
6:32505126	["G","T"]	-4.41e-01	7.55e-02	-5.84e+00	5.21e-09	5	True	False

Figure 3.6. Top 15 SNPs sorted by p-value from the *Clostridioides difficile* GWAS

3.4 Discussion

Overall, this aim utilized knowledge gained from the user needs evaluation from Aim 1 to create user personas in order to capture key characteristics of the different types of novice genomics researchers. Those personas along with existing best practice guidelines on software development were utilized to create UF-GWAS (a cloud-agnostic user-friendly GWAS tool). Although many GWAS tools exist in the bioinformatics landscape, very few utilize user-centered design throughout design and deployment. In the following sections, we will explore key insights gained from persona development and throughout UF-GWAS's design and testing.

3.4.1 Personas

Novice genomics researchers are a diverse group of individuals with a wide variety of both bioinformatics and computational experiences. In order to capture key characteristics across sub-groups, we consolidated the results from Aim 1 to create three distinct personas named Orlando (limited domain knowledge but computational familiarity), Irene (some domain knowledge and computational familiarity), and Scott (domain experience but lacks computational familiarity). These personas offered insight in how users interact with bioinformatics software when first entering the scene and the various challenges they face. Studies have shown that utilizing personas offer both researchers and care providers different lenses from which to approach complex challenges such as training, clinical testing, and return of results [79, 101]. As indicated by LeRouge et al., "user profiles and personas go well beyond demographics, as they attempt to 'capture' the user's mental model comprising of their expectations, prior experience and anticipated behavior … how they think, feel, and behave." [78]. By capturing the user's mental model, developers can actively associate features being developed with challenges that the users will encounter upon initial use and sustained adoption. Throughout this aim, the three developed personas played a pivotal role in determining the accessibility of UF-GWAS, including the scope of the provided background information and the extent to which the tool aided users in conducting a GWAS. Furthermore, the personas created are not limited to just GWAS analyses as they represent challenges that NGRs face across bioinformatics. As discussed in Aim 1, the current landscape of informatics tools lacks proper foundational support for novice users and these personas can easily be adapted for other types of analyses.

3.4.2 UF-GWAS

3.4.2.a Overall design

The design of UF-GWAS addresses many of the concerns voiced by the NGRs interviewed in Aim 1 and also aligns with the NHGRI's 2020 strategic vision of building strong informatics foundations in order to break barriers and allow for robust genomics analyses [57]. One of the most commonly experienced roadblocks faced by new users is initial setup and installation of a software tool. UF-GWAS, using Docker containers, circumvents this problem by packaging all required dependencies and setup instructions into a single bundle that can be deployed with ease across most cloud computing providers (Amazon Web Services, Google Cloud Platform, and Microsoft Azure). Furthermore, UF-GWAS strikes a balance between pure shell scripting and restricted user-interfaces by having a JupyterLab frontend that allows terminal access for seasoned programmers and guided JupyterNotebooks for more novice users.
UF-GWAS also incorporates many of the best practices in the development of bioinformatics software, as suggested by Veiga Leprevost et al., which include: source-code availability, software indexing and proper documentation, test code and sample dataset availability, tutorial content, and also ease of setup/access [102]. Studies have shown that by incorporating those key components listed above, software systems tend to have higher adoption rates and a more sustained user base over time [9].

3.4.2.b Hail

The decision to utilize Hail as the primary backend Python library for data organization and statistical analysis was made primarily to fully utilize the parallelization capabilities of cloud computing. Depending on the configuration utilized during initial setup of UF-GWAS, Hail dynamically allocates available computational resources during all stages of the GWAS (data loading, quality control, ancestry correction, statistical testing, and visualization of results). Furthermore, the default data storage schema of the MatrixTable allows for efficient execution of experiments, as it is able to support random data access compared to the traditional Variant Call Format (VCF) file format commonly used across the genomics space. Hail's built in error messages and progress indicators were also useful in creating a more user-friendly analysis interface.

3.4.2.c *Clostridioides difficile* GWAS

In order to test the efficiency and accuracy of UF-GWAS, we conducted a GWAS analysis on the *Clostridioides difficile* phenotype. UF-GWAS was able to replicate the results in a fraction of the time, only 1 hour and 49 minutes, and cost roughly $17 to complete. As cloud computing costs are often ambiguous, this finding provides some perspective in the amount of computational resources needed to conduct a full-scale analysis. Existing literature has shown costs of storage and computational costs per hour, but have lacked in showing the cost of practical experiments [103]. This study along with the work done by Krissaane et al. offer much needed insight into hidden costs of cloud research [91]. Additionally, as Hail scaled fairly linearly with the amount of samples and variants, it is reasonable to utilize the cost it took to run the *Clostridioides difficile* GWAS to estimate the cost it would take to run other experiments if using the same computational resource group.

3.4.3 Limitations and Suggested Improvements

Although the personas created were reviewed by a qualitative expert, the primary investigator may have introduced bias based on their experience during the persona creation process. As a result, the personas that were created represent one interpretation of how the data from Aim 1 could be used to represent theoretical novice users. For future studies, further persona refinement through multiple investigators or via focus groups can be done in order to better represent the target user base.

UF-GWAS was developed mainly by the primary investigator which as a result introduces bias that stems from design and development experience. Although design input was taken from the knowledge gained via the needs assessment, final decisions on the frontend and backend components were made by one individual. However, the primary investigator did receive help from more experienced programmers to troubleshoot certain aspects of development and that did add more perspectives to the final product. In the future, multiple developers would work on a project simultaneously in order to incorporate diverse design and development viewpoints into the creation of the final product. Additionally, UF-GWAS in its current form is limited in the types of analyses it can do, only GWAS and some quality control, and how many people can work on an analysis together. This can be expanded in future iterations of the tool to support more analyses and even scaled to offer built-in collaborative coding. Lastly, due to limited time and budget, the *Clostridioides difficile* GWAS was only run on UF-GWAS running on Google Cloud Platform. As a result, the cost and runtime estimates may not be truly generalizable to other cloud providers. Future work would replicate this GWAS on Amazon Web Services and Microsoft Azure to test for cost and runtime consistencies.

3.5 Conclusion

In this aim, we created a user-friendly genome wide association study software tool (UF-GWAS) informed by the needs assessment conducted in Aim 1 and three novice user personas. UF-GWAS is a cloud-agnostic tool that is containerized using Docker containers, offering easy setup and deployment across all three major cloud providers (AWS, GCP, Azure). It also includes an interactable frontend using JupyterLab and utilizes the Hail Python analysis library to provide efficient data quality control and statistical analyses. Experiments are easily reproducible and shareable through JupyterNotebooks and the entire codebase is available on Github. There exists a gap in proper user-centered design in bioinformatics tools which often leads to inability for initial adoption and lack of sustained use by NGRs for most existing software tools in the genomics landscape. UF-GWAS aims to help lessen that gap by specifically eliminating many common challenges faced by NGRs in order to help ease their transition into the bioinformatics space. In the next chapter, we will explore the usability of UF-GWAS by performing user evaluations and we will utilize that knowledge to provide a set of design and development guidelines for future bioinformatics tool development.

CHAPTER 4: EVALUATION OF UF-GWAS AND GUIDELINES FOR FUTURE DEVELOPMENT OF BIOINFORMATICS SOFTWARE (AIM 3)

Abstract

User evaluation of bioinformatics software is scarce and limited, often leading to a frustrated user base, unexpected variability in results, and eventually abandonment of published software. Although most new bioinformatics tools claim to provide some sort of dedicated user support, most do not utilize user-centered design (UCD) during the tool's design and development and only offer barebone documentation without considering the needs of diverse users. Novice genomics researchers (NGRs) are keystone users of such new informatics software that are often overlooked and continue to face usability barriers despite the push by large organizations, such as the National Human Genome Research Institute (NHGRI), for better analysis foundations and a more diverse workforce. UF-GWAS is a cloud-agnostic, user-friendly GWAS software tool that was developed to aid NGRS by applying UCD principle via semi-structured interviews and user persona development to better assist target novice users. In this aim, we evaluated the usability of UF-GWAS by NGRs, tested UF-GWAS against industry standard tools, and created a set of design guidelines that future researchers can utilize to create more usable bioinformatics software. Users who evaluated UF-GWAS reported the tool to be highly usable across two standard usability instruments, the System Usability Scale and the Post-Study System Usability Questionnaire. The key components of UF-GWAS that users appreciated include: ease of onboarding, quick access to relevant background knowledge, shareability and reproducibility of results, and clarity of the documentation. UF-GWAS performed comparably to industry standard software in terms of computational cost and time taken to complete analyses. Lastly, the set of guidelines generated encompassed a wide variety of

knowledge sources and act as effective scaffolding for future researchers to expand and apply to fit their own needs. As NGRs continue to explore bioinformatics, it is important for researchers to acknowledge their unique challenges and create more usable tools. By doing so, we can build stronger foundations and help build a better future for informatics research.

4.1 Introduction

Bioinformatics analysis tools and packages are one of the most rapidly growing types of published software in the informatics landscape today. From genome browsers to sequence aligners and expression profilers to risk predictors, the sheer breadth of topics being analyzed by modern day tools is massive [104]. These tools have not only changed how analyses are done in siloed laboratories, but have changed how bioinformatics is conducted across the world. Wren et al.'s review on highly cited publications found that papers that included some type of analysis software were 31-fold over-represented amongst the highest impact bioinformatics papers of the past two decades [105]. Couple this with the dropping costs of computation and storage and it is easy to see why there has been a sudden boom in the development and publication of bioinformatics software. However, there continues to be a lack of proper evaluation and quality control of these tools prior to deployment and adoption by the informatics community [106]. With respect to standards for evaluation, the International Organization of Standardization (ISO) has already provided guidelines on both ergonomic human-system interaction, ISO 9241, and software product quality requirement and evaluation, ISO 25062 [107, 108]. Other disciplines of study such as mobile application development by large software corporations such as Google and Facebook regularly utilize these guidelines during development and evaluation to reduce usability barriers and push towards sustained usage of their products [109]. However, in bioinformatics, these usability standards are not regularly considered and enforced, leading to frustrated users, poor quality results, and ultimately abandoned software [9, 13].

Albeit scarce, some prior evaluation work of bioinformatics software has been conducted and then subsequently utilized to create more usable analysis tools and information dashboards. In 2022, Bauer-Negrini et al. developed a usability heuristic-based framework for the evaluation of circRNA identification tools [110]. Additionally, Shyr et al. provided some lessons learned and evaluation guidelines when conducting usability testing in clinical exome analysis software [111]. Furthermore, Duan et al. conducted a systematic evaluation of the accuracy of bioinformatics tools for identification of long noncoding RNAs which uncovered flaws related to input data sensitivity and lack of robustness in commonly used software [112]. However, all of the evaluations above either focused on usability of the overall system or the accuracy of the results generated. Thorough evaluation, according to the ISO standards mentioned above, must approach the assessment of a system from both perspectives. Without doing so, we are only able to get a partial understanding of the barriers users face and will fail to provide adequate solutions to aid them in overcoming usability challenges.

Another underexplored perspective of proper evaluation is the creation of context-specific guidelines for the development of analytical software in the future and how these software can be best utilized to support the target users. This is critical because without proper translation of knowledge gained from evaluation into concrete guidelines, it becomes difficult for future developers to quickly assess whether or not their design choices are appropriate for different contexts. Blanket usability guidelines, such as the ones that exist currently in the literature, offer barebone support but lack the background needed to best serve a target user base [102, 113]. Some prior work has been done to characterize the core competencies needed of bioinformaticians, such as the need for strong statistical and biological foundations suggested by Brazas et al. or Tractenberg et al. 's informatics mastery rubric [47, 114]. Additionally, Sayres et al. even outline a set of training objectives for undergraduates interested in bioinformatics [115]. However, all of these competencies can only be achieved with strong foundational support from both the literature and available bioinformatics software tools. Friedman et al's “fundamental theorem” of biomedical informatics postulates that “a person working in partnership with an information source is ‘better’ than the same person unassisted” [116]. In today’s data driven world, that “information source” is often some type of program or analytical software. By improving the information source, we can create strong supportive tools to help researchers best approach complex problems. The NHGRI 2020 strategic vision calls for the creation of a more diverse workforce, but this vision is not achievable without robust software built using user-centered design and thorough evaluation specifically within the context of novice users [117].

To evaluate the usability of UF-GWAS in the context of NGRs and to provide a set of guidelines for future software development tailored towards novice users, in this aim, we will employ a mixed-methods study design consisting of two parts: a task-based usability evaluation of UF-GWAS by target users and the application of the evaluation rubric developed in Aim 1 on both UF-GWAS and PLINK (a widely-used GWAS analysis software). First, a prospective usability analysis will be conducted where novice researchers will be asked to perform a GWAS experiment and associated tasks utilizing UF-GWAS and be given an opportunity to provide feedback on its usability along with any challenges that arose during the evaluation period.

Afterwards, a prospective comparison between UF-GWAS and PLINK will be made utilizing the evaluation rubric from Aim 1. The primary goal of this aim will be to assess the usability of UF-GWAS and to provide a list of guidelines for future bioinformatics software development. We hypothesize that the NGRs will have positive reviews of UF-GWAS, due to the incorporation of many user-centered design choices in Aim 2, and that UF-GWAS will outperform PLINK in the lenses of usability via the evaluation framework.

4.2 Methods

4.2.1 User Evaluations

4.2.1.a Participant Recruitment

Prior to recruiting participants, an Institutional Review Board (IRB) application for this study was submitted to the University of Washington's Human Subjects Division on January 12th, 2023. This application was processed and approved on January 13th, 2022 with exempt status. After obtaining approval from the IRB, a target sample of 10 study participants were recruited for interviews. Although existing usability studies report a sample size of 5-6 participants is adequate to uncover most user concerns, recent studies in usability sample size report better saturation between 10-12 participants [118–120]. The inclusion criteria for study participants were as follows:

- Graduate student currently belonging to a lab at the University of Washington and Tulane University
- Prior experience with bioinformatics analysis methodologies
- Prior experience with informatics tools and software
- Less than 3 years of experience with genomic analyses

In order to recruit participants, an initial list of potential candidates was compiled using a contact list provided by the doctoral committee chair (sharing contacts across UW and Tulane), the public UW Biomedical and Health Informatics graduate student directory, and public UW and Tulane faculty directories whose labs conducted research involving bioinformatics and genomics. The final potential interview list consisted of 18 potential interviewees: 12 from the University of Washington and 8 from Tulane University. This list was stored in a password-protected file and was only shared with the doctoral committee members. Beginning in February 2023, potential interview candidates were contacted via email to participate in the evaluation study, starting with the UW candidates and followed by Tulane candidates. By March 2022, all 18 candidates were invited and out of those 18 candidates, 10 agreed to participate in the study.

4.2.1.b Evaluation Study

The primary investigator developed a set of evaluation instructions that would be given to each participant during their 60 minute evaluation period. These instructions consisted of a set of tasks that each user would have to complete, primarily consisting of running a sample GWAS using the 1000 Genomes dataset [121]. These tasks were to be completed using UF-GWAS (the user-friendly GWAS tool developed in Aim 2) and were reviewed by both the doctoral committee and a qualitative expert. After incorporating feedback, a final instruction set was created and can be found in **Appendix E**.

Each evaluation was scheduled for 60 minutes, which included time for introductions, background on the purpose of the study, informed consent, and the evaluation itself. Consent was acquired verbally utilizing a consent guide approved by the IRB. A task-based evaluation methodology was utilized to assess the usability of UF-GWAS. Shah and White et al. have shown that task-based evaluations have been well researched in the past and are robust in uncovering user challenges that may arise during all stages of a tool's lifecycle (adoption and setup, first use, and sustained usage) [122]. As the primary goal of the user evaluation was to elucidate the challenges faced by novice genomics researchers when using UF-GWAS, this methodology was effective in reaching that goal.

