Elucidating the Role and Use of Bioinformatics Software in Life Science Research

Sarah Morrison-Smith[1], Christina Boucher[1], Andrea Bunt[2], and Jaime Ruiz[1]

[1]Department of Computer Science
Colorado State University
Fort Collins, Colorado, USA
{sarahms,cboucher,jgruiz}@cs.colostate.edu

[2]Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
bunt@cs.umanitoba.ca

ABSTRACT

Life science research requires critical evaluation of data handling and analytical software usability. We present the results of semi-structured interviews which provide insight into the effects of bioinformatics software usability on life science research. Results from our study confirm much of the prior anecdotal evidence of standalone bioinformatics software usability. More importantly, we show that usability issues and life scientists' lack of expertise in applying computational methods to biological research is limiting their research objectives and contributing to researchers' reliance on computational experts to conduct their research.

CCS Concepts

•**Human-centered computing** → *Empirical studies in HCI;*

Keywords

Data-Driven Research, Qualitative Study, Bioinformatics Software

1. INTRODUCTION

Researchers in the life sciences, which we define as any field of scientific research that involves the study of living organisms (e.g. biology, microbiology, medicine, veterinarian medicine), are increasingly relying on computational tools to discover patterns, derive hypotheses, and develop conclusions [22]. These computational tools (also referred to as *bioinformatics tools*) are enabling significant biomedical breakthroughs, particularly those involving genomic data, such as the Human Genome Project [9]. Advances in the ability to create data have resulted in the quick and cheap generation of large amounts of data. As a result, analysis of data has become the bottleneck in genomic discovery [21] and life science researchers are increasingly relying on computational tools.

With the increasing popularity of integrating bioinformatics tools into their research workflow, anecdotal usability issues have been a recurrent topic of discussion in the life science community. Informal evidence indicates that bioinformatics software, such as web-based database search tools, are difficult to use and result in high user frustration [11, 24, 26]. Common problems range from difficult tool installation to struggling to efficiently link several tools together [11]. However, we still do not fully understand how life scientists conduct research and the extent to which bioinformatics tools are, or are not, supporting their work. Furthermore, a more focused investigation is warranted that will identify and characterize the factors that cause use of bioinformatics software to be a hindrance in life science research, including the influence of computational experience on scientists' ability to conduct work, the extent to which individual tasks and the overall workflow are or are not supported by existing tools, and any other usability breakdowns that have yet to be considered.

The goal of this work is to investigate the role that bioinformatics software plays in life science research. To reach this goal, we conducted a formalized study examining the use of bioinformatics software by life science researchers. Results from our study confirm much of the prior anecdotal evidence of standalone bioinformatics software usability. More importantly, we show that usability issues and life scientists' lack of expertise in applying computational methods to biological research is limiting their research objectives and contributing to researchers' reliance on computational experts to conduct their research. Our results also show that while the reliance on domain experts is resulting in collaborative research projects, the computational tools used in these projects not only do not facilitate collaboration, but make it more difficult to share data due to the size of the results returned.

The rest of this paper is organized as follows. First, we present some related work regarding bioinformatics software. Next, we describe our qualitative study investigating researchers' current work practices with bioinformatics software. This is followed by the presentation of our findings including an overview of researchers' workflow, goals, and challenges associated with using software in their research. Lastly, we discuss our findings in relation to research productivity, quality, and the need for collaboration.

British HCI 2015, July 13 - 17, 2015, Lincoln, United Kingdom
© 2015 ACM. ISBN 978-1-4503-3643-7/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2783446.2783581

2. RELATED WORK

2.1 Computational Workflow of Life Scientists

Given the recent shift to include computation in the life sciences, various aspects of conducting bioinformatics research have been topics of consideration. These topics include investigating the development and maintenance of bioinformatics software [17, 7]; understanding context-specific information retrieval, tasks, and workflows [18, 3, 28]; considering user needs in a related field, clinical translational science [6]; and developing tools to facilitate bioinformatics and other life science research [17, 12, 1, 2, 27, 29]. These studies provide valuable insight into the workings and challenges associated with bioinformatics research.

Investigations focusing on the development and improvement of computational pipelines [1, 2] and depiction of methods to improve the reproducibility of bioinformatics research [22] revealed that bioinformatics data analysis often includes using multiple software applications in succession to transform gigabytes or terabytes of raw data into concrete, comprehensible results [1, 2, 22]. These applications are typically strung together in a pipeline with custom scripts using the output of one tool as input for the next [1, 2, 22]. A pipeline may begin with enforcing quality control on raw data and then performing calculations, validations, and statistical analyses before displaying summarizations and visualizations of results [22]. While the contributions of Abouelhoda et al. [1, 2] and Preeyanon et al. [22] provide insight into the use and usability of these pipelines, neither shed light into the role and importance of pipeline software within life scientists' overall workflow. This is necessary to understand how they conduct research and determine the extent to which bioinformatics tools are supporting their work.

