

Designing and Evaluating Scientific Workflows for Big Data Interactions

Ronak Etemadpour¹, Paul Murray², Matthew Bomhoff³, Eric Lyons³, and Angus Graeme Forbes²

¹Oklahoma State University, etemadp@okstate.edu

²University of Illinois at Chicago, {pmurra5, aforbes}@uic.edu

³University of Arizona, {mbomhoff, ericlyons}@email.arizona.edu

Abstract—This paper explores the specialized nature of research-oriented web applications that enable interactions with and the visual analysis of “Big Data,” i.e., large, heterogeneous scientific datasets. We introduce a pragmatic methodology for the design and evaluation of *scientific workflows* in research-oriented web applications. Through an in-depth usability study of the *CoGe* web application, a system that provides a rich set of tools for exploring genomic datasets, we demonstrate: how to identify bottlenecks in multi-step tasks; how to analyze these bottlenecks in order to provide effective solutions for improving user experience; and how these solutions may more generally apply to similar research-oriented websites in other scientific domains that also enable scientific workflows. Specifically, we provide details regarding: our user interviews, the visualization system we created to analyze complex tasks associated with scientific workflows, and how this analysis directly leads to suggestions for improvements in the current implementation of the *CoGe* web application. A follow-up study was carried out which indicates that our suggestions improved the ability of *CoGe* users to navigate and complete custom workflows, leading us to believe that our approach could also be applied to other research-oriented web applications that utilize scientific datasets.

Keywords—*Scientific workflows, Big Data, user evaluation, visual analytics.*

I. INTRODUCTION

Providing on-line tools that allow members of research communities to collaboratively explore and to analyze large datasets has become an increasingly important component of scientific research. A defining characteristic of research-oriented web applications is that they are simultaneously highly specific and highly flexible— They provide an integrated set of tools that enable various scientific tasks, while at the same time encouraging users to mix-and-match which tools are used so that they can more easily generate and validate novel hypotheses about their data. This flexibility means, however, that unlike with most web applications, neither the tasks nor the goals of a user’s session will necessarily be known in advance. Moreover, the community of users and the types of tasks in which they are interested in are expected to evolve as new scientific discoveries are made and new topics are introduced.

This paper introduces a pragmatic methodology for the design and evaluation of *scientific workflows* in research-oriented web applications. Based on existing techniques for task evaluation, but adapted for research-oriented web applications, our approach provides a way for application developers to identify issues in common tasks shared by different scientific

workflows. Our methodology consists of: identifying common multi-step tasks shared by different workflows; recording expert users as they execute these tasks; coding these tasks in terms of distinct steps; creating a visual map of how users execute (or fail to execute) these tasks; comparing this map to an “ideal” task flow as anticipated by the application developers; analyzing user comments from the video recording where the ideal and actual steps diverge; and offering suggestions for the application developers that could most positively impact user experience. We present an in-depth case study where we observe and analyze expert users of the *CoGe* web application, an online suite of tools for comparative genomics. Based on this study, we gave a number of suggestions to the application developers indicating how they could facilitate effective workflows. Additionally, we present a follow-up study where we evaluated user interactions after our suggestions were implemented by the application developers. We show that our methodology was effective at discovering issues in the web application, and that our suggestions led to a series of design improvements that enabled users to more efficiently complete their tasks. We believe that both the results of our evaluation and the evaluation methodology are valuable to comparative genomics applications as well as to other applications that aim to improve scientific workflows. Our contributions include:

- Identifying and articulating issues that are especially relevant for the design and implementation of scientific web applications, explaining how they are different from other types of web applications;
- Presenting an effective user experience evaluation of a complex scientific web application, identifying many bottlenecks that were then improved by the designers of the web application;
- Providing easy-to-follow guidelines for researchers aiming to improve their tools for various scientific communities with differing levels of expertise working with complex data sets;
- Introducing a practical methodology that could be used by designers of research-oriented web applications to quickly identify user experience issues.