After each evaluation, participants were asked to complete a short demographic questionnaire and two usability instruments, approved by the IRB, and this data was stored securely in a password-protected UW Google Drive. Participants also had the opportunity to provide feedback on the positives, challenges, and suggested improvements of UF-GWAS in the form of free-response questions. The usability instruments utilized were the System Usability Scale (SUS) and the Post-Study System Usability Questionnaire (PSSUQ) [123, 124]. Both instruments have long been utilized for usability evaluation in a wide variety of contexts, including "hardware, software, mobile devices, websites and applications" [125]. No audio files of the evaluation were stored and all participant demographic/survey information was de-identified prior to any analysis. Finally, the data was transferred to an encrypted device and imported into Microsoft Excel in order to generate descriptive statistics and calculate usability values following protocols given by the SUS and PSSUQ.

4.2.2 Rubric and Performance Evaluation of UF-GWAS and PLINK

In order to compare the usability of UF-GWAS against industry standard bioinformatics software, the evaluation rubric from Aim 1 was utilized to compare UF-GWAS and PLINK (version 2.0). PLINK is a free, open-source genome association analysis tool, designed to perform large-scale analyses in a computationally efficient manner [126]. Widely cited, PLINK is one of the most commonly used bioinformatics software tools and oftentimes the first exposure novice researchers have when entering the GWAS analysis space. To evaluate UF-GWAS and PLINK in terms of usability, the evaluation rubric developed during Aim 1 was applied to both tools by the primary investigator. The results were then imported into Microsoft Excel to generate descriptive statistics. To evaluate both tools in terms of efficiency, the *Clostridioides difficile* GWAS from Aim 2 was run using both UF-GWAS and PLINK, with each tool executing the GWAS 5 times each. The times to complete the analysis and the individual SNPs that met the significance threshold (5×10^{-8}) were recorded. The computational configuration utilized when running the both analyses was identical to that used in Aim 2 (164 CPU units and 616GB of RAM) and was run on the Google Cloud Platform (GCP).

4.2.3 Guideline Development

Practice guidelines are pivotal in properly laying a foundation for the successful development and evaluation of both existing and future resources. In order to summarize the feedback received from the user evaluations and the knowledge gained throughout this dissertation, the primary investigator developed a set of guidelines for user-centered design of bioinformatics tools, specifically when the target user base consists of novice researchers. Jaeschke et al.'s work on guideline development in clinical practice was adapted to fit the context of this study and Woolf et al.'s work on how to synthesize different knowledge sources was used to both evaluate the guidelines developed and to effectively communicate the derived set of recommendations [127, 128]. Additionally, subsections from the integrative review methodology updated by Whittemore et al. in 2005 were utilized for effective content extraction and conclusion drawing [129]. The set of knowledge sources utilized to inform the guideline development include: literature review from Aim 1, needs assessment from Aim 1, persona development from Aim 2, software design and deployment of UF-GWAS from Aim 2, and the user evaluations from Aim 3. The initial draft of guidelines were then reviewed by the doctoral committee and edited to include any feedback.

4.3 Results

4.3.1 Evaluation Study

Table 1 displays demographic information about the participants. In summary, most participants (N = 10) were female (60%), Asian (70%), non-Hispanic/Latino (90%), had less than 2 years of experience working with genetic data (80%), ranged in age from 21 to 30 (mean = 25) years old, and had varied education and familiarity with genetics. Participants, on average, took 15.3 minutes (min: 9.8 minutes; max: 22.1 minutes) to complete the main evaluation task (running a complete sample GWAS on the 1000Genomes dataset). All 10 participants were able to complete the main evaluation task and 9/10 (90%) participants were able to complete the "additional" testing evaluation task (modifying the Python code to change several quality control parameters and rerunning the GWAS), refer to **Appendix E** for task details.

		N
Age	Mean Median	25 (SD 3) years 26 (range 21 - 30) years
Sex	Male Female	4 (40%) 6 (60%)
Race	White African-American Asian Other	2 (20%) - 7 (70%) 1 (10%)

Ethnicity	Hispanic or Latino Non Hispanic or Latino	1 (10%) 9 (90%)
Education Level	Bachelor’s Degree Master’s Degree	4 (40%) 6 (60%)
Familiarity with genetics (out of 5)	Median	3 (range 1 - 4)
Years of experience	Less than 1 year 1 - 2 years More than 2 years	5 (50%) 3 (30%) 2 (20%)

Table 4.1. Participant Demographics from User Evaluations

Tables 2 and 3 report the SUS and PSSUQ survey results. Overall, participants reported an average SUS rating of 91.5/100 (min: 77.5, max: 97.5, closer to 100 equals higher usability), well above (p-value = 0.00195) the average SUS rating of 68/100 as reported by Brooke et al. when testing the scale across a wide variety of usability scenarios [130]. On the PSSUQ, participants reported an average PSSUQ rating of 1.51/7 (min: 1, max: 2.25, closer to 1 equals higher usability), well below (p-value = 0.00273) the average PSSUQ rating of 2.82/7 as reported by Sauro and Lewis et al. in their review of usability questionnaires [131]. A one-sample Wilcoxon Signed Rank Test was utilized to compare usability from the study and literature for both survey instruments. Refer to **Appendix F** for more detail on the responses from each participant.

	Strongly Disagree			**Strongly Agree**	
Questions	**1**	**2**	**3**	**4**	**5**
I think that I would like to use this system frequently.	0	1	0	3	6
I found the system unnecessarily complex.	9	1	0	0	0

I thought the system was easy to use.	0	0	0	5	5
I think that I would need the support of a technical person to be able to use this system.	8	2	0	0	0
I found the various functions in this system were well integrated.	0	0	1	2	7
I thought there was too much inconsistency in this system.	10	0	0	0	0
I would imagine that most people would learn to use this system very quickly.	0	0	1	3	6
I found the system very cumbersome to use.	9	1	0	0	0
I felt very confident using the system.	0	0	1	2	7
I needed to learn a lot of things before I could get going with this system.	5	4	1	0	0

Table 4.2. SUS results displaying participant counts for each question

	Strongly Agree					**Strongly Disagree**	
Questions	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Overall, I am satisfied with how easy it is to use this system.	8	1	1	0	0	0	0
It was simple to use this system.	9	1	0	0	0	0	0
I was able to complete the tasks and scenarios quickly using this system.	8	2	0	0	0	0	0
I felt comfortable using this system.	9	1	0	0	0	0	0
It was easy to learn to use this system.	9	1	0	0	0	0	0
I believe I could become productive quickly using this system.	9	1	0	0	0	0	0
The system gave error messages that clearly told me how to fix problems.	1	0	0	9	0	0	0
Whenever I made a mistake using the system, I could recover easily and quickly.	3	0	2	5	0	0	0
The information (such as online help, on-screen messages, and other documentation) provided with this system was clear.	9	0	1	0	0	0	0
It was easy to find the information I needed.	9	1	0	0	0	0	0

The information was effective in helping me complete the tasks and scenarios.	10	0	0	0	0	0	0
The organization of information on the system screens was clear.	9	0	1	0	0	0	0
The interface of this system was pleasant.	7	2	0	1	0	0	0
I liked using the interface of this system.	7	2	0	1	0	0	0
This system has all the functions and capabilities I expect it to have.	4	4	1	1	0	0	0
Overall, I am satisfied with this system.	7	3	0	0	0	0	0

Table 4.3. PSSUQ results displaying participant counts for each question

The last portion of the evaluation survey involved three free-response questions, refer to **Appendix G** for full survey questions and details. When asked "What are some of the positives you noticed when utilizing UF-GWAS?", most participants reported they appreciated the clear documentation and reference background materials, use of simple user-interface (UI) features such as sliders and buttons, overall code structure and comments, and the visual feedback given through graphics and printouts:

> *"The instructions were clear. I liked how methodological the steps were, starting from visualizing the structure and content of the data, running initial QC, preprocessing/running QC again, etc. I liked how there were visuals for most of the steps." (P1)*

> *"I found the background information and explanations for each section to be clearly written in non-jargony language (i.e. easy to understand) … I also think that using plots/graphs, and pointing out the specific parts of the plots/graphs that are most relevant*

> *to the tutorial (such as the significance threshold line, the p-values, the clustering plot from the PCA, etc.) also made the tool easier to navigate and understand." (P2)*

> *"I think the documentation on the goal of this process and also scientific background is very clear and intuitive. And for some steps I really appreciate if I made some mistakes I do not need to re-run the whole Jupyter notebook." (P5)*

When asked, "What are some of the negatives/challenges you noticed when utilizing UF-GWAS?", some participants reported potential information overload for the given evaluation time period:

> *"I think that if this tool were to be introduced to a broader audience (e.g. maybe those who don't have background with genetics research but are interested, people who have no coding experience and don't have prior experience navigating Jupyter notebooks, etc.), some users may be intimidated by the amount of text for some explanations, or the reliance on more coding-based language/terminology." (P2)*

> *"Maybe it is due to the time constraint, I was not able to read through all the information. However, the final summary in the tutorial notebook was very helpful for me to recapitulate the procedures." (P6)*

However, other participants wanted more information than what was provided:

> *“I would want a bit more context about the data, more context for certain steps (i.e. HWE).” (P1)*

> *"1. I think it would be helpful if there were more information about the functions used for e.g., conducting QC, PCA and plotting. For example, it would be helpful for me to learn what parameters the functions need, what assumptions (default values) we took etc.” (P7)*

Some participants had errors that arose due to the browser they were using to access the tool:

> *“I was using the Safari browser and I found the hover and annotation for the plot was unstable (sometimes it worked perfectly, sometimes it blinked).” (P5)*

Lastly, when asked, “Do you have any suggestions/improvements for UF-GWAS?”, most participants believed UF-GWAS was in a good state and minor improvements would have helped their personal satisfaction when using the tool:

> *“Would love to see areas where we have more UI control (for example being able to play around with the sliders for AF, HWE, and sample call rate).” (P1)*

> *“Not sure. I think it's just me but more checkpoints, like small quiz questions or interactions with users to make sure users are actually paying full attention and actually understand what he/she is running.” (P7)*

One participant did report adding more visual cues for when a code block is finished executing:

> *"If possible, I would maybe try to incorporate more visual cues to guide people through the use of the tool/tutorial. This might be in the form of more information provided in the code blocks to give users feedback on whether or not the code is running properly, etc." (P2)*

Overall, most participants enjoyed utilized UF-GWAS and saw potential of how it could help other novice researchers:

> *"This is such an awesome tool. I'm excited to see this type of work being done because documentation in the bioinformatics space can be sparse. This is great and easy to follow for beginner GWAS analysts!" (P1)*

> *"UF-GWAS seems to be very promising. I think it will be a boon to those starting out with genomic data analysis, and the good throughput will probably keep them around!" (P3)*

4.3.2 Evaluation of UF-GWAS and PLINK

Using the evaluation rubric developed in Aim 1, UF-GWAS performed exceptionally in the categories of "robustness of genetic analyses", "tutorials/documentation", "accessibility", and "UCD design considerations". In the categories of "data restrictions" and "reproducibility/interaction with other tools", UF-GWAS performed at an acceptable standard.

PLINK, on the other hand, only performed exceptionally in the “robustness of genetic analyses” category and was acceptable at all other categories of the evaluation rubric except for “UCD design considerations”, where it only met marginal expectations. Overall, UF-GWAS scored 16 (out of a possible 18) points receiving a grade of “EXCEPTIONAL” while PLINK scored 12 points receiving a grade of “ACCEPTABLE”. **Appendix H** breaks down the grading criteria for both UF-GWAS and PLINK.

The *Clostridioides difficile* GWAS conducted in Aim 2 was replicated using PLINK. As PLINK accepts many file formats, native BED/BIM/FAM and VCF formats for the data were utilized for evaluation purposes. The runtime and cost results for those analyses are displayed in **Table 4** below:

	Average Runtime	Average Cost (estimated)
UF-GWAS (Hail)	1hr 49min*,**	$17.37
PLINK (VCF format)	10hr 32min**,***	$100.48
PLINK (BED/BIM/FAM format)	52 min*,***	$8.28

Table 4.4. Runtime and estimated costs of running *Clostridioides difficile* GWAS using UF-GWAS and PLINK; * (p-value = 0.02108), ** (p-value = 0.01008), *** (p-value = 0.003977) using a Mann-Whitney U Test

All tools were statistically significant with respect to run time against one another, at an alpha value of 0.05. Both tools reported identical genomic results as Aim 2. As a recap, all of the the top variants were on chromosome 6, and the top 5 most significant variants (rs68148149, rs3828840, rs35882239, rs35882239, rs35222480) mapped to the intergenic region between the HLA-DRB5 and HLA-DRB1 genes in the beta block of the MHC Class II region.

4.3.3 Guidelines for bioinformatics software development within the context of NGRs

After reviewing all of the knowledge sources from this dissertation, the following guidelines were created to better inform user-centered bioinformatics tool design to best engage novice users (also refer to **Figure 4.1** below which frames the UCD ideology in the context of the dissertation):

1. **Engage the target user base prior to software tool development**
    a. Knowledge source: Aim 1 Literature Review, Aim 1 Needs Assessment; UCD Framework Section "Understand User"
    b. Reasoning: As a developer, challenges that you face may not be the same as those faced by your users. Without engaging the target user base, you are limiting yourself to only a few perspectives of how the tool should be designed and the features that it will have. By engaging the target user base through surveys or interviews, a more complete view of what niche the tool will serve is able to be formed. Most available bioinformatics tools from the literature do not properly engage with their target users and suffer usability challenges as a result.
2. **Provide easy access to background biological literature and computational knowledge required to utilize the software tool**
    a. Knowledge source: Aim 1 Needs Assessment; UCD Framework Section "Understand User"

   b. Reasoning: Most participants reported the need for more background information than what is currently provided by most bioinformatics tools. Learning how to use the software was limited by their knowledge of the underlying biology and computational skills. By providing tutorial content and links to relevant publications, users are able to more quickly orient themselves with the knowledge required to begin utilizing the software.

3. **Provide curated datasets to quickly jumpstart analyses** (plenty of publicly available datasets i.e. subsets of 1000 Genomes and Human Genome Diversity Project)
   a. Knowledge source: Aim 1 Needs Assessment; UCD Framework Section "Understand User"
   b. Reasoning: Most participants reported challenges in initial adoption of current widely-used tools related to a lack of available, formatted data. With a curated dataset packaged with the tool, users are able to quickly run tutorial content and begin experimenting with the various parameters to better learn the scope and limits of the software.
4. **Create reproducible and shareable experiments/analyses**
   a. Knowledge source: Aim 1 Needs Assessment; UCD Framework Section "Understand User"
   b. Reasoning: Learning comes from repeated practice and re-running experiments that have been validated as correct from other scientists. Without reproducible and shareable analyses, users have trouble with initial adoption and reported persisting issues with the sustained use of the tool.
5. **Containerize the software tool for easy deployment and initial setup by the users**

    a. Knowledge source: Aim 2 Tool Development, UCD Framework Section "Develop/Refine prototype"
    b. Reasoning: Containerization removes the need for users to install independent dependencies and worry about whether or not the software tool will work on their machine. Additionally, it allows for easy updates when new versions are released and allows users to create their own versions of the software to fit their needs by simply extending the base container image.

6. **Error messages should be consistent, readable, and provide adequate feedback**
    a. Knowledge source: Aim 2 Tool Development, UCD Framework Section "Develop/Refine prototype"
    b. Reasoning: Proper error messages are critical in both the development and use of any bioinformatics tools. However, it is not enough to just report which part of the code has errors or simply print out the troubleshooting log information. Effective error messages are consistent, readable, and quickly enable users to either solve the error or reference material needed to begin active problem solving.
7. **Reduce unnecessary mental effort during use by having a simple user-interface (UI) with consistent navigation mechanisms**
    a. Knowledge source: Aim 2 Tool Development, UCD Framework Section "Develop/Refine prototype"
    b. Reasoning: Simple navigation between different aspects of the tool saves time and reduces overall errors that can occur during use. Complicated navigational and file tree structures lead to user frustration. Minimalistic UIs that utilize universally

    utilized icons (such as arrows and the floppy disk save icon) help users better orient themselves to the different components of the software tool.