Although life scientists typically follow strict, written protocols when performing wet-lab experiments [27], research indicates that data analysis is less likely to be conducted while following a strict procedure [18]. Previous studies indicate that bioinformatics experts follow different home grown strategies when using data analysis applications [3, 28]. This is exacerbated by the observation that many life scientists do not keep a record of computational analysis procedures in the way they record wet-lab experiments [22, 28], which can cause researchers to lose track of the parameters used with analysis software and the order in which applications were ran [22]. These findings notwithstanding, no study reveals why life scientists are or are not documenting their workflow, what information is included in documentation, and the full extent to which the support or lack of support provided by existing tools affects documentation—although some tools have been created to assist in taking notes [27, 29].

Studies confirmed that life scientists possess a wide range of computational skills [7, 28], with a widespread lack of computer expertise leading to difficulties using bioinformatics software. They frequently do not understand the tools they are using and as a result, they make mistakes—such as failing to run programs on known test sets before use on actual data and using default parameters that may not be optimized for their data [22]. It has also been observed that they frequently lack awareness of existing analysis software [28] and fail to select the correct tools for their needs [22]. It is unsurprising, then, that many life scientists require programming experts to perform computational analyses on their data [17]. Unfortunately, while Tran et al. [28] examined the effect of inadequate computer proficiency on individual research tasks, and Chilana et al. [7] and Massar et al. [17] considered the ramifications of varying expertise on the process of developing bioinformatics software, no study has investigated the influence of computational experience on life scientists' overall workflow. In addition, no formal study has characterized the factors contributing to researchers' difficulties using bioinformatics software—which may include insufficient support provided by existing tools.

2.2 Analysis of Bioinformatics Software

Various software has been created to support bioinformatics research by facilitating the creation of pipelines to integrate tools [13, 11, 2, 14, 19], simplifying development of software for bioinformatics [17, 16, 25], and providing assistance for conducting wet-lab work and recording lab notes [27, 29]. However, more information is needed to determine which bioinformatics tasks are being ignored or insufficiently supported by existing tools, and how current tools support or fail to support researchers' overall workflow.

While developing these software tools, researchers identified a number of problems that have yet to be fully rectified. Although many existing tools make an effort to hide complexity and streamline use, software still frequently requires a high level of computing or programming experience to set up and use effectively [14, 20, 19, 11] which many bench life scientists' lack [20]. Tools commonly require a significant amount of effort to set up [14, 19, 11] and combine with other tools [3, 19, 13, 11]. Without this strong background in computing, users typically struggle with the command line user interface that the majority of these tools employ [22, 20]. It has been noted that users often wrestle with difficult to understand or missing documentation [19, 13] but lack support from the tool's creators [22, 19, 13]. Unfortunately, while these problems have been recognized by various researchers, none have been fully investigated to identify opportunities for improvement or the extent to which life scientists' workflows are affected. Lastly, we note that it is possible that additional problems have yet to be characterized since no study has been conducted with the specific purpose of identifying and evaluating usability issues concerning bioinformatics tools in the context of typical use.

3. METHODS

Our goal was to characterize the workflow of life scientists conducting research and examine how bioinformatics tools are supporting this workflow. We did so by identifying and evaluating specific issues concerning bioinformatics software in the context of typical use.

3.1 Interviews

A set of semi-structured interviews were conducted with ten researchers aged 31 to 46 ($\mu = 36.37$, $\sigma = 5.15$, two females) at a local university who have used bioinformatics software as part of their research activities. Participants were recruited via an interdepartmental email list. All participants had extensive education in the field of life science at the Ph.D. level or higher, and had first-hand experience using bioinformatics tools for at least one research project. The research area and title of each participant is presented in Table 1.

Table 1: Participant backgrounds.

Participant	Position	Department
P1	DVM[3]/Ph.D. Candidate	Microbiology/Veterinary Sciences
P2	Faculty	Plant Biology
P3	Postdoc Researcher	Plant Biology
P4	DVM[3]/Ph.D. Candidate	Clinical Sciences
P5	Faculty	Immunology & Microbiology
P6	Postdoc Researcher	Microbiology
P7	Postdoc Researcher	Immunology & Microbiology
P8	Faculty	Genetics
P9	Faculty	Biology
P10	Faculty	Epidemiology

The use of a semi-structured interview technique allowed us to cover additional topics as they were encountered, reducing the likelihood that important issues were overlooked [15]. Interviews took place at each of the participant's primary workspaces (offices or labs), providing opportunities for us to photograph their work environments. This also allowed us to review and photograph samples of relevant work materials. Interviews were approximately 45 to 60 minutes in duration and were recorded in audio format. In an effort to mitigate privacy concerns, interviewees were given an option to allow us to photograph their work provided that these photos were not disseminated, although one participant requested that photographs not be taken for reasons of privacy. All photographs collected during the interviews were taken at the request of the interviewer (as opposed to being at the participant's suggestion).