II. THE COGE WEB APPLICATION

The *CoGe* web application¹, also referred to by its fuller name, *CoGe: Accelerating Comparative Genomics*, was first introduced in 2008 to provide a means for researchers from

¹<https://genomeevolution.org/CoGe/>

various disciplines requiring tools for comparative genomics to compare and analyze genomic datasets [1]. Since then, the user base has grown to well over 1000 registered users; between September 2013 and September 2014 there were approximately 3100 visitors to the website per month, carrying out an average of 5300 sessions per month. While some of these sessions involved the use of a single tool, *CoGe* is becoming increasingly well known as a resource where scientists can create custom workflows in which a series of data-intensive tasks can be executed. *CoGe* has been used for a variety of different research investigations related to the analysis of gene sequences for particular organisms or for helping to model genomic evolutionary processes. Some recent research avenues include: explorations of stone fruit genes [2], the aliquoting of flowering plant genomes [3], comparing cabbage families [4], modeling poultry genomics [5], analyzing gene deletion in grasses [6], comparing syntenic regions between papaya, poplar, and grape plants [7], exploring the dynamics of various elements in the plants that produce canola oils [8], analyzing *E. Coli* sequences [9], and screening synteny blocks in pairwise genome comparisons [10], among others.

The main functionalities of *CoGe* include the ability to work with genomic data in flexible ways. Genomic datasets are often extremely large, and *CoGe* allow users to upload, compare, analyze, and visualize multiple datasets simultaneously. The largest genome sequence available for comparative analytics in *CoGe* (and thus far the largest that has ever been sequenced) is the Loblolly Pine, which totals 14GB. The entire set of genome sequence data available is over 1.2TB. The largest quantitative measurement dataset in *CoGe* is 18GB, and all quantitative datasets are a total size of over 1.7TB. Additionally, the analysis of genome sequences can also result in large datasets. A recent analysis compared two cotton genomes, producing a 133GB dataset. *CoGe* is powered by the iPlant program, a high-performance cloud cyberinfrastructure that allows web applications to use remote servers for computation, data analysis, and storage.²

CoGe provides a suite of tools for analyzing genomic data. An example of some of the tools available on *CoGe* include: *OrganismView*, which allows users to get an overview of an organism’s genomic make-up, and to interactively visualize them using a genome browser; *CoGeBlast*, which lets a user search for local alignments within any number of genomic sequences; *SynMap*, which can identify regions of synteny within any two genomes; and *GEvo*, which can discover patterns of microsynteny genome evolution. Furthermore, each of these tools can be chained together to create a sophisticated workflow. For example, Tang and Lyons [8] describe a workflow used to explore the *Brassica rapa* genome: Load the *B. rapa* genome sequence; perform a genomic comparison of *B. rapa* with itself; perform genome comparison between *B. rapa* and *A. thaliana* (a genome already pre-loaded on the *CoGe* site); identify *B. rapa* regions that are orthologous with *A. thaliana*’s TOC1 gene; finally, perform a high-resolution comparison for the regions of interest and find conserved non-coding sequences. While this workflow ultimately lead to insight about the *B. rapa* genome, the authors of this study note: “[T]here is no specific workflow or analytical pipeline one must follow. Instead, the questions asked and the

discoveries made drive the direction of the analyses” [8].

The number of tools and their functionality is growing, and thus the potential workflows that are available to a user are also increasing. For instance, as indicated in the *CoGePedia*³ (a public Wiki maintained by the community of *CoGe* users), in the last few months a dozen new features have been added, including the ability to export data via a webservices API, the ability to search data in the *SynMap* tool more effectively, new ranking features for *CoGeBlast*, the ability to annotate analyses shared by other users, tools to import and view RNASeq data, along with a host of bug fixes and security improvements. New tools or updates to existing tools are regularly presented at relevant venues in order to engage the bioinformatics community directly [11], [12], [13].