8. **Cloud deployment allows for easy access by a diverse user base**
    a. Knowledge sources: Aim 2 Tool Development and Aim 3 Tool Evaluation, UCD Framework Sections “Develop/Refine prototype” and “Observe prospective users”
    b. Reasoning: Although cloud computing has its drawbacks, such as unintuitive cost structure and initial onboarding of new users, it can be a great alternative to local high performance compute (HPC) clusters commonly used by bioinformaticians. UF-GWAS was hosted on the cloud and it was easily accessed by all of the participants. No downloads were necessary and the only requirements for use were a web browser and internet. Also, no significant challenges were faced during deployment and modern computational costs charged by the major cloud providers (Google Cloud, Amazon Web Services, and Microsoft Azure) made cloud deployment a reasonable alternative to local HPCs.

9. System status should be readily visible and continuously updated
    a. Knowledge source: Aim 3 Tool Evaluation, UCD Framework Section “Observe prospective users”
    b. Reasoning: Users want to know what the system is doing and whether or not their actions have impacted the status of the tool. It is important to provide visual cues when actions have taken place (whether that be button presses or background methods/functions being run). By doing so, users can better understand the

outcome of their prior interactions and decide what to do next quickly and intuitively.

10. **Re-engage the users after initial prototyping to effectively evaluate development progress**
    a. Knowledge source: Aim 3 Tool Evaluation, UCD Framework Section “Observe prospective users”
    b. Reasoning: UCD is an iterative process, as shown in **Figure 1** below. Oftentimes there is a time delay between initial engagement of the target users and the development of the software tool. To limit biases that may have formed over this time delay and to evaluate the progress made, it is best to re-engage the user base and have them carefully evaluate sections of the tool and provide feedback.
11. **Write readable documentation and update routinely**
    a. Knowledge sources: Aim 1 Needs Assessment and Aim 3 Tool Evaluation, UCD Framework Sections “Understand User” and “Observe prospective users”
    b. Reasoning: Proper documentation is the backbone of any widely-praised software tool. Well designed documentation allows for easier onboarding, error resolution, and provides users with solid reference material. Most participants of UF-GWAS reported how readable the documentation was and how they were able to utilize that documentation to quickly localize and resolve errors related to both biological and computational knowledge.

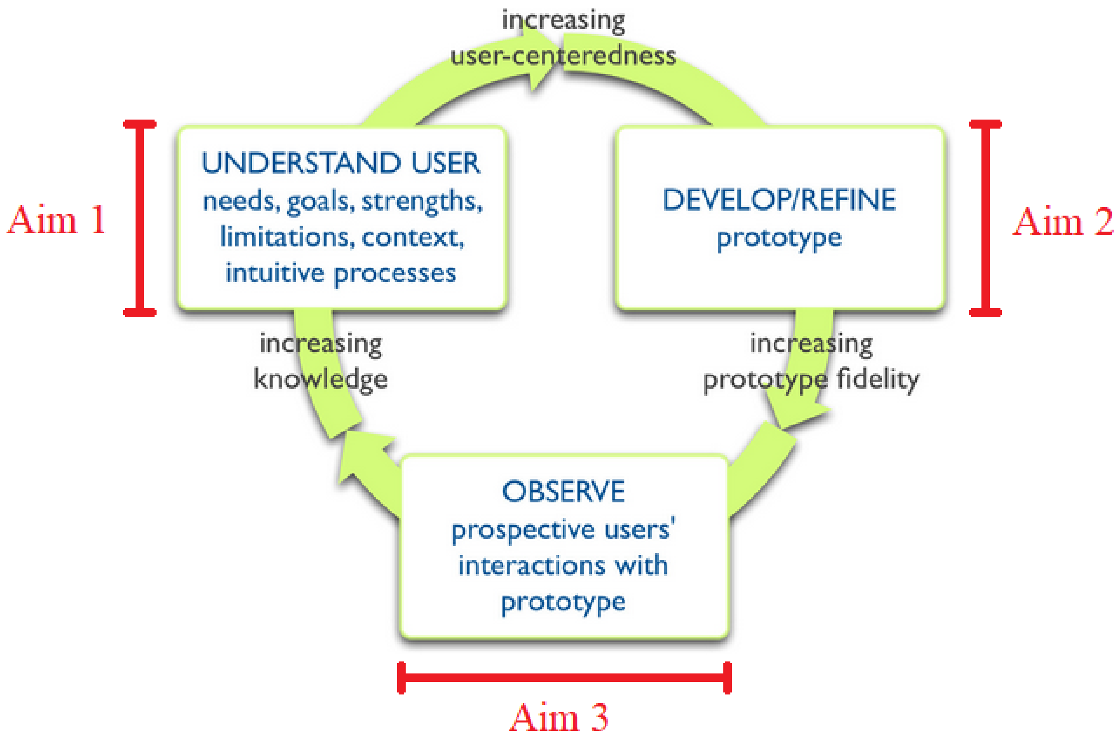


Figure 4.1. Schematic of User-Centered Design Framework. Source: Witteman et al. 2015 [16]

4.4 Discussion

Overall, this aim evaluated the usability of UF-GWAS within the context of novice genomics researchers. It also compared the efficiency and cost of running a GWAS using UF-GWAS versus one of the industry standard software tools, PLINK. Lastly, knowledge gained from all three aims was utilized to create a set of guidelines for future user-centered bioinformatics software development. In the following sections, we will explore key insights that were gained from both the UF-GWAS evaluation and guideline development.

4.4.1 UF-GWAS Evaluation

4.4.1.a Novice User Evaluations

The user evaluations conducted in this aim offered a more detailed perspective on how NGRs utilize new software tools and the different challenges they face during initial adoption, exploration, and experimentation. Almost all of the users who evaluated UF-GWAS responded very positively to the overall design and content provided. JupyterNotebooks allowed for participants who did not have an extensive computational background to complete a tutorial GWAS using 1000 Genomes data and the provided biological background helped reinforce the biological domain knowledge needed to complete the analysis. As evident by the results from the SUS and PSSUQ survey instruments, participants perceived UF-GWAS to be quite user-friendly and from the free-response section of the evaluation survey, they attributed a good bit of usability to the detailed documentation and content built into UF-GWAS. Participants 2 and 5 sum it up nicely by saying:

> *"I understand that there's a handoff with information overload, but* ***I think I appreciated having more explanations/details that are readily available than having to do extra searching elsewhere.****" (P2)*

> *"****UF-GWAS seems to be very promising, especially the provided background information.*** *I think it will be a boon to those starting out with genomic data analysis, and the good throughput will probably keep them around!" (P5)*

This behavior of information gathering by novice users has been studied in other disciplines but much work is still needed in the bioinformatics landscape [9]. UF-GWAS sought to strike a balance between providing necessary onboarding content and unnecessary information overload. By splitting up the tool into manageable chunks, the various JupyterNotebooks, users were able to quickly ascertain the tool's purpose and navigate to specific sections to complete the evaluation task.

Additionally, users responded positively to how easy the tool was to access and utilize due it being hosted on the cloud. Without the burden of installing dependencies and having to set up a work environment, users were able to complete an entire GWAS within the hour allotted for the evaluation. Initial burden of setup and a lack of formatted data to jumpstart tool usage were some of the challenges NGRs reported during Aim 1 of this dissertation and UF-GWAS's containerized design helped circumnavigate these usability burdens. Users did, however, report other usability challenges that UF-GWAS and other bioinformatics tools will have to solve in order to garner extended usage. These challenges include: more adjustable UI components, background information on Hail (and other underlying dependencies used for statistical analysis), and knowledge checkpoints/quizzes. Potential solutions to these challenges include: providing additional support beyond documentation in the form of a support email or educational forum where users can interact with one another and solve problems collaboratively, adding more interactive widgets throughout the tool, and building in the ability to create workspaces to allow for easier project management and for tutorial purposes. Existing tools that have incorporated these solutions along with some others report higher throughput of results and more sustained tool use [30, 32]. Much work is still needed to be done to fully explore the feasibility of existing solutions to the aforementioned challenges faced by NGRs.

4.4.1.b UF-GWAS and PLINK

In order to evaluate the efficiency and accuracy of UF-GWAS versus PLINK, an industry standard software tool commonly utilized to conduct GWAS experiments, we recreated the *Clostridioides difficile* GWAS presented in Aim 2. Both UF-GWAS and PLINK were able to replicate the original results by Ferar et al. but differed in the time taken to complete the experiments and overall cost of the computational architecture [100]. Although UF-GWAS was slower than PLINK when the PLINK binary file (BED/BIM/FAM) format was utilized, it performed much faster and at a fraction of the cost when PLINK utilized the more common VCF file format. The time it takes to convert the VCF file into the PLINK binary files was not added to the PLINK binary experiment time or cost measurements. Future work will have to be done in order to compare how UF-GWAS compares versus PLINK in terms of time and cost if the entire analysis pipeline (data cleaning, data conversion, quality control, statistical analyses, visualization) is run. Although existing literature has shown the historical and growing costs of storing genomic data, this study, along with the work done by Krissaane et al., aim to offer practical estimates of the time and cost needed to run large-scale genomic analyses in the cloud [91, 103].

4.4.2 Guideline Development

Throughout this dissertation, many different sources of knowledge were gathered from literature review, qualitative engagement with NGRs, and quantitative analyses related to the *Clostridioides difficile* GWAS and the design/deployment of UF-GWAS across different cloud platforms. In order to concentrate the key insights gained throughout these investigations, 11 guidelines were created in order to guide future bioinformaticians and developers to create more user-friendly software tools. Quite a few guidelines exist on bioinformatics training and the required competencies, yet there is a lack of recommendations on how we should best develop the tools needed by these novice researchers during their training and initial investigations [47, 114]. The guidelines presented in this study encompass all components of the UCD framework and can be utilized to guide software development from initial inception all the way to deployment and dissemination. Although the guidelines were developed by the primary investigator utilizing the knowledge gained from only this dissertation, they mirror the generalizability of practice guidelines and current protocols available in other disciplines of study such as Lavappa et al.'s work on guidelines needed when creating usable software for architecture buildings and Kumuthini et al.'s work on developing protocols for effective bioinformatics research teams [132, 133]. The guidelines developed in this study act as scaffolding needed to create more usable bioinformatics software. This scaffolding is much needed especially within the context of the NHGRI's 2020 strategic vision, calling for the construction of stronger foundations and breaking down barriers faced by new and veteran investigators [117].

4.4.3 Limitations and Suggested Improvements

Although the user evaluations offered a richer, more nuanced perspective into the challenges faced by NGRs when utilizing UF-GWAS, a relatively small number of participants were involved in the evaluation process. However, despite this, average SUS/PSSUQ scores and free-response themes did not appear to change significantly after 6 participants, indicating theoretical saturation. Furthermore, participants were only selected from the University of Washington and Tulane University and were mostly of self-identified Asian descent. Their training and experience may not reflect the views of NGRs from other institutions and backgrounds. For future investigations, it is important to include a more diverse participant pool in order to capture viewpoints that were not present in this study.

With respect to the evaluation of UF-GWAS and PLINK, due to limited time and budget, the *Clostridioides difficile* GWAS was only run on Google Cloud Platform. As a result, the cost and runtime estimates may not be truly generalizable to other cloud providers. Future work would replicate this GWAS on Amazon Web Services and Microsoft Azure to test for cost and runtime consistencies. Finally, it is important to recognize the somewhat subjective nature of guideline generation and knowledge interpretation. While the usability guidelines were reviewed by the doctoral committee, it was the primary investigator who organized and synthesized the overall findings. The final results only represent one possible interpretation of the available data and many other interpretations may exist depending on the investigator.

4.5 Conclusion

In this aim, we evaluated the usability of UF-GWAS through user evaluations by novice genomic researchers. We also tested the accuracy and efficiency of UF-GWAS against the industry standard GWAS tool PLINK. Through the user evaluations, we found that UF-GWAS scored highly across two survey instruments, the SUS and PSSUQ, for being user-friendly. Users appreciated the amount of domain knowledge provided, ease of access via cloud hosting, and expressed interest in utilizing the tool or similar tools in the future. UF-GWAS also performed well against PLINK in terms of computational cost and time taken to perform analyses. Finally, in this aim, we compiled all knowledge gained throughout this dissertation into a set of practical guidelines that future researchers can utilize to develop more user-friendly bioinformatics software. These guidelines are generalizable and hopefully will be utilized to create a stronger foundation of analysis software to enable more robust analyses by future bioinformaticians. User-centered design is an interactive process and requires conscious effort to execute properly. Although this study displayed the strengths of applying the UCD framework, it is important to recognize that more work needs to be done in order to sustain the usability of the system over time. Future work should involve exploring how users utilize these UCD-based software tools after initial adoption and experimentation in order to adapt these tools to fit the users' evolving needs.

CHAPTER 5: CONCLUSIONS AND SUMMARY OF CONTRIBUTIONS

In this work, we utilized the user-centered design (UCD) framework to create a more user-friendly bioinformatics tool (UF-GWAS), targeted specifically for novice genomics researchers (NGRs). Using knowledge gathered from literature review, needs assessment, user personas, the design and deployment of UF-GWAS, and its evaluation by the target user base, we also developed an evaluation rubric for bioinformatics tools and put forth a set of guidelines that future informaticians can follow to create more usable software. User-centered design is iterative and in order to fully unlock its potential, more diligent work will have to be done by both developers and their users in the future. When UCD is executed effectively, bioinformatics software can evolve to better support the changing needs of existing researchers and the incoming diverse workforce, leading to a strong foundation that is necessary for the future of genomics research.

This work offers several contributions to the field of bioinformatics, consumer health informatics, cloud-agnostic software development, and user-centered design research:

1. A novel evaluation rubric informed by both a literature review of existing bioinformatics software and a needs assessment of an under-represented keystone user base (NGRs) (**Aim 1**)
2. A novel user friendly, cloud-agnostic GWAS tool (UF-GWAS) that is capable of running large-scale genomic analyses efficiently and accurately while remaining cost-efficient compared to existing software (**Aim 2**)

3. Guidelines for best practices when developing user-friendly bioinformatics software that can be generalized to other projects with diverse users, methodologies, and funding capabilities (**Aim 3**)

APPENDIX

Appendix A: Preliminary Interview Guide

1. Background/Experience of utilizing (genomics software)

- **Could you please briefly introduce yourself** and provide a bit of background on your experience working with genomic data?
- **What do you remember** about your first experiences working with genomics data? How did you get started?
- **EMOTIONAL: How did you feel** when you were working with (genomics software)? Did your feelings change as you worked more with that analysis tool?
- **COGNITIVE: What concerns or questions** did you have about utilizing (genomics software) when you first started using it?

2. Perceived positives and negatives of (genomics software)

- **POS/NEG**: Thinking back, what do you see as **positives or negatives** about utilizing (genomics software)?
- **PIPELINE IMPACT**: How has utilizing (genomics software) **influenced your analysis pipeline**, if at all?
- **OVERALL IMPACT**: Has utilizing (genomics software) **influenced any other aspect of your experience** being a genomics researcher?

3. Preferences for (genomics software)

- **DO IT AGAIN**: If you could do your analysis process all over again, would you utilize (genomics software) again or utilize something else?
- **WHAT WENT WELL**: What went well about the process? (how notified, educational needs, provider support)
- **WHAT TO CHANGE**: Do you have any recommendations for changes or additions to (genomics software) that would aid in your research? Could you perhaps rank these changes/additions? Have you considered ___, ___, or ___?