More generally, the participants were asked to educate us about their research practices and how they performed their daily work. Our interviews sought to answer the following interview questions:

- What is this researcher's goal? What is the product of his or her work?
- What characterizes the researcher's workflow? How is his or her work accomplished?
- What tools are used in problem solving, at what point during the work process are they used, and why?
- What types of tasks are best supported by the different tools and why?
- What preferences does the researcher have with respect to tools and media?

Lastly, participants were asked to walk through specific examples of conducting recent research in order to reduce recall bias and ground the interviews. We asked participants to discuss which computational method(s) were used to accomplish each specific task, and why they were chosen during this exercise.

3.2 Data Analysis

We analyzed participants' responses to the interview questions, the observations of software use, and the photographed work material by creating an affinity diagram as described in [4]. This construction revealed common themes in their work practices and research goals.

4. RESULTS

Participants' answers to interview questions and our examination of their software use provided insight into some of the key challenges that researchers working in this area face—both in terms of the specific computational tools they use and in finding access to, and coordinating with, collaborators. Before describing these challenges, we provide some context by summarizing what the researchers are trying to achieve, and how they go about doing so.

4.1 Participant Goals

In their research, our participants indicated that they seek to answer a variety of biological questions, which can be broadly categorized as follows:

Variation Discovery: identifying the presence of variation on the genetic (DNA) or transcriptomic (RNA) level for one or more species, and relating that variation to a particular trait.

Genome Assembly and/or Annotation: discovering the genetic make-up of a species; i.e., determining the sequence of nucleotides in the DNA of a species.

Evolutionary History: ascertaining how species have evolved or how they relate to each other.

Gene Regulation and/or Transcription: determining the conditions in which a gene is transcribed into RNA.

4.2 Workflows

Our participants described similar workflows despite diverse end goals. The following outlines our understanding of their workflow.

Collection. Organism(s) of the species of interest (e.g., plants, domestic cats, West-Nile virus, worms) are either captured or accumulated in their natural habitats and then brought to their research labs, or purchased from a laboratory supply company.

Extraction. Biological material (DNA or RNA) is extracted from the organisms over one or more time periods using wet laboratory methods.

Preparation. This biological material is then prepared for the next step by fragmenting the samples into smaller pieces. This is necessary since state-of-the-art instruments that accept biological material as input (DNA or RNA) and return the genetic sequence corresponding to that sample can only handle very small segments of biological material at one time. For example, genome sizes are typically within the range of 4.5 million to 3 billion nucleotides long, the instruments can only sequence between 100 and 150 nucleotides at a time (billions of small pieces are sequenced in parallel).

Sequencing. Prepared samples are then converted into the data for downstream analysis by way of *sequencing.* Since sequencing is typically performed off-site at a dedicated facility (referred to as a *sequencing core*), the physical samples have to be sent to the facility.

[3] Doctor of Veterinary Medicine

Transferring. Output from the sequencing stage is transferred from the sequencing facility to the participant's computing via SFTP, Windows Remote Desktop, and/or a portable hard drive. This output is a one ore more data files (FASTQ) that contains the sequence of nucleotides corresponding to each small fragment of biological material, and a quality score that corresponds to a likelihood that each nucleotide was read correctly by the machine. Hence, the FASTQ file(s) contains millions of short (100 in length) sequences of nucleotides (called *reads*) and their sequence of quality scores.

Analysis. The participants begin their analysis once they have their data files. In this stage, participants reported using bioinformatics software and methods for a range of purposes, including:

Genome assembly: building large contiguous regions (*contigs*) corresponding to the genome from the sequenced data.

Genome annotation: using the sequence data corresponding to RNA to identify the start and end of each gene in an assembled genomes.

RNA transcript assembly: using the sequence data corresponding to RNA to assemble or build the RNA transcripts.

Read alignment: aligning the sequence data to a reference genome or a draft genome.

SNP detection: determining all single-nucleotide polymorphisms (SNPs) for a genome or gene.

Large variant detection: determining all large genetic variations (e.g., a large deleted region in a gene) for a genome or gene.

The results of this bioinformatics analysis is sometimes visually inspected for correctness, and other times be validated through further wet-lab experiments.