III. WORKFLOW ANALYSIS

Despite the success of the *CoGe* system within the bioinformatics community, the developers of the web application were aware that the site appeared formidable and overly complex to many users. There is a high learning curve for some of the new tools, and an analysis of web traffic (using Google Analytics) showed that many users seemed to abandon genomic analyses midstream. Moreover, the application developers received numerous emails from users asking for help with simple tasks that were meant to be self-evident, and in some cases it was unclear to users how certain goals could be achieved, such as: how to move backwards to a previous state; how to determine when a task was finished; or how to tell who had access to their data analysis. A central philosophy guiding the development of the application is that it must remain flexible so that it can be used by a wide range of users for a variety of analysis tasks, and furthermore that users never feel constrained by a lack of possibilities. Whereas many usability studies aim to improve specific goals, we wanted to improve user experience, especially for those less familiar with the application, while at the same time not limiting more experienced users. That is, we wanted to maintain a balance between functionality and usability, two characteristics that, as is noted by Herr et al. [14] and, more generally, by Goodwin [15], are often in opposition.

Our work relates to previous work on visualizing and evaluating workflows. For example, Heer et al. [16] transform user logs into “behavior graphs” in order to better understand branching and revisitation patterns and evaluate visualization design. Dou et al. [17] describe a tool called *WireVis* that enables designers of visualization analytics systems to recover users’ reasoning processes from their interactions. Zhang et al. [18] introduce a tool called *FlowRecommender* to automatically recommend effective workflows to users of scientific applications. Guo et al. [19] utilize a task analysis methodology to design tools for examining EEG data that support the user’s visual analysis process. Freire et al. [20] introduce *VisTrails* to capture each step taken by a user when constructing a set of data products, enabling a user to see and to streamline their data exploration process. Our work presents a holistic investigation of the user experience of designing and using scientific workflows, investigating not only the individual analysis steps of the workflows, but also the design and usability of the web application that enables these workflows.