4. Exploratory questions:

- As the NHGRI and other genomics research organizations are moving toward cloud computing, what challenges/concerns do you anticipate facing when conducting genomic research in the cloud?

Appendix B: Final Interview Guide

1. Background/Experience of utilizing (genomics software)

- **Could you please briefly introduce yourself** and provide a bit of background on your experience working with genomic data?

- **What do you remember** about your first experiences working with genomics data? How did you get started? What educational resources did you use, if any?

- **What kinds of genomic analysis software** have you utilized to conduct your research? Could you name a few examples of the tools you have used in the past? What are you using the (genomics software) for (how did/does your workflow look like)?

- **EMOTIONAL: How did you feel** when you were working with (genomics software)? Did your feelings change as you worked more with that analysis tool?

- **COGNITIVE: What concerns or questions** did you have about utilizing (genomics software) when you first started using it? What, if any information was missing? Do you think this played a big role in the analysis timeline/performance?

- **OVERALL EXPERIENCE:** Overall, how would you describe your experience working with (genomics software)?

2. Perceived positives and negatives of (genomics software)

- **POS/NEG**: Thinking back, what do you see as **positives or negatives** about utilizing (genomics software)?

- **PIPELINE IMPACT**: How has utilizing (genomics software) **influenced your analysis pipeline**, if at all?

- **UPKEEP/COST IMPACT**: Has utilizing (genomics software) **influenced your cost/upkeep of your experiments**, if at all? (i.e. having to store intermediate files? Using more computational resources?)

- **OVERALL IMPACT**: Has utilizing (genomics software) **influenced any other aspect of your experience** being a genomics researcher?

3. Preferences for (genomics software)

- **DO IT AGAIN**: If you could do your analysis process all over again, would you utilize (genomics software) again or utilize something else?

- **WHAT WENT WELL**: What went well about the process? (how notified, educational needs, provider support)

- **WHAT TO CHANGE**: Do you have any recommendations for changes or additions to (genomics software) that would aid in your research? Could you perhaps rank these changes/additions? Have you considered ___, ___, or ___?

4. Exploratory questions:

- As the NHGRI and other genomics research organizations are moving toward cloud computing, what challenges/concerns do you anticipate facing when conducting genomic research in the cloud?
  - Privacy
  - Cost

Appendix C: Literature Review Study Characteristics

Name of Bioinformatics Software/Tool	Paper/Study /Tool Link	Year Published	Target Audience	Target Analysis	User-centered Design Considerations	Cloud-Based	GUI	GITHUB/Code Open-Source	Tutorials/Documentation	Free-To-Use	Data Restrictions
PLINK	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1950838/	2007	Intermediate/Advanced Researchers	Whole-Genome Association and Population-Based Linkage Analyses	None	FALSE	TRUE	TRUE	TRUE	TRUE	None
ANVIL Terra	https://anvilproject.org/learn/introduction/intro-to-terra		Beginner/Intermediate/Advanced Researchers	Analyze high-value open and controlled access genomic datasets with popular analysis	Claims to have done user needs assessment, provides feedback opportunities	TRUE	TRUE	TRUE	TRUE	FALSE (Credits)	Controlled Datasets
AllOfUs Research Platform	https://databrowser.researchallofus.org/ehr/conditions, https://www.nejm.org/doi/full/10.1056/NEJMsr1809937	2019	Beginner/Intermediate/Advanced Researchers	Enroll a diverse group of at least 1 million persons in the United States in order to accelerate biomedical research and improve health	Claims to have done user needs assessment, provides feedback opportunities	TRUE	TRUE	TRUE	TRUE	FALSE (Credits)	Only AllOf US Dataset
GWASTools	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3519456/	2012	Intermediate/Advanced Researchers	Quality control and analysis of genome-wide association studies (GWAS)	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
Hail	https://blog.hail.is/introtohail/	2015	Intermediate/Advanced Researchers	Genetic data manipulation and analysis	None	TRUE	FALSE	TRUE	TRUE	TRUE	None
OrdinalGWAS-JULIA	https://openmendel.github.io/OrdinalGWAS.jl/latest/	2020	Intermediate/Advanced Researchers	Whole-genome analyses, including shrinkage estimation and variable selection methods	None	FALSE	TRUE (Jupyter Notebooks)	TRUE	TRUE	TRUE	None
HAPPI GWAS	https://academic.oup.com/	2020	Intermediate/Advanced Researchers	Data cleaning, GWAS, Visualization/Gene Identification	Mentions in	FALSE	FALSE	TRUE	TRUE	TRUE	None

	bioinformatics/article/36/17/4655/5862010?login=false				Introduction but no usability methods						
GW-SEM 2.0	https://pubmed.ncbi.nlm.nih.gov/33604756/	2021	Intermediate/Advanced Researchers	Multi-variate GWAS	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
LDPRED2	https://academic.oup.com/bioinformatics/article/36/22-23/5424/6039173	2020	Intermediate/Advanced Researchers	Deriving polygenic scores based on summary statistics and a matrix of correlation	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
GeneSpy	https://academic.oup.com/bioinformatics/article/35/2/329/5038465?login=false	2019	Intermediate/Advanced Researchers	Visualization of genomic context can provide essential information to infer functional links between genes	Claims to be user-friendly but does not specify in implementation	FALSE	TRUE	TRUE	TRUE	TRUE	None
RNAdetector	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8173825/	2021	Beginner/Intermediate Researchers	RNA-Seq data analyses	GUI, focused on interests of "non-expert" users	TRUE	TRUE	TRUE	TRUE	TRUE	None
BioWardrobe	https://genomebiology.biomedcentral.com/articles/10.1186/s13059-015-0720-3	2015	Intermediate/Advanced Researchers	Integrated platform for analysis of epigenomics and transcriptomics data	None	FALSE	TRUE	TRUE	TRUE	TRUE	None
easyRNASeq	https://academic.oup.com/bioinformatics/article/28/19/2532/290157	2012	Intermediate/Advanced Researchers	RNA-Seq data processing and analysis	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
BioJupies	https://www.cell.com/cell-systems/fulltext/S2405-4712(18)30432-0?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405471218304320%3Fshowall%3Dtrue	2018	Beginner/Intermediate Researchers	RNA-seq data analyses	Some, utilizes Jupyter-notebooks to guide users	TRUE	TRUE (Jupyter Notebooks)	TRUE	TRUE	TRUE	None

TRAPLINE	https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0873-9	2016	Beginner/Intermediate Researchers	RNA-Seq data processing and analysis	Some, aims to target "researchers with limited or no programming skills"	TRUE	TRUE	TRUE	TRUE	TRUE	None
NGSCloud2	https://peerj.com/articles/11237/	2021	Intermediate/Advanced Researchers	De novo RNAseq analysis of non-model species	None	TRUE	TRUE	TRUE	TRUE	TRUE	None
InputAccur	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9351229/	2022	Intermediate/Advanced Researchers	Calculation of genotype-imputation accuracy-measures	Mentions in Introduction but no usability methods	FALSE	FALSE	TRUE	TRUE	TRUE	None
GWASinspector	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8034536/	2021	Intermediate/Advanced Researchers	Quality control of genome-wide association study results	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
eQTpLot	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8285863/	2021	Intermediate/Advanced Researchers	Visualization of colocalization between eQTL and GWAS signals (R-package)	Mentions in Introduction but no usability methods	FALSE	FALSE	TRUE	TRUE	TRUE	None
VCFtools	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137218/	2011	Intermediate/Advanced Researchers	Parsing, analyzing and manipulating VCF files	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
MaxQuant	https://www.nature.com/articles/nprot.2016.136	2016	Intermediate/Advanced Researchers	Mass-spectrometry (MS)-based proteomics data analysis	GUI, focused on interests of "non-expert" users	FALSE	TRUE	TRUE	TRUE	TRUE	None
Galaxy	https://academic.oup.com/nar/article/46/W1/W537/5001157	2018	Intermediate/Advanced Researchers	Wide array of biomedical analyses	GUI, focused on interests of "non-expert" users	TRUE	TRUE	TRUE	TRUE	TRUE	None
BioBin (Introduction)	https://bmcmedgenomics.biomedcentral.com/articles/10.1186/1755-8794-6-S2-S6	2013	Intermediate/Advanced Researchers	Automating the binning of rare variants using publicly available biological knowledge	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
BLINK	https://academic.oup.com/gigascience/article/8/2/giy154/5238723	2018	Intermediate/Advanced Researchers	(GWAS) to identify genes controlling human diseases and agricultural traits. The algorithm uses Bayes and linkage disequilibrium information.	None	FALSE	FALSE	TRUE	TRUE	TRUE	None

Rice GWAS Pipeline	https://www.nature.com/articles/ncomms10532	2016	Beginner/Intermediate Researchers	GWAS Pipeline including visualization for rice	GUI, focused on interests of "non-expert" users	FALSE	TRUE	TRUE	TRUE	TRUE	None
GWAS4D	https://academic.oup.com/nar/article/46/W1/W114/4996581	2018	Intermediate/Advanced Researchers	Multidimensional analysis of context-specific regulatory variant for human complex diseases and traits	None	FALSE	TRUE	TRUE	TRUE	TRUE	None
GWAtoolbox	https://academic.oup.com/bioinformatics/article/28/3/444/189687	2011	Intermediate/Advanced Researchers	Fast quality control and handling of genome-wide association studies meta-analysis data	None	FALSE	FALSE	TRUE	TRUE	TRUE	None
Haploview	https://academic.oup.com/bioinformatics/article/21/2/263/186662	2004	Intermediate/Advanced Researchers	Analysis and visualization of LD and haplotype maps	None	FALSE	TRUE	TRUE	TRUE	TRUE	None
IPGWAS (QC Info)	https://www.sciencedirect.com/science/article/abs/pii/S0006291X12007991?via%3Dihub	2012	Intermediate/Advanced Researchers	Integrated pipeline for rational quality control and association analysis of genome-wide genetic studies	None	FALSE	TRUE	TRUE	FALSE	TRUE	None
MGAS	https://academic.oup.com/bioinformatics/article/31/7/1007/181296	2015	Intermediate/Advanced Researchers	Multivariate gene-based genome-wide association analysis	None	FALSE	FALSE	TRUE	FALSE	TRUE	None
Seurat	https://www.cell.com/cell/fulltext/S0092-8674(21)00583-3?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS0092867421005833%3Fshowall%3Dtrue	2021	Beginner/Intermediate Researchers	Integrated analysis of multimodal single-cell data	Some, but not mentioned in the paper	FALSE	FALSE	TRUE	TRUE	TRUE	None
Spot	https://academic.oup.com/nar/article/38/suppl_2/W201/1123082?login=false	2010	Intermediate/Advanced Researchers	Using biological databases to prioritize SNPs after a genome-wide association study	None	FALSE	TRUE	TRUE	TRUE	TRUE	None
Garfield	https://www.biorxiv.org/content/10.110	2016	Intermediate/Advanced Researchers	GWAS Analysis of Regulatory or Functional Information Enrichment with LD correction	None	FALSE	FALSE	TRUE	TRUE	TRUE	None

	1/085738v1										
gwasurvivr	https://pubmed.ncbi.nlm.nih.gov/30395168/	2018	Intermediate/Advanced Researchers	Genome wide survival analysis	None	FALSE	FALSE	TRUE	TRUE	TRUE	None

Appendix D: Personas

Orlando Mccarthy

- Job Title: 1st year Graduate Student
- Demographics:
    - Age: 23
    - Gender: Male
    - Education: Bachelor's Degree
    - Interests: Playing guitar and piano, running with his dog, cultivating small plants
- Goals/tasks:
    - He is motivated to learn as much as he can about informatics and wants to utilize his background in public health to study health disparities between populations. His major concern is lacking biological knowledge to add context to the results he generates.
        - He spends his time:
            - Reading up background literature on genomics
            - Exploring publicly available datasets and methodologies
            - Running small scale analyses on his university's HPC
- Environment: He is comfortable using a computer and considers himself an intermediate Internet user. He has prior experience programming from coursework and small projects,

but has not officially been involved with a large scale research project. Languages he knows include basic bash, Java, and basic Python.

Irene Young

- Job Title: Research Consultant
- Demographics:
    - Age: 29
    - Gender: Female
    - Education: Masters Degree
    - Interests: Kayaking, exploring local farmers markets, doodling
- Goals/tasks:
    - She is big-picture focused and works primarily on coordinating efforts to improve data standardization. Her primary concern is utilizing many different softwares and data types to perform a single analysis (leading to many intermediate files being generated and constantly switching softwares).
    - She spends her time:
        - Organizing different files by type, origin, and their metadata
        - Running genomics analysis for variant discovery
- Environment: She has prior experience with research projects and has conducted two genomics related studies for different grants as part of her RA-ship. Languages she knows include bash, Python, SQL, and R.

Scott Foster

- Job Title: 3rd year Graduate Student
- Demographics:
    - Age: 38
    - Gender: Male
    - Education: MD
    - Interests: College football, building go-karts with his kids, hiking
- Goals/tasks:
    - He is detail-oriented with a good leadership background. One of his major concerns is utilizing new software tools without fully having the computational background to do so.
    - He spends his time:
        - Looking up documentation and tutorials on commonly used software
        - Writing up results to publish in journals
        - Leading a small data coordination effort as part of his RA-ship
- Environment: From his medical training, he has strong biological knowledge and has some experience with biological research. He has some experiences utilizing computers, but mostly for casual browsing. He has very limited programming experience, only from small projects from different courses taken. Languages he knows are basic Python.

Appendix E

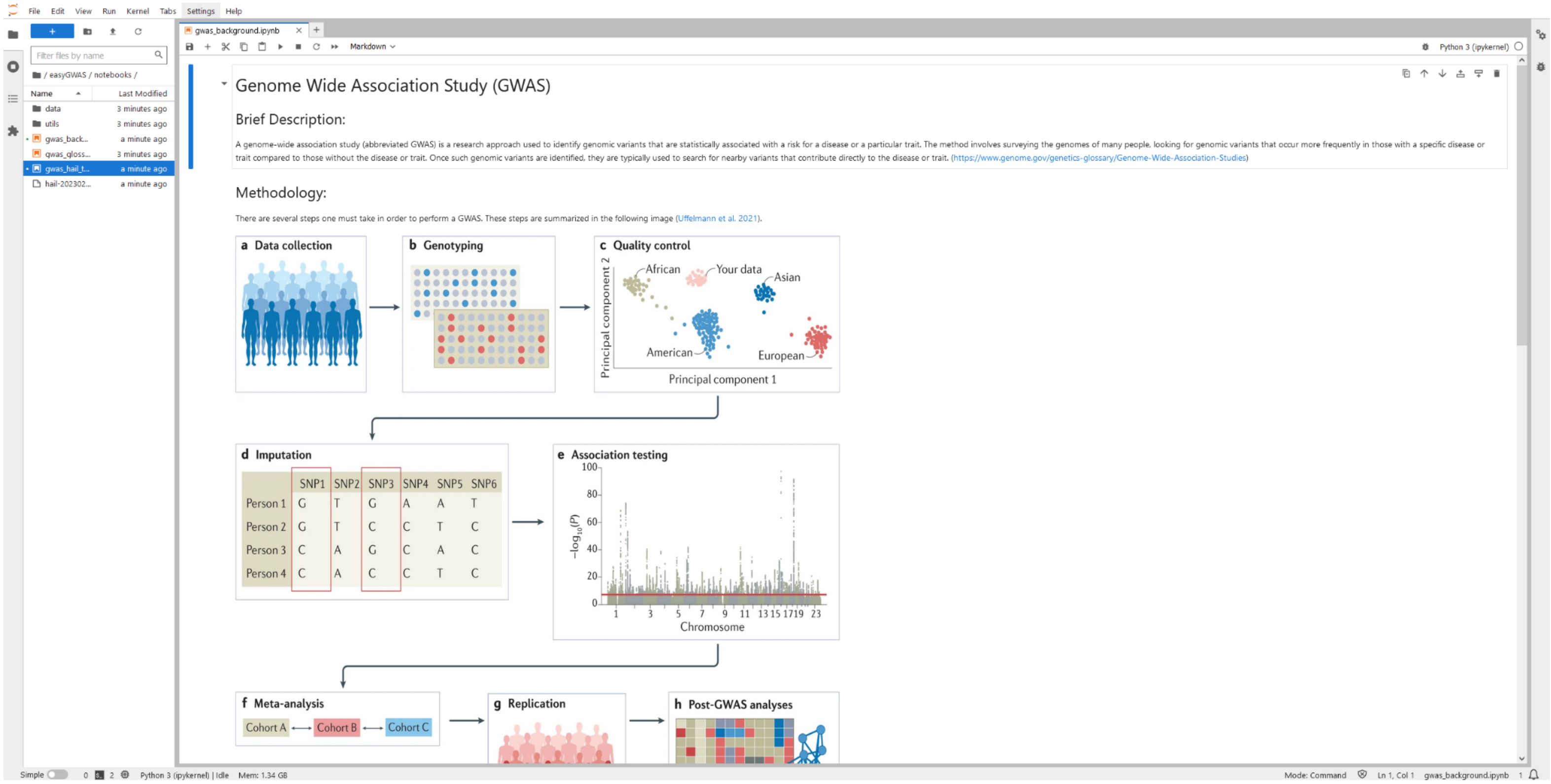
User Evaluation Instruction Set:

Introduction

Thank you for taking the time to participate in the evaluation of UF-GWAS, a user-friendly genome wide association study software tool. Before you begin, please ensure you have roughly 30 - 60 minutes of available time in order to fully complete the evaluation instructions and fill out a survey at the end. If any instruction is unclear or you are unable to progress beyond any stage of the evaluation, please note that down (a portion of the final survey will have room for you to provide feedback) as it will help me improve the software tool for future users.