4.3 Challenges

Our participants expressed a number of challenges when conducting analysis. Some of these frustrations relate to specific usability issues associated with the bioinformatics tools, confirming previous anecdotal reports [24, 26, 10], and others pertain to more complex issues, such as handling the data volume, finding the right collaborators, and coordinating large-scale collaborative projects.

4.3.1 Working with Big Data

The sequencing of the genetic material results in very large data sets (in the magnitude of 100s of GB for each sample). As a result, our participants often expressed issues transferring, working with, and visualizing data sets.

Transferring. Researchers typically used file transfer protocol (FTP/SFTP) software to transfer raw data from sequencing facilities over the internet. However, datasets frequently were too large for this method of transmission to be feasible. In these cases, as described by P2 and P5, data is often stored on a hard drive that physically mailed to the researchers:

> "Sometimes datasets get so big that it's actually faster to snail mail it on a hard disk than it is to transport it over the internet." (P2)

Storing. The size and number of files produced for a single experiment can be massive—up to terabytes in size. P7 describes the challenges associated with storing data of this size:

> "So we got to here and this is just a thing to try to save space on our server we don't have a lot of space so I am getting rid of sample files [. . .] I think we got 12 terabytes." (P7)

While a single experiment can result in large amounts of data, the issue is compounded by the fact that certain life science research requires the analysis of thousands of experiments.

Working with the data. The majority of bioinformatics tools are created for Linux servers with high performance computing facilities due to the volume of data. For example, state-of-the-art genome assemblers require up to 512GB of RAM to run on moderate-sized genomes. Often the participants had insufficient computational memory to run the software, which lead to slow processing times, software crashes, or inability to run.

> "I have problems with [the genome assembly software's] memory intensivity (sic)." (P4)

Even with reasonable computing resources, the size of the data can have a large impact on how long some bioinformatics software takes to run. For example, state-of-art genome assemblers will take three to four hours to run on a small genome (e.g. *E.coli*) but will require days or even weeks to run on a significantly larger genome (e.g. bonobo) [31]. The uncertainty in predicting when the program will complete or require further researcher input can be confusing and frustrating:

> ". . .sitting here and waiting for the next step, or knowing when it's done with that stuff so you can enter the next one can be a problem." (P1)

As a consequence, researchers periodically check if a program is done running, usually by manually checking whether or not the command line has returned to a prompt or by using a command such as "htop" to check the status of their program. Although notification via email is possible when researchers run programs on servers that employ a job queuing system, researchers must rely on the maintainers of the server to support this feature.

Visualizing. Participants also described relying on visual inspection to verify the output of their analysis and data, searching for specific data features (e.g., gaps) or taking a general, "bird's-eye" view of their data in order to determine accuracy:

> "Sometimes when I run this, it's just very clearly by eye not properly aligned." (P6)

This seems to be particularly important with large datasets, as indicated by P8:

> "So, being visual really makes a ton of difference for me. Especially with these big datasets. So the other software that are out there . . . they may put out a table that can be very difficult to read. The same data, the same information is there and you could reach the same research conclusion with the two tools, but one of them requires you to learn a lot more about how these things are done on the inside of the box whereas this one I like it a lot and I think it's good because it shows me directly what I need to see, visually." (P8)

In some instances, participants expressed that information was impossible to obtain without visualization:

> "I could not do this part without being able to visualize it because I have to see, at some point, the actual alignment." (P1)

4.3.2 Computational Tool Transparency

Due to the size of the data, most bioinformatics research is focused on translating a specific biological problem into an algorithmic framework for which efficient heuristics can be conceived of and implemented. The majority of parameters for bioinformatics software correspond to parameters in the underlying algorithmic problem. Unfortunately, our interviews revealed that setting these parameters in a principled way is a serious concern for our participants. Coming from life science backgrounds, our participants frequently described not understanding the usage of the parameters or how they should be set:

> "I don't necessarily know enough to make sure I'm picking our [settings correctly]." (P1)

Our participants reported using a variety of parameter setting strategies, including consulting the web, attempting to consult with experts (which we discuss further in the next section), using their own personal experience—such as extensive prior knowledge about the process, known rules of thumb, and educated guesses—or beginning with the default settings and using a trial-and-error approach. The description from P1 of how *"key parameters"* were incrementally adjusted in order to converge on an optimal analysis exemplifies the last strategy. This lack of transparency of the parameters, and how they should be adjusted, has two specific effects: they have a significant effect on the output of the programs [22], and they greatly affect software use. The participants' general discomfort with parameter setting was reported as a factor influencing their selection of tools, with participants expressing a preference for changing as few parameters as possible:

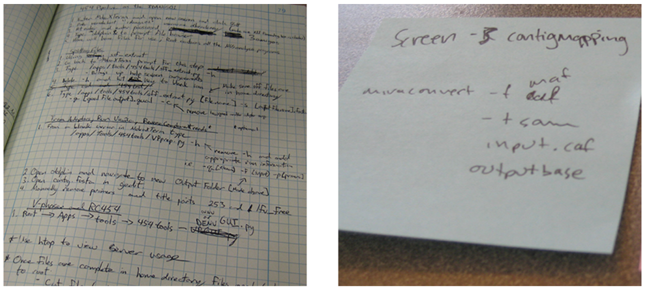
> "...it comes with very accepted default settings that come with the program but it's fully customizable if I am capable of doing that. It doesn't require me to do it, but I can change it if I want to. And then if I mess it up I can just click 'go back to default' and it's like it never happened." (P8)

Participants also expressed confusion over the nature of the algorithms themselves and what types of problems could be addressed with the different types of bioinformatics software. For example, Velvet [30] is a genome assembler that builds a de Bruijn graph from the set of reads and traverses that graph to build the sequences corresponding to the assembly. However, P6 believed that the tool could not handle this type of analysis:

> "They're de Bruijn graph assemblers so I think that's different from, like, Velvet and stuff, and I don't really know the specifics." (P6)

Unfortunately, current software tools are doing little to make their algorithms and parameters accessible to those without extensive computer science training. As an example, P1 expresses frustration regarding the readability of user manuals:

> "I can't understand a damn thing most of them say. Now, it's a little bit better now that I've gone through it, but when I was trying it was impossible. I just wish that they would speak like humans and not like robots." (P1)



```
Screen -S contigmapping

miraconvert -f maf
               caf
            -t sam
            input.caf
            outputbase
```

Figure 1: Examples of physical work artifacts. On the left, a lab notebook outlines the exact command-line statements used to complete analysis of a specific data set. On the right, a sticky note lists a few commonly used commands.

Thus, the current software does not provide sufficient algorithm transparency or accessibility to those without computer science training.

4.3.3 Access to Expertise

The participants reported lack of access to appropriate expertise as being a major concern in their research. At the basic level, some participants had difficulties working with Linux-based tools, which tend to assume a high degree of computer literacy. Out of the participants we interviewed, some had learned some basic Linux skills and programming on their own, but this was also limited:

> "I didn't even know how to change a directory without a click of a mouse. I didn't know a single thing. [The system support person at the university] came over and wrote the ten essential [Linux] commands. (sic)" (P1)

Contrary to the results of Preeyanon et al. [22] and Tran et al. [28], we observed participants referring to notes documenting previous computation procedures. In fact, participants were observed relying heavily on personal notes or "cheat sheets" to run bioinformatics programs. These notes were typically used to store Linux commands that were copied and pasted directly into the terminal, but were also used to keep a record of previously completed steps. We observed participants using a combination of digital notes (e.g. text files, OneNote documents, saved emails) and physical notes (e.g. work artifacts such as sticky notes and lab notebooks, shown in Figure 1). Documenting research methodology is an established aspect of research; however, the observed level of reliance on cheat sheets to perform virtually all tasks indicates that bioinformatics tool use is complicated and consists of difficult-to-remember steps.

Other participants would purchase commercial software packages that offered extensive support services. One participant (P8) considers availability of this software support to be of such importance that he regularly purchases software (e.g. Nexus Copy Number [5], CLC Genomics Workbench [8]) that costs over $20,000 per year in order to have guaranteed 24/7 support.

> "It costs a ton of money and it's very hard on my budget for the lab... [Even though] I've had problems with it, I've had outstanding support...This company, to me, is great,

> partly because their product is great, but their product support is super excellent." (P8)

While the above findings suggest obvious usability issues with existing bioinformatic software in that they are difficult for their target audience to use, they also suggest that facilitating access to technical and procedural help deserves further consideration. We return to this point in our discussion.

This participant—as well as others—expressed the importance of access to bioinformatics expertise and the difficulty of finding this expertise:

> "Someone like me relies heavily on finding a collaborator in bioinformatics who will have the time and the interest to put my project ahead of the other 250 projects that people around campus are proposing to them. It's not that you have to find a collaborator; you have to find a collaborator who thinks that your project is the coolest one on their table. And that's the hardest thing." (P8)

> "...so there's one guy at the Bule Lab, Michigan state, and there's another informaticist we know at Erie, so that's basically who we go to, our two connections. But you know, they are very busy so sometimes we are on our own." (P5)

> "I always acknowledge the people who help us ... right in the middle of the slide. I couldn't have done this without these two human beings because it was terrifying and they saved our life." (P1)