²<http://www.iplantcollaborative.org>

³<https://genomeevolution.org/wiki>

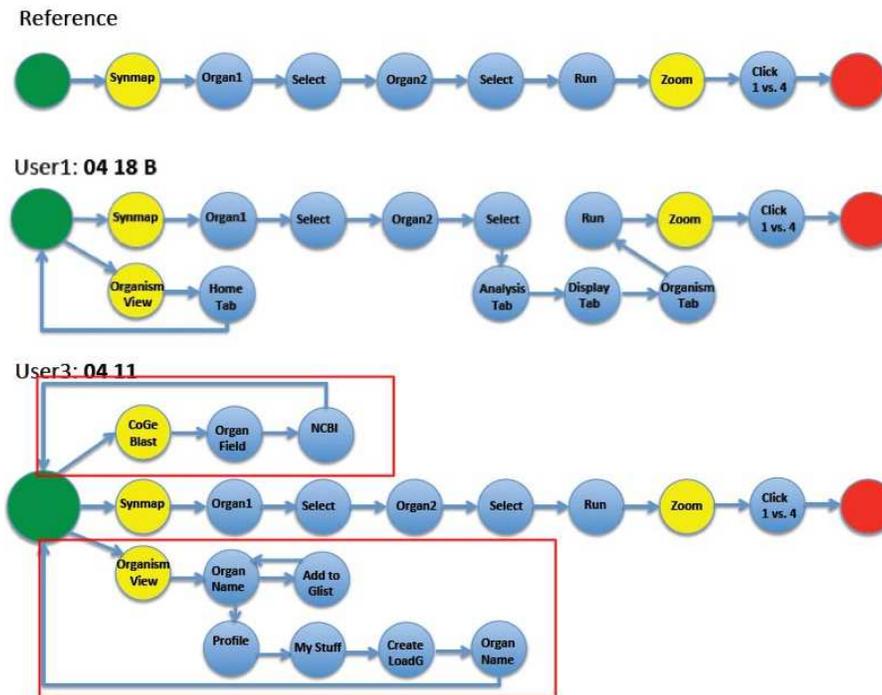


Fig. 1. Here we show the workflow for users completing Task 1. Green indicates the start of the task; red indicates the completion of the task; yellow indicates that the user moved to a new page; blue indicates that they interacted with the current page. At the top we show the “ideal” path that the application developers expect users to navigate to complete the task successfully. In the middle and bottom we show how two of our users actually completed the task, and where specifically they diverged from the ideal path. User 1 (in the middle) made a few errors, first selecting the wrong tool and then, later on being slightly sidetracked. User 3 (on the bottom) however took much longer and made more “wrong turns.” The areas in red boxes show areas where a user was particularly confused and either took a long time to complete or required multiple hints.

In order to evaluate the usability of the web application, we first gathered examples of complete workflows from the application designers. We then identified complex tasks that were repeated in many of these workflows, and which the application developers indicated were areas of the site where many users had reported confusion. Following this identification, we then chose four of these complex tasks that required multiple steps. We wanted to observe exactly how it was that users navigated these tasks, and to figure out what specifically it was that allowed a user to successfully complete the task or discover why they failed to complete the task.

A. Workflow Visualization

In order to understand how users were navigating (or failing to navigate) the site, we created “workflow graphs” for the users. These graphs are made up of nodes that indicate every discrete step a user takes as they seek to complete a task. These user workflow graphs (defined in detail below) were compared against a “ground truth” version of the workflow as expected by the application designer. By overlaying the two graphs on top of each other, it becomes immediately apparent at which steps a user gets confused, and where the expectations of the developer diverge from real-world scenarios.

B. Video Coding Complex Tasks

Although the application developers could observe how a user navigated the site, they were not able to determine the reason why a user diverged from expected behavior. By video recording sessions with expert users, we were able to ask users

at each step the reasoning for making a particular step, and thus we could provide ample evidence regarding which design strategies were effective and which were not. Additionally, we encouraged users to talk aloud as they completed tasks, and the free-form commentary also provided us with insight about the usability of the site.

C. Preliminary Studies

In conducting a long-term user study, we had two primary goals: In a pragmatic sense, we hoped to improve the usability of the *CoGe* system. At the same time, we saw an opportunity to develop and test novel user assessment methodologies geared towards scientific applications— such as *CoGe*— where user workflows are often complex.

We first conducted exploratory studies with a small number of graduate student biologists who were familiar with the biological concepts involved in the *CoGe* application, but who had not used the application extensively. In these initial studies we monitored researchers as they explored the *CoGe* application. Two of *CoGe*’s application developers were also present to cue the user with common tasks that a typical researcher might perform. These initial user studies were designed to demonstrate how researchers in general would typically use the *CoGe* application, and thus the instructions given to the user were intentionally open-ended. In this phase, we did not give the user specific tasks or instructions, but instead made general observations in an effort to understand the workflow involved in a typical analysis.

Combining the feedback gathered through these exploratory studies with further input from two experienced *CoGe* developers allowed us to build a set of complex, representative tasks that would likely be a part of the typical *CoGe* user’s workflow.

IV. EVALUATING COMPLEX TASKS

After performing preliminary user studies, we sought to evaluate the ability of users to navigate through the *CoGe* system and to perform tasks common to a researcher interested in genomics datasets. To this end, and in order to perform this evaluation in a controlled fashion, we constructed four complex tasks that we considered to be representational components of a typical researcher’s workflow. The chosen tasks involved some of the most commonly used application features, as shown on application usage logs. The tasks were also chosen because of their relative complexity, with each task requiring a minimum of between six and eleven distinct steps to complete, and also because they were considered by the application developers to be some of the most distinguishing features on the *CoGe* system. All four tasks were related to the retrieval and comparison of genomic information and sequence data, and each task was framed in terms of a typical use-case, in which a researcher had as a main goal the aim of better understanding patterns within large genomic datasets. The four tasks are defined as follows:

Task 1 – Genome Comparison: “Generate a dotplot genome comparison for two chicken genomes: Red jungle fowl and reference chicken genome. Zoom into a region of interest: Chromosome one in red jungle fowl versus chromosome four in reference.” This task required users to use the *SynMap* tool to perform a comparison of two genomes (using a process of syntenic alignment) and to generate a visual comparison of two genomes (called a “dotplot”).