Evaluation of UF-GWAS

1. Please navigate to https://uf-gwas.org and use the username provided in the instruction email as your login credential. For the password field, please choose any password (as long as it is at least eight characters). You will utilize this password if you wish to login anytime in the future after the evaluation period.
    a. The first-time login may take a few minutes as the tool is creating your analysis environment for the first time.
2. You should be greeted with a JupyterLab interface similar to the one pictured below:



Genome Wide Association Study (GWAS)

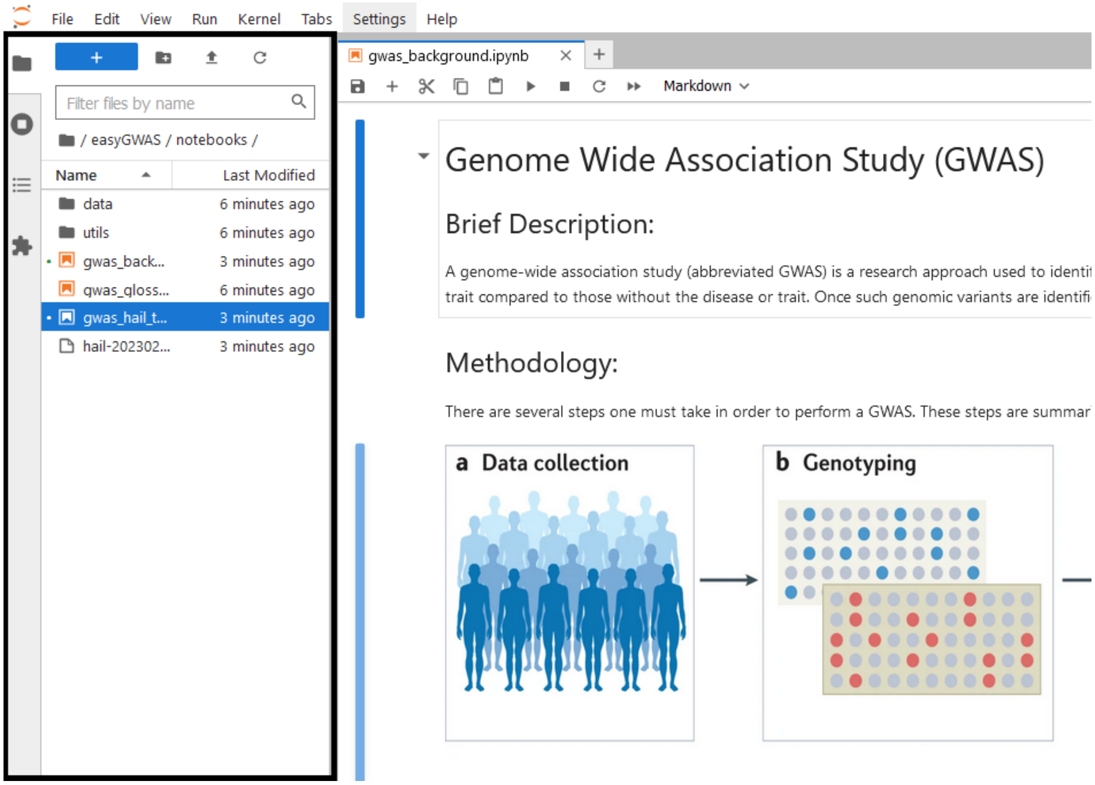
Brief Description:

A genome-wide association study (abbreviated GWAS) is a research approach used to identify genomic variants that are statistically associated with a risk for a disease or a particular trait. The method involves surveying the genomes of many people, looking for genomic variants that occur more frequently in those with a specific disease or trait compared to those without the disease or trait. Once such genomic variants are identified, they are typically used to search for nearby variants that contribute directly to the disease or trait. (https://www.genome.gov/genetics-glossary/Genome-Wide-Association-Studies)

Methodology:

There are several steps one must take in order to perform a GWAS. These steps are summarized in the following image (Uffelmann et al. 2021).

3. Take some time and read through the JupyterNotebook titled **gwas_background.ipynb**. If you utilized the link above, the **gwas_background** notebook should automatically open as the only tab in your analysis environment.
4. Navigate to the **gwas_hail_tutorial.ipynb** notebook utilizing the navigation window pictured below.



Name	Last Modified
data	6 minutes ago
utils	6 minutes ago
gwas_back...	3 minutes ago
gwas_gloss...	6 minutes ago
gwas_hail_t...	3 minutes ago
hail-202302...	3 minutes ago

Genome Wide Association Study (GWAS)

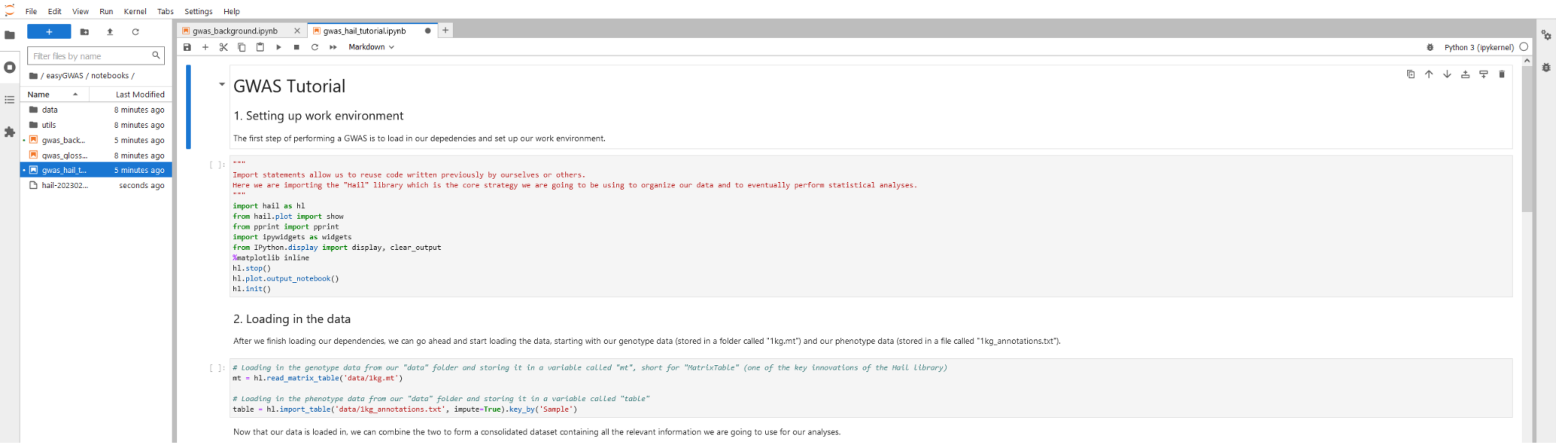
Brief Description:

A genome-wide association study (abbreviated GWAS) is a research approach used to identi
trait compared to those without the disease or trait. Once such genomic variants are identifi

Methodology:

There are several steps one must take in order to perform a GWAS. These steps are summar

If done successfully, you should see the following notebook open as a new tab.



Name	Last Modified
data	8 minutes ago
utils	8 minutes ago
gwas_back...	5 minutes ago
gwas_gloss...	8 minutes ago
gwas_hail_t...	5 minutes ago
hail-202302...	seconds ago

GWAS Tutorial

1. Setting up work environment

The first step of performing a GWAS is to load in our depedencies and set up our work environment.

```
[ ]: """
Import statements allow us to reuse code written previously by ourselves or others.
Here we are importing the "Hail" library which is the core strategy we are going to be using to organize our data and to eventually perform statistical analyses.
"""
import hail as hl
from hail.plot import show
from pprint import pprint
import ipywidgets as widgets
from IPython.display import display, clear_output
%matplotlib inline
hl.stop()
hl.plot.output_notebook()
hl.init()
```

2. Loading in the data

After we finish loading our dependencies, we can go ahead and start loading the data, starting with our genotype data (stored in a folder called "1kg.mt") and our phenotype data (stored in a file called "1kg_annotations.txt").

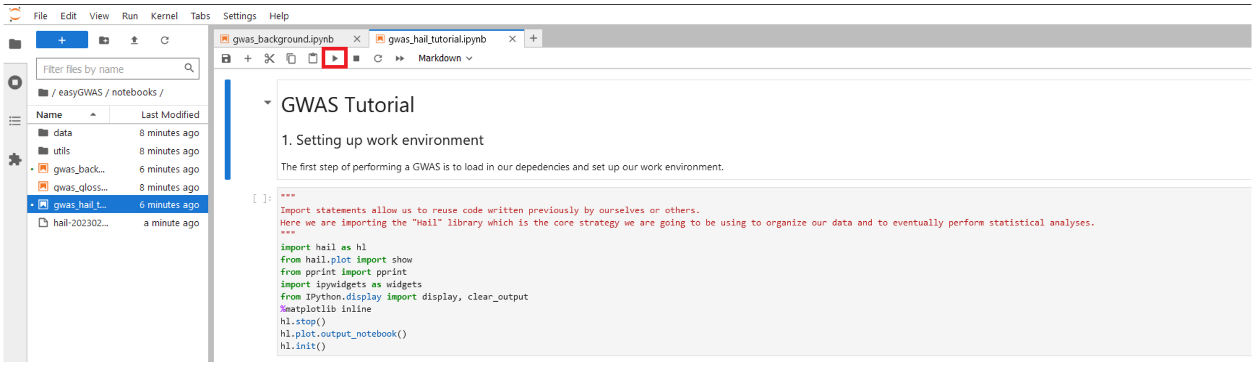
```
[ ]: # Loading in the genotype data from our "data" folder and storing it in a variable called "mt", short for "MatrixTable" (one of the key innovations of the Hail library)
mt = hl.read_matrix_table('data/1kg.mt')

# Loading in the phenotype data from our "data" folder and storing it in a variable called "table"
table = hl.import_table('data/1kg_annotations.txt', impute=True).key_by('Sample')
```

Now that our data is loaded in, we can combine the two to form a consolidated dataset containing all the relevant information we are going to use for our analyses.

5. Utilizing either the play button outlined in red below or the keyboard shortcut "SHIFT+ENTER", begin running each code and markdown cell in the

gwas_hail_tutorial notebook. You will have to click play or use the keyboard shortcut for each cell in the notebook.



GWAS Tutorial

1. Setting up work environment

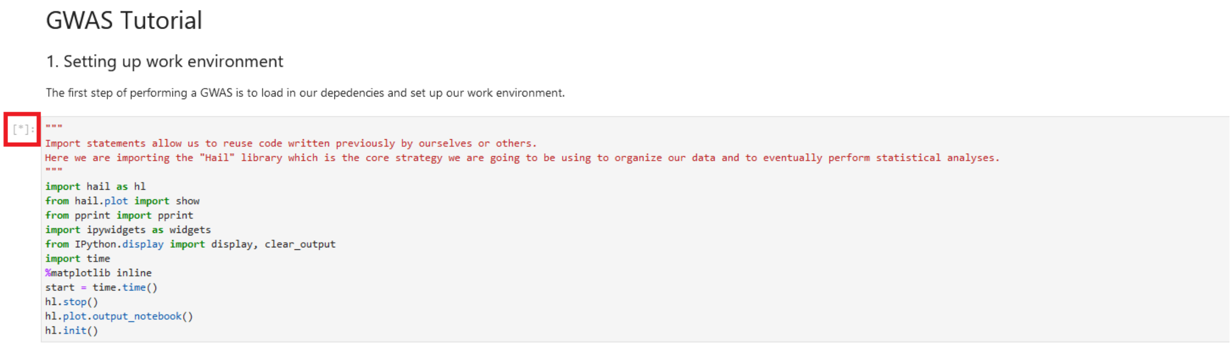
The first step of performing a GWAS is to load in our depedencies and set up our work environment.

```
"""
Import statements allow us to reuse code written previously by ourselves or others.
Here we are importing the "Hail" library which is the core strategy we are going to be using to organize our data and to eventually perform statistical analyses.
"""
import hail as hl
from hail.plot import show
from pprint import pprint
import ipywidgets as widgets
from IPython.display import display, clear_output
%matplotlib inline
hl.stop()
hl.plot.output_notebook()
hl.init()
```

a. If you have never utilized JupyterNotebooks before, there is a quick tutorial notebook titled "**broad_juptyer_hail_tutorial.ipynb**" in the **utils** folder. Feel free to explore that notebook before continuing.

    i. Quick tip: In Jupyter, if a cell is running, the corresponding "box" next to the cell will be filled in with an asterisk. Once the cell is done running, it will be filled with a number, referring to the order of the cells that were run. Check the images below:

    Code running:



GWAS Tutorial

1. Setting up work environment

The first step of performing a GWAS is to load in our depedencies and set up our work environment.

```
"""
Import statements allow us to reuse code written previously by ourselves or others.
Here we are importing the "Hail" library which is the core strategy we are going to be using to organize our data and to eventually perform statistical analyses.
"""
import hail as hl
from hail.plot import show
from pprint import pprint
import ipywidgets as widgets
from IPython.display import display, clear_output
import time
%matplotlib inline
start = time.time()
hl.stop()
hl.plot.output_notebook()
hl.init()
```

Code finished running:

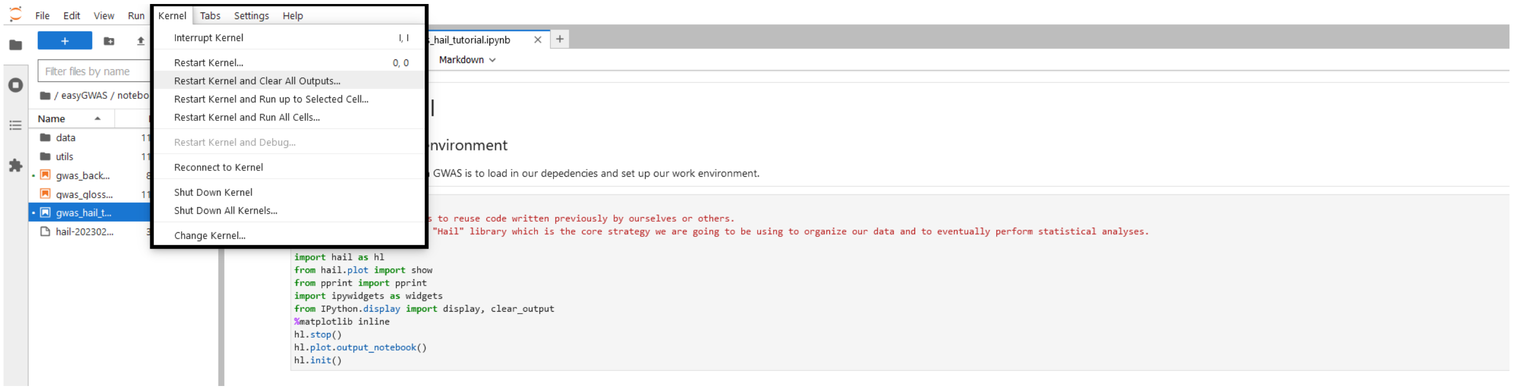
GWAS Tutorial

1. Setting up work environment

The first step of performing a GWAS is to load in our depedencies and set up our work environment.

```
[1]: """
     Import statements allow us to reuse code written previously by ourselves or others.
     Here we are importing the "Hail" library which is the core strategy we are going to be using to organize our data and to eventually perform statistical analyses.
     """
     import hail as hl
     from hail.plot import show
     from pprint import pprint
     import ipywidgets as widgets
     from IPython.display import display, clear_output
     import time
     %matplotlib inline
     start = time.time()
     hl.stop()
     hl.plot.output_notebook()
     hl.init()
```

b. If you get stuck during any stage of the analysis, you can restart the tutorial by going to the Kernel menu at the top and selecting "Restart Kernel and Clear Output" as pictured below:



nvironment

GWAS is to load in our depedencies and set up our work environment.

```
s to reuse code written previously by ourselves or others.
 "Hail" library which is the core strategy we are going to be using to organize our data and to eventually perform statistical analyses.

import hail as hl
from hail.plot import show
from pprint import pprint
import ipywidgets as widgets
from IPython.display import display, clear_output
%matplotlib inline
hl.stop()
hl.plot.output_notebook()
hl.init()
```

6. Once you finish the tutorial, explore some of the supporting notebooks in the **utils** folder by double clicking the folder from the navigation menu and selecting any of the following supported notebooks: "gwas_glossary.ipynb", "gwas_template.ipynb", "broad_jupyter_hail_tutorial.ipynb"
7. If you want to check the results of the tutorial, please open the "gwas_hail_tutorial_solution.pdf" file located in the **utils** folder.
8. Fill out the survey utilizing the link provided in the instruction email.

Additional testing:

1. Navigate into the **utils** folder and open the **gwas_template.ipynb**. This file contains all of the Python code from the tutorial.
2. Run the first few cells up until the QC steps.
3. Modify the code so that during the QC steps, a **minor allele frequency of 0.03** is used, a **sample call rate of 0.95** is used, and the **number of principal components used during**

the GWAS is two (the first two principal components) instead of the three used in the tutorial.

4. Run the QC cell after modifying the values above and run the cells leading up to the GWAS
5. Rerun the GWAS and observe the changes.

Appendix F

SUS Survey Results by Participant (1: Strongly Disagree, 5: Strongly Agree, green = better usability)

Questions	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	AVERAGE
I think that I would like to use this system frequently.	5	4	5	5	5	5	4	5	4	2	4.4
I found the system unnecessarily complex.	2	1	1	1	1	1	1	1	1	1	1.1
I thought the system was easy to use.	4	4	5	5	4	4	5	4	5	5	4.5
I think that I would need the support of a technical person to be able to use this system.	2	1	1	1	1	1	1	1	2	1	1.2
I found the various functions in this system were well integrated.	4	3	4	5	5	5	5	5	5	5	4.6
I thought there was too much inconsistency in this system.	1	1	1	1	1	1	1	1	1	1	1
I would imagine that most people would learn to use this system very quickly.	4	3	5	4	5	5	5	5	4	5	4.5
I found the system very cumbersome to use.	1	1	1	1	1	1	1	1	2	1	1.1
I felt very confident using the system.	5	3	5	5	4	5	4	5	5	5	4.6
I needed to learn a lot of things before I could get going with this system.	2	2	3	1	2	1	1	1	2	1	1.6

PSSUQ Survey Results by Participant (1: Strongly Agree, 7: Strongly Disagree, green = better usability)

Questions	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	AVERAGE
Overall, I am satisfied with how easy it is to use this system.	3	2	1	1	1	1	1	1	1	1	1.3
It was simple to use this system.	1	2	1	1	1	1	1	1	1	1	1.1
I was able to complete the tasks and scenarios quickly using this system.	1	1	1	1	1	2	1	2	1	1	1.2
I felt comfortable using this system.	1	2	1	1	1	1	1	1	1	1	1.1
It was easy to learn to use this system.	1	2	1	1	1	1	1	1	1	1	1.1
I believe I could become productive quickly using this system.	2	1	1	1	1	1	1	1	1	1	1.1
The system gave error messages that clearly told me how to fix problems.	4	4	4	4	4	4	4	4	4	1	3.7
Whenever I made a mistake using the system, I could recover easily and quickly.	3	4	1	4	1	4	3	4	4	1	2.9
The information (such as online help, on-screen messages, and other documentation) provided with this system was clear.	1	1	1	1	1	1	3	1	1	1	1.2
It was easy to find the information I needed.	1	1	1	1	1	1	2	1	1	1	1.1
The information was effective in helping me complete the tasks and scenarios.	1	1	1	1	1	1	1	1	1	1	1
The organization of information on the system screens was clear.	1	3	1	1	1	1	1	1	1	1	1.2
The interface of this system was pleasant.	2	4	2	1	1	1	1	1	1	1	1.5
I liked using the interface of this system.	2	4	2	1	1	1	1	1	1	1	1.5
This system has all the functions and capabilities I expect it to have.	4	2	2	1	1	2	3	2	1	1	1.9
Overall, I am satisfied with this system.	2	2	1	1	1	1	2	1	1	1	1.3

Appendix G
User Evaluation Survey Questions:

Demographic Questions:

1. Please enter your name below:
2. Please enter your email address below:
3. Please enter your age below (or "Prefer not to say"):
4. Please enter your highest education level below (ex: Bachelor's Degree, Master's Degree, etc., or "Prefer not to say"):
5. Please enter what sex were you assigned at birth, on your original birth certificate? (or "Prefer not to say"):
6. Please select the most appropriate choice for your ethnicity (self-identified, check all that apply or "Prefer not to say"):
7. On a scale of 1 to 5, what would you rank your familiarity with genomics/genetics research or analyses?

SUS Questions:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

PSSUQ Questions:

1. Overall, I am satisfied with how easy it is to use this system.
2. It was simple to use this system.
3. I was able to complete the tasks and scenarios quickly using this system.
4. I felt comfortable using this system.
5. It was easy to learn to use this system.
6. I believe I could become productive quickly using this system.
7. The system gave error messages that clearly told me how to fix problems.
8. Whenever I made a mistake using the system, I could recover easily and quickly.
9. The information (such as online help, on-screen messages, and other documentation) provided with this system was clear.
10. It was easy to find the information I needed.
11. The information was effective in helping me complete the tasks and scenarios.
12. The organization of information on the system screens was clear.
13. The interface of this system was pleasant.

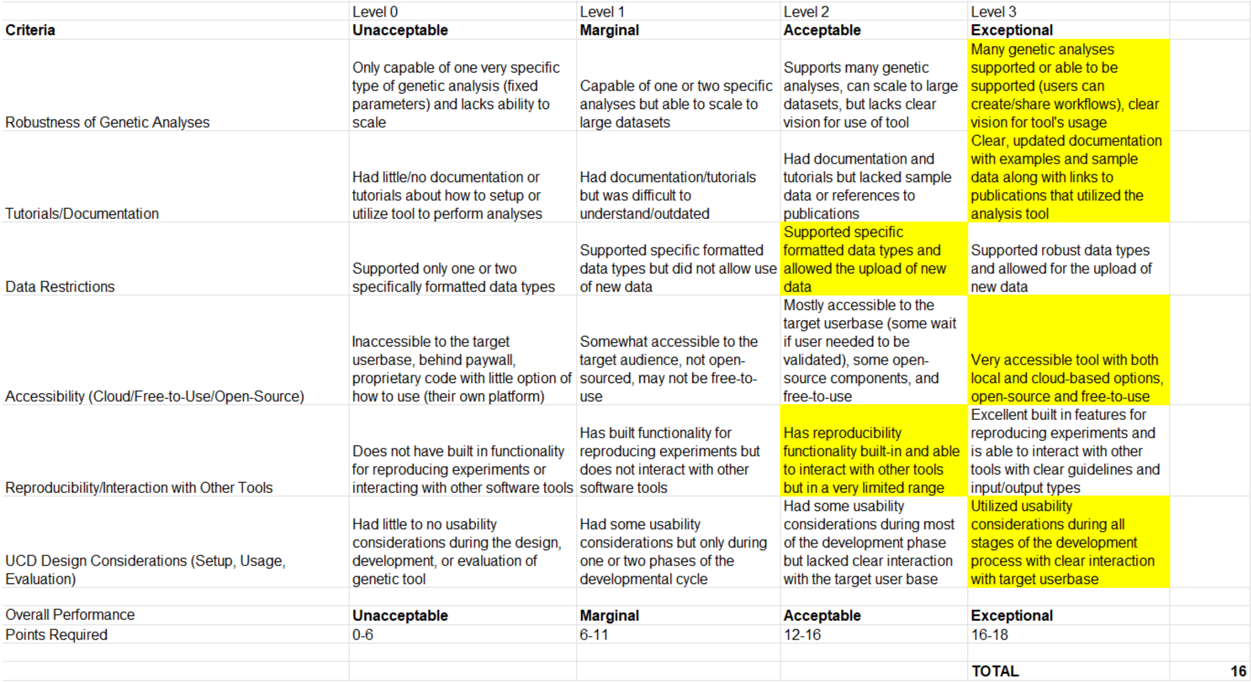
14. I liked using the interface of this system.
15. This system has all the functions and capabilities I expect it to have.
16. Overall, I am satisfied with this system.

Free-response Questions:

1. What are some of the positives you noticed when utilizing UF-GWA ?
2. What are some of the negatives/challenges you noticed when utilizing UF-GWAS?
3. Do you have any suggestions/improvements for UF-GWAS?
4. Any additional comments about your experience using UF-GWAS that were not captured during this survey?

Appendix H

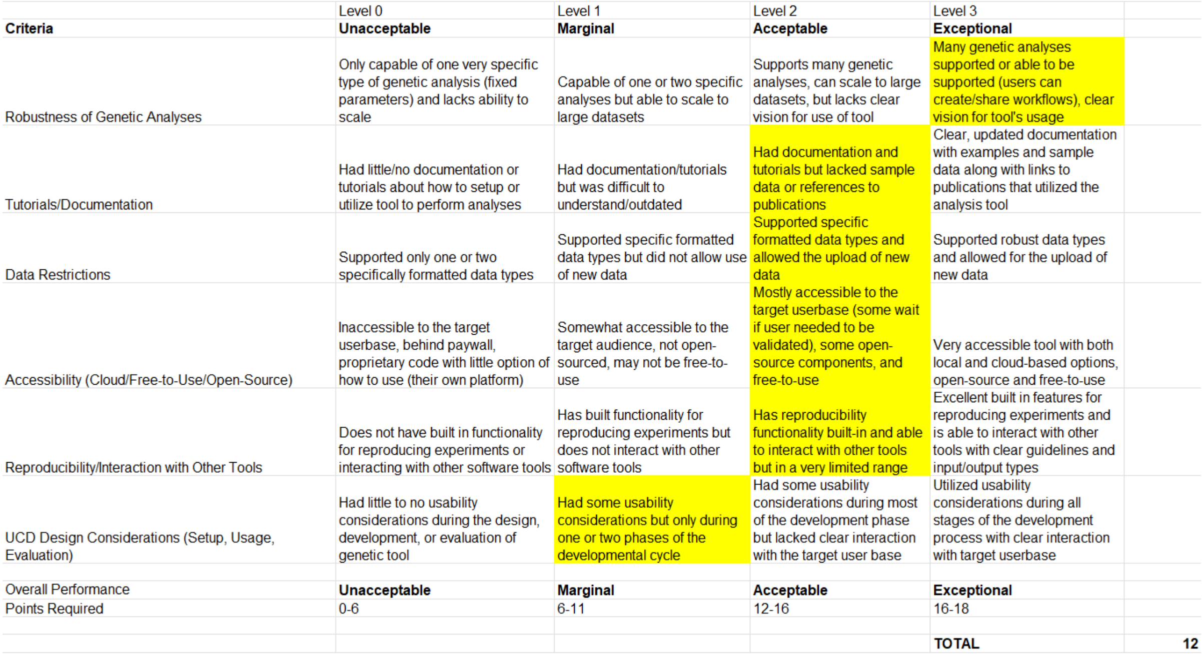
Evaluation Rubric applied to UF-GWAS:



	Level 0	Level 1	Level 2	Level 3	
Criteria	**Unacceptable**	**Marginal**	**Acceptable**	**Exceptional**	
Robustness of Genetic Analyses	Only capable of one very specific type of genetic analysis (fixed parameters) and lacks ability to scale	Capable of one or two specific analyses but able to scale to large datasets	Supports many genetic analyses, can scale to large datasets, but lacks clear vision for use of tool	Many genetic analyses supported or able to be supported (users can create/share workflows), clear vision for tool's usage	
Tutorials/Documentation	Had little/no documentation or tutorials about how to setup or utilize tool to perform analyses	Had documentation/tutorials but was difficult to understand/outdated	Had documentation and tutorials but lacked sample data or references to publications	Clear, updated documentation with examples and sample data along with links to publications that utilized the analysis tool	
Data Restrictions	Supported only one or two specifically formatted data types	Supported specific formatted data types but did not allow use of new data	Supported specific formatted data types and allowed the upload of new data	Supported robust data types and allowed for the upload of new data	
Accessibility (Cloud/Free-to-Use/Open-Source)	Inaccessible to the target userbase, behind paywall, proprietary code with little option of how to use (their own platform)	Somewhat accessible to the target audience, not open-sourced, may not be free-to-use	Mostly accessible to the target userbase (some wait if user needed to be validated), some open-source components, and free-to-use	Very accessible tool with both local and cloud-based options, open-source and free-to-use	
Reproducibility/Interaction with Other Tools	Does not have built in functionality for reproducing experiments or interacting with other software tools	Has built functionality for reproducing experiments but does not interact with other software tools	Has reproducibility functionality built-in and able to interact with other tools but in a very limited range	Excellent built in features for reproducing experiments and is able to interact with other tools with clear guidelines and input/output types	
UCD Design Considerations (Setup, Usage, Evaluation)	Had little to no usability considerations during the design, development, or evaluation of genetic tool	Had some usability considerations but only during one or two phases of the developmental cycle	Had some usability considerations during most of the development phase but lacked clear interaction with the target user base	Utilized usability considerations during all stages of the development process with clear interaction with target userbase	
Overall Performance	**Unacceptable**	**Marginal**	**Acceptable**	**Exceptional**	
Points Required	0-6	6-11	12-16	16-18	
				TOTAL	**16**

Reasons for Acceptable Category: Data restrictions are limited to specific file types and do not fully allow for all phenotype files to be accepted. Although reproducibility is excellent in UF-GWAS, it has limited interaction with other bioinformatics tools.