Despite the importance that bioinformatics researchers play in life science research, easy access to their expertise is not guaranteed. This high demand for bioinformaticians results in the inability to find collaborators. Without access to such expertise, the participants have to restrict the problems that they tackle and questions that they can answer to what their current bioinformatics software can answer:

> "I try to plan things out and I generate data such that the highest question that I ask is something that I can answer with this tool. If I can't answer it, I don't even try to go there." (P8)

> "I could ask much more sophisticated questions, but the reason I don't is because I know I'll hit a bottleneck in the bioinformatics analysis." (P8)

Further, participants expressed that the lack of access to bioinformatics and domain expertise also required them to put a lot of faith in the output of the software they use, which they justified with the idea that the tools they were using had been validated by the scientific community:

> "...If you're not a bioinformaticist, you have to go on the expertise of being widely used, and it's been validated and in that sense it seems that people really are getting useful data out of it then that's what you make your call on." (P7)

> "[At] the end of the day, you just have faith that the program's working or not." (P2)

This practice is a cause for major concern since it suggests that the results of the analysis are not necessarily properly verified.

4.3.4 Large-Scale Research Collaboration

Problems revolving around the reliance of experts are exacerbated as the size of the research groups becomes large and/or geographically dispersed. Many recent scientific findings have resulted from the coordination of large research groups that encompass upwards of 100 researchers with diverse expertise. Even smaller projects are no longer performed by individuals but groups of researchers.

> "We can have between 40 and 50 people on any project" (P6)

> "The number of collaborators on any project ranges from 2 to approximately 50. The median is around 10." (P10)

One of the main reasons for having a large number of collaborators is because the problems studied by our participants are multifaceted and require many different types of data analysis. P10 describes the different types of expertise required for one of her projects:

> "So, like, if we're doing a study on antimicrobial resistance on feedlots, which is what we're doing right now, we have to have people on there who are on the feedlots, we have to have veterinarians on there because they know about antibiotics, we have to have animal science people on there because they know about food safety, we need to have the some epidemiologists on there because they know about the ecology of resistance in feedlots, we need the bioinformaticists to help sequence data that we get, we need people who do the sequencing, we need statisticians to help with the statistics of the bioinformatics output. So, there's just a lot of different parts. So, for some of the grants we also have education components, so we need education people on there. So it just gets big really fast." (P4)

Scheduling and managing the research for groups of this size and be difficult. Geography seems to have a large impact; our participants described how they "frequently collaborate with people who are strong collaborators but are not co-located" (P10) so they rely on teleconferencing. When P10 was asked how he manages to remain connected with everyone on a large research project, his reply was that:

> "Well, you're assuming that I do, so that's a first thing" (P10)

implying that it is immensely challenging, if not impossible, at the current moment. Thus, he and other participants described that they had resort to focusing on the timeline of the project and ultimate goal.

> "I think we do mostly [connect with collaborators] but [are] probably goal- and timeline-oriented setup. So what's the ultimate project goal and breaking that down into subgoals and then looking at the timeline for when the work has to be done and when the money has to be spent, those kinds of things." (P10)

5. DISCUSSION

Results from our study confirmed prior anecdotal evidence of the usability of bioinformatics software. In particular, it supports the fact that the lack of documentation describing the software and its parameters, error messages, assumptions made about the data, and how to interpret the output of the software is limiting the objectives of the researchers. However, our results also demonstrate that these usability issues are contributing to researchers' reliance on computational experts to conduct their research. More importantly, when the needed experts are unavailable, they are altering their research questions to those that they feel they can confidently answer by running experiments and analyses that match the expertise they gained on previous studies. This practice is likely resulting in novel findings being delayed. Our results also demonstrate that life scientists' reliance on bioinformatics experts results from an incomplete understanding of the underlying algorithmic mechanics of the software, due to the software's lack of transparency. To overcome this, some of our participants suggested abstracting away from the underlying mechanics and algorithms of the software through automatic parameter selection and/or graphical user interfaces. Others simply used default parameters during analysis which may not be appropriate for their dataset. Hence, life science researchers are treating bioinformatics software as a "black box" where they don't quite understand what to put in or what is coming out. This has two major implications: (1) Resnick et al. stated:

> "By building their own instruments, and understanding the capabilities and limitations of those instruments, scientists have historically gained deeper insights into the nature of the phenomena under investigation" [23]

which suggests that life scientists are missing out on gaining deeper insights that could preclude important scientific findings, and (2) treating bioinformatics tools as black boxes can result in misinterpretation of the data, as seen by the recent number of retractions due to non-robust data analysis.