Evaluation Rubric applied to PLINK:



	Level 0	Level 1	Level 2	Level 3	
Criteria	**Unacceptable**	**Marginal**	**Acceptable**	**Exceptional**	
Robustness of Genetic Analyses	Only capable of one very specific type of genetic analysis (fixed parameters) and lacks ability to scale	Capable of one or two specific analyses but able to scale to large datasets	Supports many genetic analyses, can scale to large datasets, but lacks clear vision for use of tool	Many genetic analyses supported or able to be supported (users can create/share workflows), clear vision for tool's usage	
Tutorials/Documentation	Had little/no documentation or tutorials about how to setup or utilize tool to perform analyses	Had documentation/tutorials but was difficult to understand/outdated	Had documentation and tutorials but lacked sample data or references to publications	Clear, updated documentation with examples and sample data along with links to publications that utilized the analysis tool	
Data Restrictions	Supported only one or two specifically formatted data types	Supported specific formatted data types but did not allow use of new data	Supported specific formatted data types and allowed the upload of new data	Supported robust data types and allowed for the upload of new data	
Accessibility (Cloud/Free-to-Use/Open-Source)	Inaccessible to the target userbase, behind paywall, proprietary code with little option of how to use (their own platform)	Somewhat accessible to the target audience, not open-sourced, may not be free-to-use	Mostly accessible to the target userbase (some wait if user needed to be validated), some open-source components, and free-to-use	Very accessible tool with both local and cloud-based options, open-source and free-to-use	
Reproducibility/Interaction with Other Tools	Does not have built in functionality for reproducing experiments or interacting with other software tools	Has built functionality for reproducing experiments but does not interact with other software tools	Has reproducibility functionality built-in and able to interact with other tools but in a very limited range	Excellent built in features for reproducing experiments and is able to interact with other tools with clear guidelines and input/output types	
UCD Design Considerations (Setup, Usage, Evaluation)	Had little to no usability considerations during the design, development, or evaluation of genetic tool	Had some usability considerations but only during one or two phases of the developmental cycle	Had some usability considerations during most of the development phase but lacked clear interaction with the target user base	Utilized usability considerations during all stages of the development process with clear interaction with target userbase	
Overall Performance	**Unacceptable**	**Marginal**	**Acceptable**	**Exceptional**	
Points Required	0-6	6-11	12-16	16-18	
				TOTAL	**12**

Reasons for Acceptable/Marginal Category: Tutorial/Documentation not updated and no sample data provided for learning. Data restrictions are limited to specific file types and do not fully allow for all phenotype files to be accepted. Only offline options readily available for accessibility. Although reproducibility is excellent in PLINK, it has limited interaction with other bioinformatics tools. In terms of UCD, very little is mentioned/utilized in PLINK documentation or related publications about any user-centered considerations.

REFERENCES

1. Stephens, Z.D., Lee, S.Y., Faghri, F., Campbell, R.H., Zhai, C., Efron, M.J., Iyer, R., Schatz, M.C., Sinha, S., Robinson, G.E.: Big Data: Astronomical or Genomical? PLoS Biol. 13, e1002195 (2015)
2. Human Factors in the Design and Evaluation of Bioinformatics Tools. Procedia Manufacturing. 3, 2003–2010 (2015)
3. Initial sequencing and analysis of the human genome. Nature. 409, 860–921 (2001)
4. Blell, M., Hunter, M.A.: Direct-to-Consumer Genetic Testing's Red Herring: "Genetic Ancestry" and Personalized Medicine. Front. Med. 0, (2019). https://doi.org/10.3389/fmed.2019.00048
5. Balaji, D., Terry, S.F.: Benefits and Risks of Sharing Genomic Information. Genet. Test. Mol. Biomarkers. 19, 648–649 (2015)
6. Clément, L., Emeric, D., J, Gonzalez Bruno, Laurent, M., David, L., Eivind, H., Kristian, V.: A data-supported history of bioinformatics tools. (2018). https://doi.org/10.48550/ARXIV.1807.06808
7. Terra, https://app.terra.bio/
8. All of Us Research Program, https://allofus.nih.gov/future-health-begins-all-us
9. Pavelin, K., Cham, J.A., de Matos, P., Brooksbank, C., Cameron, G., Steinbeck, C.: Bioinformatics Meets User-Centred Design: A Perspective. PLoS Comput. Biol. 8, e1002554 (2012)
10. de Matos, P., Cham, J.A., Cao, H., Alcántara, R., Rowland, F., Lopez, R., Steinbeck, C.: The Enzyme Portal: a case study in applying user-centred design methods in bioinformatics. BMC Bioinformatics. 14, 103 (2013)
11. Recchia, G., Chiappi, A., Chandratillake, G., Raymond, L., Freeman, A.L.J.: Creating genetic reports that are understood by nonspecialists: a case study. Genet. Med. 22, 353–361 (2020)
12. Alberto, Iñiguez-Jarrín, C., Lopez, O., Gonzalez-Ibea, D., Pérez-Román, E., Borredà, C., Terol, J., Ibanez, V., Talón, M.: Applying user centred design to improve the design of genomic user interfaces. In: Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering. SCITEPRESS - Science and Technology Publications (2021)

13. Bolchini, D., Finkelstein, A., Perrone, V., Nagl, S.: Better bioinformatics through usability analysis, http://dx.doi.org/10.1093/bioinformatics/btn633, (2009)
14. Funding for Research Training, https://www.genome.gov/careers-training/NHGRI-Funding-Opportunities-Training-Career-Development
15. Norman, D.A., Draper, S.W.: User Centered System Design: New Perspectives on Human-computer Interaction. Hillsdale, N.J. : L. Erlbaum Associates (1986)
16. Witteman, H.O., Dansokho, S.C., Colquhoun, H., Coulter, A., Dugas, M., Fagerlin, A., Giguere, A.M., Glouberman, S., Haslett, L., Hoffman, A., Ivers, N., Légaré, F., Légaré, J., Levin, C., Lopez, K., Montori, V.M., Provencher, T., Renaud, J.-S., Sparling, K., Stacey, D., Vaisson, G., Volk, R.J., Witteman, W.: User-centered design and the development of patient decision aids: protocol for a systematic review. Syst. Rev. 4, 11 (2015)
17. Wolpin, S., Stewart, M.: A deliberate and rigorous approach to development of patient-centered technologies. Semin. Oncol. Nurs. 27, 183–191 (2011)
18. Clément, L., Emeric, D., J, Gonzalez Bruno, Laurent, M., David, L., Eivind, H., Kristian, V.: A data-supported history of bioinformatics tools. (2018). https://doi.org/10.48550/arXiv.1807.06808
19. Terra, https://app.terra.bio/
20. All of Us Research Program, https://allofus.nih.gov/future-health-begins-all-us
21. [No title], https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en
22. Bazzano, A.N., Martin, J., Hicks, E., Faughnan, M., Murphy, L.: Human-centred design in global health: A scoping review of applications and contexts. PLoS One. 12, e0186744 (2017)
23. Göttgens, I., Oertelt-Prigione, S.: The Application of Human-Centered Design Approaches in Health Research and Innovation: A Narrative Review of Current Practices. JMIR mHealth and uHealth. 9, (2021). https://doi.org/10.2196/28102
24. Carpenter, A.E., Kamentsky, L., Eliceiri, K.W.: A Call for Bioimaging Software Usability. Nat. Methods. 9, 666
25. Mangul, S., Martin, L.S., Eskin, E., Blekhman, R.: Improving the usability and archival stability of bioinformatics software. Genome Biol. 20, (2019). https://doi.org/10.1186/s13059-019-1649-8

26. 2020 NHGRI Strategic Vision, https://www.genome.gov/2020SV
27. Funding for Research Training, https://www.genome.gov/careers-training/NHGRI-Funding-Opportunities-Training-Career-Development
28. Attwood, T.K., Blackford, S., Brazas, M.D., Davies, A., Schneider, M.V.: A global perspective on evolving bioinformatics and data science training needs. Brief. Bioinform. 20, 398–404 (2017)
29. Machado, S., Paixão-Cortes, W.R., de Souza, N.: Decision-Making for Interactive Systems: A Case Study for Teaching and Learning in Bioinformatics. Learning and Collaboration Technologies. Technology in Education. 90–109 (2017)
30. Murphy, S.N., Gainer, V., Chueh, H.C.: A visual interface designed for novice users to find research patient cohorts in a large biomedical database. AMIA Annu. Symp. Proc. 2003, (2003)
31. Britto, M.T., Jimison, H.B., Munafo, J.K., Wissman, J., Rogers, M.L., Hersh, W.: Usability Testing Finds Problems for Novice Users of Pediatric Portals. J. Am. Med. Inform. Assoc. 16, 660–669 (2009)
32. Zhang, L., Yang, X., Cota, Z., Cui, H., Ford, B., Chen, H.-L., Macklin, J.A., Reznicek, A., Starr, J.: Which methods are the most effective in enabling novice users to participate in ontology creation? A usability study. Database . 2021, baab035 (2021)
33. Harris, J.D., Quatman, C.E., Manring, M.M., Siston, R.A., Flanigan, D.C.: How to write a systematic review. Am. J. Sports Med. 42, (2014). https://doi.org/10.1177/0363546513497567
34. [No title], https://www.mendeley.com/reference-management/reference-manager
35. Vasileiou, K., Barnett, J., Thorpe, S., Young, T.: Characterising and justifying sample size sufficiency in interview-based studies: systematic analysis of qualitative health research over a 15-year period. BMC Med. Res. Methodol. 18, 1–18 (2018)
36. Braun, V., Clarke, V.: Successful Qualitative Research: A Practical Guide for Beginners. SAGE (2013)
37. DeJonckheere, M., Vaughn, L.M.: Semistructured interviewing in primary care research: a balance of relationship and rigour. Family Medicine and Community Health. 7, e000057 (2019)

38. Scribe, H.: Happy Scribe: Audio Transcription & Video Subtitles, https://www.happyscribe.com/
39. ATLAS.ti, https://atlasti.com
40. Crabtree, B.F., Miller, W.L., Swenson, M.M.: Doing Qualitative Research, http://dx.doi.org/10.1097/00006199-199507000-00011, (1995)
41. Krippendorff, K.: Computing Krippendorff's Alpha-Reliability. (2011)
42. Krippendorff, K.: Content Analysis: An Introduction to Its Methodology. SAGE (2004)
43. Scriven, M.: Evaluation Thesaurus. SAGE (1991)
44. Davidson, E.: Evaluation Methodology Basics: The Nuts and Bolts of Sound Evaluation, http://dx.doi.org/10.4135/9781452230115, (2005)
45. Rubrics, https://www.betterevaluation.org/en/evaluation-options/rubrics
46. Xie, C., Jauhari, S., Mora, A.: Popularity and performance of bioinformatics software: the case of gene set analysis. BMC Bioinformatics. 22, 191 (2021)
47. Tractenberg, R.E., Lindvall, J.M., Attwood, T.K., Via, A.: The Mastery Rubric for Bioinformatics: A tool to support design and evaluation of career-spanning education and training. PLoS One. 14, e0225256 (2019)
48. Markiewicz, A., Patrick, I.: Developing Monitoring and Evaluation Frameworks, http://dx.doi.org/10.4135/9781071878774, (2016)
49. Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P., Moher, D.: [The PRISMA 2020 statement: an updated guideline for reporting systematic reviewsDeclaración PRISMA 2020: una guía actualizada para la publicación de revisiones sistemáticas]. Rev. Panam. Salud Publica. 46, e112 (2022)
50. Baker, M.: 1,500 scientists lift the lid on reproducibility. Nature. 533, 452–454 (2016)
51. National Academies of Sciences, Engineering, and Medicine, Policy and Global Affairs, Committee on Science, Engineering, Medicine, and Public Policy, Board on Research Data and Information, Division on Engineering and Physical Sciences, Committee on Applied and Theoretical Statistics, Board on Mathematical Sciences and Analytics, Division on Earth and Life Studies, Nuclear and Radiation Studies Board, Division of Behavioral and

Social Sciences and Education, Committee on National Statistics, Board on Behavioral, Cognitive, and Sensory Sciences, Committee on Reproducibility and Replicability in Science: Reproducibility and Replicability in Science. National Academies Press (2019)

52. Hutter, C.M.: Genome-Wide Association Studies (GWAS), https://www.genome.gov/genetics-glossary/Genome-Wide-Association-Studies
53. Watanabe, K., Stringer, S., Frei, O., Umićević Mirkov, M., de Leeuw, C., Polderman, T.J.C., van der Sluis, S., Andreassen, O.A., Neale, B.M., Posthuma, D.: A global overview of pleiotropy and genetic architecture in complex traits. Nat. Genet. 51, 1339–1348 (2019)
54. Visscher, P.M., Wray, N.R., Zhang, Q., Sklar, P., McCarthy, M.I., Brown, M.A., Yang, J.: 10 Years of GWAS Discovery: Biology, Function, and Translation. Am. J. Hum. Genet. 101, 5 (2017)
55. Uffelmann, E., Huang, Q.Q., Munung, N.S., de Vries, J., Okada, Y., Martin, A.R., Martin, H.C., Lappalainen, T.: Genome-wide association studies. Nature Reviews Methods Primers. 1, 1–21 (2021)
56. Tam, V., Patel, N., Turcotte, M., Bossé, Y., Paré, G., Meyre, D.: Benefits and limitations of genome-wide association studies. Nat. Rev. Genet. 20, 467–484 (2019)
57. 2020 NHGRI Strategic Vision, https://www.genome.gov/2020SV
58. Genomic Analysis, Visualization and Informatics Lab-space (AnVIL), https://www.genome.gov/Funded-Programs-Projects/Computational-Genomics-and-Data-Science-Program/Genomic-Analysis-Visualization-Informatics-Lab-space-AnVIL
59. What is AnVIL?, https://anvilproject.org/overview
60. All of Us Research Program, https://allofus.nih.gov/future-health-begins-all-us
61. Mell, P.M., Grance, T.: The NIST definition of cloud computing, http://dx.doi.org/10.6028/nist.sp.800-145, (2011)
62. Golightly, L., Chang, V., Xu, Q.A., Gao, X., Liu, B.S.C.: Adoption of cloud computing as innovation in the organization, http://dx.doi.org/10.1177/18479790221093992, (2022)
63. Foote, K.D.: A Brief History of Cloud Computing, https://www.dataversity.net/brief-history-cloud-computing/
64. Langmead, B., Nellore, A.: Cloud computing for genomic data analysis and collaboration. Nat. Rev. Genet. 19, 325 (2018)
65. Kent, W.J., James Kent, W., Sugnet, C.W., Furey, T.S., Roskin, K.M., Pringle, T.H., Zahler,

A.M., Haussler, A.D.: The Human Genome Browser at UCSC, http://dx.doi.org/10.1101/gr.229102, (2002)

66. Heath, A.P., Greenway, M., Powell, R., Spring, J., Suarez, R., Hanley, D., Bandlamudi, C., McNerney, M.E., White, K.P., Grossman, R.L.: Bionimbus: a cloud for managing, analyzing and sharing large genomics datasets. J. Am. Med. Inform. Assoc. 21, 969–975 (2014)
67. Grossman, R.L.: Data Lakes, Clouds, and Commons: A Review of Platforms for Analyzing and Sharing Genomic Data, http://dx.doi.org/10.1016/j.tig.2018.12.006, (2019)
68. Reynolds, S.M., Miller, M., Lee, P., Leinonen, K., Paquette, S.M., Rodebaugh, Z., Hahn, A., Gibbs, D.L., Slagel, J., Longabaugh, W.J., Dhankani, V., Reyes, M., Pihl, T., Backus, M., Bookman, M., Deflaux, N., Bingham, J., Pot, D., Shmulevich, I.: The ISB Cancer Genomics Cloud: A Flexible Cloud-Based Platform for Cancer Genomics Research, http://dx.doi.org/10.1158/0008-5472.can-17-0617, (2017)
69. Birger, C., Hanna, M., Salinas, E., Neff, J., Saksena, G., Livitz, D., Rosebrock, D., Stewart, C., Leshchiner, I., Baumann, A., Voet, D., Cibulskis, K., Banks, E., Philippakis, A., Getz, G.: FireCloud, a scalable cloud-based platform for collaborative genome analysis: Strategies for reducing and controlling costs, http://dx.doi.org/10.1101/209494
70. Lau, J.W., Lehnert, E., Sethi, A., Malhotra, R., Kaushik, G., Onder, Z., Groves-Kirkby, N., Mihajlovic, A., DiGiovanna, J., Srdic, M., Bajcic, D., Radenkovic, J., Mladenovic, V., Krstanovic, D., Arsenijevic, V., Klisic, D., Mitrovic, M., Bogicevic, I., Kural, D., Davis-Dusenbery, B.: The Cancer Genomics Cloud: Collaborative, Reproducible, and Democratized—A New Paradigm in Large-Scale Computational Research, http://dx.doi.org/10.1158/0008-5472.can-17-0387, (2017)
71. Grossman, R.L., Heath, A.P., Ferretti, V., Varmus, H.E., Lowy, D.R., Kibbe, W.A., Staudt, L.M.: Toward a Shared Vision for Cancer Genomic Data, http://dx.doi.org/10.1056/nejmp1607591, (2016)
72. Charlebois, K., Palmour, N., Knoppers, B.M.: The Adoption of Cloud Computing in the Field of Genomics Research: The Influence of Ethical and Legal Issues. PLoS One. 11, e0164347 (2016)
73. Prasanna, A., Pooja, R., Suchithra, V., Ravikumar, A., Niranjan, V.: Cloud Based Solutions for Genome Informatics: Challenges and Applications,

http://dx.doi.org/10.1016/j.matpr.2017.12.340, (2018)

74. Leu, M.G., Weinberg, S.T., Monsen, C., Lehmann, C.U.: Web Services and Cloud Computing in Pediatric Care. Pediatrics. 148, (2021). https://doi.org/10.1542/peds.2021-052048
75. Wang, S.L., Chiang, I.E., Kuo, A., Lin, H.I.: Design and Usability Evaluation of Mobile Cloud Healthcare System for Diabetes Prevention. Stud. Health Technol. Inform. 257, (2019)
76. Kolhar, M., Alameen, A.: Before Teaching Cybersecurity in the University: Usability Analysis of the Cloud Platform with Learner's Perspective, https://www.proquest.com/openview/2d9920f5ca2303bc8718551e996e52d3/1?pq-origsite=gscholar&cbl=2046213
77. Pruitt, J., Adlin, T.: The Persona Lifecycle: Keeping People in Mind Throughout Product Design. Elsevier (2010)
78. User profiles and personas in the design and development of consumer health technologies. Int. J. Med. Inform. 82, e251–e268 (2013)
79. Patient decision-making personas: An application of a patient-centered cognitive task analysis (P-CTA). Appl. Ergon. 87, 103107 (2020)
80. Assistant Secretary for Public Affairs: Personas. (2013)
81. (3), J.R.: Employee personas and how to create them, https://www.steptwo.com.au/papers/kmc_personas/
82. Project Jupyter, https://jupyter.org/
83. Jupyter Lab, https://docs.jupyter.org/en/latest/
84. Grüning, B.A., Rasche, E., Rebolledo-Jaramillo, B., Eberhard, C., Houwaart, T., Chilton, J., Coraor, N., Backofen, R., Taylor, J., Nekrutenko, A.: Jupyter and Galaxy: Easing entry barriers into complex data analyses for biomedical researchers. PLoS Comput. Biol. 13, (2017). https://doi.org/10.1371/journal.pcbi.1005425
85. Pimentel, J.F., Murta, L., Braganholo, V., Freire, J.: Understanding and improving the quality and reproducibility of Jupyter notebooks. Empirical Software Engineering. 26, (2021). https://doi.org/10.1007/s10664-021-09961-9
86. Rule, A., Birmingham, A., Zuniga, C., Altintas, I., Huang, S.-C., Knight, R., Moshiri, N., Nguyen, M.H., Rosenthal, S.B., Pérez, F., Rose, P.W.: Ten simple rules for writing and

sharing computational analyses in Jupyter Notebooks. PLoS Comput. Biol. 15, (2019). https://doi.org/10.1371/journal.pcbi.1007007

87. Hail | Hail 0.2, https://hail.is/docs/0.2/index.html#
88. Atkinson, E.G., Maihofer, A.X., Kanai, M., Martin, A.R., Karczewski, K.J., Santoro, M.L., Ulirsch, J.C., Kamatani, Y., Okada, Y., Finucane, H.K., Koenen, K.C., Nievergelt, C.M., Daly, M.J., Neale, B.M.: Tractor uses local ancestry to enable the inclusion of admixed individuals in GWAS and to boost power. Nat. Genet. 53, 195–204 (2021)
89. Karaca, M., Atceken, N., Karaca, Ş., Civelek, E., Şekerel, B.E., Polimanti, R.: Phenotypic and Molecular Characterization of Risk Loci Associated With Asthma and Lung Function. Allergy Asthma Immunol. Res. 12, 806–820 (2020)
90. De Lillo, A., De Angelis, F., Di Girolamo, M., Luigetti, M., Frusconi, S., Manfellotto, D., Fuciarelli, M., Polimanti, R.: Phenome-wide association study of TTR and RBP4 genes in 361,194 individuals reveals novel insights in the genetics of hereditary and wildtype transthyretin amyloidoses. Hum. Genet. 138, (2019). https://doi.org/10.1007/s00439-019-02078-6
91. Krissaane, I., De Niz, C., Gutiérrez-Sacristán, A., Korodi, G., Ede, N., Kumar, R., Lyons, J., Manrai, A., Patel, C., Kohane, I., Avillach, P.: Scalability and cost-effectiveness analysis of whole genome-wide association studies on Google Cloud Platform and Amazon Web Services. J. Am. Med. Inform. Assoc. 27, 1425–1430 (2020)
92. IBM Cloud Education: containerization, https://www.ibm.com/cloud/learn/containerization
93. Ratliff, J.: What is a Container?, https://www.docker.com/resources/what-container/
94. Kim, B.S., Lee, S.H., Lee, Y.R., Park, Y.H., Jeong, J.: Design and Implementation of Cloud Docker Application Architecture Based on Machine Learning in Container Management for Smart Manufacturing. NATO Adv. Sci. Inst. Ser. E Appl. Sci. 12, 6737 (2022)
95. Jackman, S.D., Mozgacheva, T., Chen, S., O'Huiginn, B., Bailey, L., Birol, I., Jones, S.J.M.: ORCA: a comprehensive bioinformatics container environment for education and research. Bioinformatics. 35, 4448–4450 (2019)
96. Menegidio, F.B., Barbosa, D.A., dos S Gonçalves, R., Nishime, M.M., Jabes, D.L., de Oliveira, R.C., Nunes, L.R.: Bioportainer Workbench: a versatile and user-friendly system that integrates implementation, management, and use of bioinformatics resources in Docker environments. Gigascience. 8, (2019). https://doi.org/10.1093/gigascience/giz041

97. Stanaway, I.B., Hall, T.O., Rosenthal, E.A., Palmer, M., Naranbhai, V., Knevel, R., Namjou-Khales, B., Carroll, R.J., Kiryluk, K., Gordon, A.S., Linder, J., Howell, K.M., Mapes, B.M., Lin, F.T.J., Joo, Y.Y., Hayes, M.G., Gharavi, A.G., Pendergrass, S.A., Ritchie, M.D., de Andrade, M., Croteau-Chonka, D.C., Raychaudhuri, S., Weiss, S.T., Lebo, M., Amr, S.S., Carrell, D., Larson, E.B., Chute, C.G., Rasmussen-Torvik, L.J., Roy-Puckelwartz, M.J., Sleiman, P., Hakonarson, H., Li, R., Karlson, E.W., Peterson, J.F., Kullo, I.J., Chisholm, R., Denny, J.C., Jarvik, G.P., eMERGE Network, Crosslin, D.R.: The eMERGE genotype set of 83,717 subjects imputed to ~40 million variants genome wide and association with the herpes zoster medical record phenotype. Genet. Epidemiol. 43, 63–81 (2019)
98. Balsells, E., Shi, T., Leese, C., Lyell, I., Burrows, J., Wiuff, C., Campbell, H., Kyaw, M.H., Nair, H.: Global burden of Clostridium difficile infections: a systematic review and meta-analysis, http://dx.doi.org/10.7189/jogh.09.010407, (2019)
99. Coia, J.E., Kuijper, E.J.: The ESCMID Study Group for Clostridium difficile: History, Role and Perspectives, http://dx.doi.org/10.1007/978-3-319-72799-8_14, (2018)
100. Ferar, K.D.: Deriving a sociotechnical model for discovery in genomics-enabled learning health systems, http://hdl.handle.net/1773/49582, (2023)
101. Gonzales, S., O'Keefe, L., Gutzman, K., Viger, G., Wescott, A.B., Farrow, B., Heath, A.P., Kim, M.C., Taylor, D., Champieux, R., Yen, P.-Y., Holmes, K.: Personas for the translational workforce. Journal of Clinical and Translational Science. 4, 286 (2020)
102. Leprevost, F. da V., da Veiga Leprevost, F., Barbosa, V.C., Francisco, E.L., Perez-Riverol, Y., Carvalho, P.C.: On best practices in the development of bioinformatics software, http://dx.doi.org/10.3389/fgene.2014.00199, (2014)
103. Practical estimation of cloud storage costs for clinical genomic data. Practical Laboratory Medicine. 21, e00168 (2020)
104. Teufel, A., Krupp, M., Weinmann, A., Galle, P.R.: Current bioinformatics tools in genomic biomedical research (Review). Int. J. Mol. Med. 17, 967–973 (2006)
105. Wren, J.D.: Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades. Bioinformatics. 32, 2686–2691 (2016)
106. Gardner, P.P., Paterson, J.M., McGimpsey, S., Ashari-Ghomi, F., Umu, S.U., Pawlik, A., Gavryushkin, A., Black, M.A.: Sustained software development, not number of citations or

journal choice, is indicative of accurate bioinformatic software. Genome Biol. 23, 56 (2022)

107. ISO 9241-210:2010, https://www.iso.org/standard/52075.html
108. ISO/IEC 25062:2006, https://www.iso.org/standard/43046.html
109. Moumane, K., Idri, A., Abran, A.: Usability evaluation of mobile applications using ISO 9241 and ISO 25062 standards. Springerplus. 5, 1–15 (2016)
110. Bauer-Negrini, G., Cordenonsi da Fonseca, G., Gottfried, C., Herbert, J.: Usability evaluation of circRNA identification tools: Development of a heuristic-based framework and analysis. Comput. Biol. Med. 147, 105785 (2022)
111. Shyr, C., Kushniruk, A., Wasserman, W.W.: Usability study of clinical exome analysis software: top lessons learned and recommendations. J. Biomed. Inform. 51, 129–136 (2014)
112. Duan, Y., Zhang, W., Cheng, Y., Shi, M., Xia, X.Q.: A systematic evaluation of bioinformatics tools for identification of long noncoding RNAs. RNA. 27, (2021). https://doi.org/10.1261/rna.074724.120
113. Silva, L.B., Jimenez, R.C., Blomberg, N., Oliveira, J.L.: General guidelines for biomedical software development. F1000Res. 6, (2017). https://doi.org/10.12688/f1000research.10750.2
114. Brazas, M.D., Lewitter, F., Schneider, M.V., van Gelder, C.W.G., Palagi, P.M.: A Quick Guide to Genomics and Bioinformatics Training for Clinical and Public Audiences. PLoS Comput. Biol. 10, (2014). https://doi.org/10.1371/journal.pcbi.1003510
115. Wilson Sayres, M.A., Hauser, C., Sierk, M., Robic, S., Rosenwald, A.G., Smith, T.M., Triplett, E.W., Williams, J.J., Dinsdale, E., Morgan, W.R., Burnette, J.M., III, Donovan, S.S., Drew, J.C., Elgin, S.C.R., Fowlks, E.R., Galindo-Gonzalez, S., Goodman, A.L., Grandgenett, N.F., Goller, C.C., Jungck, J.R., Newman, J.D., Pearson, W., Ryder, E.F., Tosado-Acevedo, R., Tapprich, W., Tobin, T.C., Toro-Martínez, A., Welch, L.R., Wright, R., Barone, L., Ebenbach, D., McWilliams, M., Olney, K.C., Pauley, M.A.: Bioinformatics core competencies for undergraduate life sciences education. PLoS One. 13, e0196878 (2018)
116. Friedman, C.P.: A “fundamental theorem” of biomedical informatics. J. Am. Med. Inform. Assoc. 16, 169–170 (2009)
117. 2020 NHGRI Strategic Vision, https://www.genome.gov/2020SV
118. Faulkner, L.: Beyond the five-user assumption: benefits of increased sample sizes in usability testing. Behav. Res. Methods Instrum. Comput. 35, 379–383 (2003)

119. Cazañas, A., de San Miguel, A., Parra, E.: Estimating Sample Size for Usability Testing. Enfoque UTE. 8, 172–185 (2017)
120. Six, J.M., Macefield, R.: How to Determine the Right Number of Participants for Usability Studies, https://www.uxmatters.com/mt/archives/2016/01/how-to-determine-the-right-number-of-participants-for-usability-studies.php
121. 1000 Genomes, https://www.internationalgenome.org/
122. Shah, C., White, R.W.: Task-Based Evaluation. Task Intelligence for Search and Recommendation. 75–98 (2021)
123. Will, T.: PSSUQ (Post-Study System Usability Questionnaire), https://uiuxtrend.com/pssuq-post-study-system-usability-questionnaire/
124. SUS: A "Quick and Dirty" Usability Scale, http://dx.doi.org/10.1201/9781498710411-35, (1996)
125. Assistant Secretary for Public Affairs: System Usability Scale (SUS). (2013)
126. PLINK 2.0, https://www.cog-genomics.org/plink/2.0/
127. Jaeschke, R., Jankowski, M., Brozek, J., Antonelli, M.: How to develop guidelines for clinical practice. Minerva Anestesiol. 75, (2009)
128. Woolf, S., Schünemann, H.J., Eccles, M.P., Grimshaw, J.M., Shekelle, P.: Developing clinical practice guidelines: types of evidence and outcomes; values and economics, synthesis, grading, and presentation and deriving recommendations. Implement. Sci. 7, 1–12 (2012)
129. Whittemore, R., Knafl, K.: The integrative review: updated methodology. J. Adv. Nurs. 52, (2005). https://doi.org/10.1111/j.1365-2648.2005.03621.x
130. Brooke, J. (1996) SUS—A Quick and Dirty Usability Scale. Usability Evaluation in Industry, 189, 4-7. - References - Scientific Research Publishing, https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=1664688
131. Sauro, J., Lewis, J.R.: Quantifying the User Experience: Practical Statistics for User Research. Elsevier Science (2016)
132. Kumuthini, J., Chimenti, M., Nahnsen, S., Peltzer, A., Meraba, R., McFadyen, R., Wells, G., Taylor, D., Maienschein-Cline, M., Li, J.-L., Thimmapuram, J., Murthy-Karuturi, R.,

Zass, L.: Ten simple rules for providing effective bioinformatics research support. PLoS Comput. Biol. 16, (2020). https://doi.org/10.1371/journal.pcbi.1007531

133. Lavappa, P.D., U.s. Department of Commerce National Institute of Standards and Technology, National Institute National Institute of Standards and Technology: Guidelines for Planning and Development of Software for Buildings and Building Systems. Createspace Independent Publishing Platform (2008)