While this indicates that developers should instead strive to make the underlying algorithmic pinnings more transparent, this approach places the burden back onto the scientists by requiring them to learn mathematical and computer science principles in order to analyze their data. It is the need for this expertise that has required scientists to depend on computational experts. The trade-off between abstraction and transparency is one that needs further examination.

One of the most pervasive themes highlighted by our study was the importance of collaboration in projects. Collaboration was found to be necessary in our participants' research because it provided access to a wide range of expertise that is required to conduct large and complex studies (e.g., the antimicrobial study of feedlots described above by P10). However, the collaboration between the participants and bioinformatics experts, and the collaborations between the participants and other life scientists appeared to be unique in many ways. Collaboration with bioinformatics experts was a direct result of a lack of understanding how computational tools should be used in the research workflow. Whereas collaboration with other life science researchers tended to focus on the need to answer complex biological phenomena or extend previous findings. While participants showed frustrations regarding the overhead associated with collaborating with both groups, the collaboration with computational experts was the only one mentioned to have a clear effect on the analysis performed by researcher. For example, as stated above by P8, it was not uncommon for researchers to limit the questions they ask explore based on what software they are comfortable using and their understanding of what answers the software can provide.

In addition, we found that our participants' research projects relied on traditional methods to communicate with their collaborators (e.g., in person meetings, phone calls, and emails). This is unsurprising since the computational tools used through the research workflow are not built for collaboration nor offer collaboration facilities. In fact, all of the software mentioned by our participants is such that sharing of results and data is difficult due to the size of the output and results. Participants often mentioned it was easier to share data by physically transporting it through package delivery services than transfer it over the Internet. In future work, we plan on exploring how tools can be designed to provide life scientists with the computational help they need while facilitating collaboration among a large number of geographically dispersed researchers.

Lastly, Preeyanon et al. and Tran et al. previously reported that many life scientists do not keep a record of computational analysis procedures in the way they record wet-lab experiments [22, 28]. Our results contradict these findings. All of our participants had some type of artifact that recorded the commands used to perform the analyses. In fact, early researchers often relied on this record to perform even simple tasks within the computation environment (e.g., changing directories). However, unlike with wet-lab records, none of our participants recorded the outcomes of the steps, only the procedure needed to replicate the task. Instead, they often relied on having access to the digital output. Describing the output in detail would be a insurmountable task due to the size of the data and output, however, it isn't unreasonable to expect them to write at a summary or overview of the results. A possible avenue of future work is to explore if previous proposed techniques (e.g., [29]) could be adapted to this domain to enable scientist to record their workflow in a more thorough manner.

5.1 Study Limitations

We acknowledge that this study may be affected by problems inherent to the self-reporting nature of interview data and a small sample size.

Since this study aims to identify usability issues that are present during typical use, it is important to ensure that the population under observation accurately represents the population of typical users. Although interviewees were chosen from a range of biological research areas and had varying levels of education, it would have been beneficial to expand the set of participants to have included more experienced tool users.

Interview statements that cannot be substantiated by evidence in work artifacts (e.g. lab notebooks and text files) rely solely on self-reported information that may be imprecise. Additionally, although participants were asked to demonstrate specific instances of recent research to reduce recall bias and ground the interview data, the study could be strengthened by validating statements with data gathered from observing scientists conducting actual research.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented the results of semi-structured interviews which provided insight into the effects of bioinformatics software usability on life science research. Future work includes addressing study limitations described above by interviewing additional participants and conducting additional studies to observe biological researchers using bioinformatics tools while conducting actual research. Furthermore, to ensure that our interpretations of qualitative data are accurate, future work includes conducting follow-up interviews with participants from this study. Finally, additional studies can be conducted to explore design recommendations and evaluate their effectiveness.

7. REFERENCES

[1] M. Abouelhoda, S. Alaa, and M. Ghanem. Meta-workflows: Pattern-based Interoperability Between Galaxy and Taverna. In *Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science*, Wands '10, pages 2:1–2:8, New York, NY, USA, 2010. ACM.

[2] M. Abouelhoda, S. A. Issa, and M. Ghanem. Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support. *BMC Bioinformatics*, 13:77–77, May 2012.

[3] J. C. Bartlett and E. G. Toms. Developing a protocol for bioinformatics analysis: An integrated information behavior and task analysis approach. *Journal of the American Society for Information Science and Technology*, 56(5):469–482, 2005.

[4] H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-centered Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[5] BioDiscovery. Nexus copy number, 2013. http://www.biodiscovery.com/software/nexus-copy-number/.

[6] P. K. Chilana, E. Fishman, E. M. Geraghty, P. Tarczy-Hornoch, F. M. Wolf, and N. R. Anderson. Characterizing Data Discovery and End-User Computing Needs in Clinical Translational Science. *J. Organ. End User Comput.*, 23(4):17–30, Oct. 2011.

[7] P. K. Chilana, C. L. Palmer, and A. J. Ko. Comparing Bioinformatics Software Development by Computer Scientists and Biologists: An Exploratory Study. In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, SECSE '09, pages 72–79, Washington, DC, USA, 2009. IEEE Computer Society.

[8] CLC Bio. CLC genomics workbench, 2014.

[9] F. S. Collins, A. Patrinos, E. Jordan, A. Chakravarti, R. Gesteland, L. Walters, t. m. o. t. DOE, and N. p. groups. New goals for the u.s. human genome project: 1998-2003. *Science*, 282(5389):682–689, 1998.

[10] M. Corpas, S. Fatumo, and R. Schneider. How Not to Be a Bioinformatician. *Source Code for Biology and Medicine*, 7(1):3, 2012.

[11] J. Goecks, A. Nekrutenko, J. Taylor, and T. G. Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11(8):R86, 2010.

[12] J. D. HincapiÃl'-Ramos, A. Tabard, J. Bardram, and T. Sokoler. GridOrbit Public Display: Providing Grid Awareness in a Biology Laboratory. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3265–3270, New York, NY, USA, 2010. ACM.

[13] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(suppl 2):W729–W732, 2006.

[14] A. Hunter, A. Macgregor, T. Szabo, C. Wellington, and M. Bellgard. Yabi: An online research environment for grid, high performance and cloud computing. *Source Code for Biology and Medicine*, 7(1):1, 2012.

[15] J. Lazar, J. H. Feng, and H. Hochheiser. *Research Methods in Human-Computer Interaction*. Wiley Publishing, 2010.

[16] C. Letondal. biok: Biology Interactive Object Kit. In *BOSC2001: Bioinformatics Open Source Conference, Copenhague, Denmark*, pages 19–20, 2001.

[17] J. P. Massar, M. Travers, J. Elhai, and J. Shrager. BioLingua: a programmable knowledge environment for biologists. *Bioinformatics*, 21(2):199–207, Jan. 2005.

[18] B. Mirel. Usability and usefulness in bioinformatics: Evaluating a tool for querying and analyzing protein interactions based on scientists' actual research questions. In *Intergovernmental Panel on Climate Change*, pages 1–8, Seattle, Washington, United States, 2007. IEEE International.

[19] K. Okonechnikov, O. Golosova, M. Fursov, and others. Unipro UGENE: a unified bioinformatics toolkit. *Bioinformatics*, 28(8):1166–1167, 2012.

[20] K. Pavelin, J. A. Cham, P. de Matos, C. Brooksbank, G. Cameron, and C. Steinbeck. Bioinformatics meets user-centred design: A perspective. *PLoS Comput Biol*, 8(7):e1002554, 07 2012.

[21] A. Pollack. DNA Sequencing Caught in Deluge of Data. *The New York Times*, Nov. 2011.

[22] L. Preeyanon, A. B. Pyrkosz, and C. T. Brown. Reproducible bioinformatics research for biologists. In *Implementing Reproducible Research*, The R Series. Chapman and Hall CRC, apr 2014.

[23] M. Resnick, R. Berg, and M. Eisenberg. Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *The Journal of the Learning Sciences*, 9(1):7–30, 2000.

[24] T. Seemann. Ten recommendations for creating usable bioinformatics command line software. *GigaScience*, 2(1):15, 2013.

[25] J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. R. Gilbert, I. Korf, H. Lapp, H. Lehvaslaiho, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney. The bioperl toolkit: Perl modules for the life sciences. *Genome Research*, 12:1611 – 1618, 2002.

[26] L. Stein. Creating a bioinformatics nation. *Nature*, 417(6885):119–120, May 2002.

[27] A. Tabard, J.-D. Hincapi-Ramos, M. Esbensen, and J. E. Bardram. The eLabBench: an interactive tabletop system for the biology laboratory. In *Proc. of ITS'11*, pages 202–211. ACM, 2011.

[28] D. Tran, C. Dubay, P. Gorman, and W. Hersh. Applying Task Analysis to Describe and Facilitate Bioinformatics Tasks. *Medinfo*, page 818, 2004.

[29] R. Yeh, C. Liao, S. Klemmer, F. GuimbretiÃÍre, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke. ButterflyNet: a mobile capture and access system for field biology research. In *Proc. of the CHI'06*, pages 571–580. ACM, 2006.

[30] D. R. Zerbino and E. Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–829, 2008.

[31] W. Zhang, J. Chen, Y. Yang, Y. Tang, J. Shang, and B. Shen. A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies. *PloS ONE*, 3(6):e17915, 2011.