Task 2 – Genomic Expression Data: “Load an experiment data file from the local computer and associate with an existing genome.” *CoGe* is a collaborative system, and users are able to contribute their own quantitative measurement data – such as genomic sequence data, feature analysis data, and experimental results – by uploading files. This task required users to upload a sample file containing experimental data to *CoGe* from the desktop.

Task 3 – Sharing Data With Users: “Create a new user group, add another user to the new group, and share an existing experiment with the new group.” One of *CoGe*’s strengths is its collaborative nature, and this task aimed to assess the ability of a user to share experimental data with other *CoGe* users.

Task 4 – Genomic Sequences Search: “Search a list of genomes of interest for a specific sequence, filtering on e-values less than 1e-6. Specific genome names will be given when needed.” *CoGe*’s BLAST search allows users to identify regions of similarity between a given genomic sequence of interest and *CoGe*’s database of over 20,000 genome sequences, yielding tabular results along with an interactive visualization.

For each task, we gathered input from the application developers in order to define an *optimal path* as a set of instructions that lead from the *CoGe* home page through the *CoGe* web application to a specific end page, at which point

a complex task necessary to an analysis workflow would be completed. The instructions given to the user for each task were concise, and were formulated with the intention of giving the user a clear goal, but without including any hints of how to complete that goal.

We ran our main user study on 8 participants, all of who were experienced with the genomics tasks enabled by *CoGe*, but only half of who had any direct experience with the *CoGe* website itself. The subjects, all of whom were unpaid volunteers, included two graduate students in the life sciences, two post-doctoral researchers in plant biology, three research scientists in biology, and a project manager for a large biological cyberinfrastructure system. Each subject was assigned the four tasks given above in a fixed order. Participants were provided with an introduction to the goals of the study and were told that they would be recorded while performing certain tasks within the *CoGe* application. Each user was asked about their prior knowledge and experience within the domain of molecular biology and with the *CoGe* application in particular. Users were seated at a desk in front of an Apple iMac with a 21.5” screen; all participants had normal vision. During each study, one application developer gave instructions to the participant, while three other researchers were present to observe and give guidance to the user, if necessary.

After receiving their instructions, the users were instructed to begin completing the task. They were encouraged to comment on their experiences, and vocal feedback from each user feedback was recorded throughout the study. When users reached points of confusion, the researcher providing instruction did not immediately give direct guidance, but instead repeated the general directions for the task. If the user continued to express confusion, the researcher gave instructions to the user, directing them to the next step in the optimal path for that task. These instructions were recorded as “hints” given to the user. After each task was completed, the user was asked for general feedback about their user experience.

A. Coding the Videos

Prior research has demonstrated the value of recorded video analysis. Video analysis that includes verbal feedback from a human participant provides an enhanced form of data collection, as the user’s thoughts and opinions can be recorded along side their interactions with an application. For instance, Tang et al. [21] use a “talk aloud” approach that encourages users to give verbal feedback while interacting with a novel application. Research by Kosaba [22] and by Tory and Staub-French [23] has also shown the effectiveness of combining video analysis with measures such as task completion time, task correctness, and user satisfaction.

In our study, each user’s actions were recorded with screen capture software which also recorded the user’s voice as they proceeded through the study. For each video, a follow-up analysis was performed by the researchers using an agreed-upon coding system. In these analyses, user interaction events within the application, such as mouse clicks and interactions with menus, were recorded along with notes summarizing the user’s vocal feedback at each point in the study. The combination of quantitative measures and the qualitative analysis of user feedback enhanced our understanding of the user experience

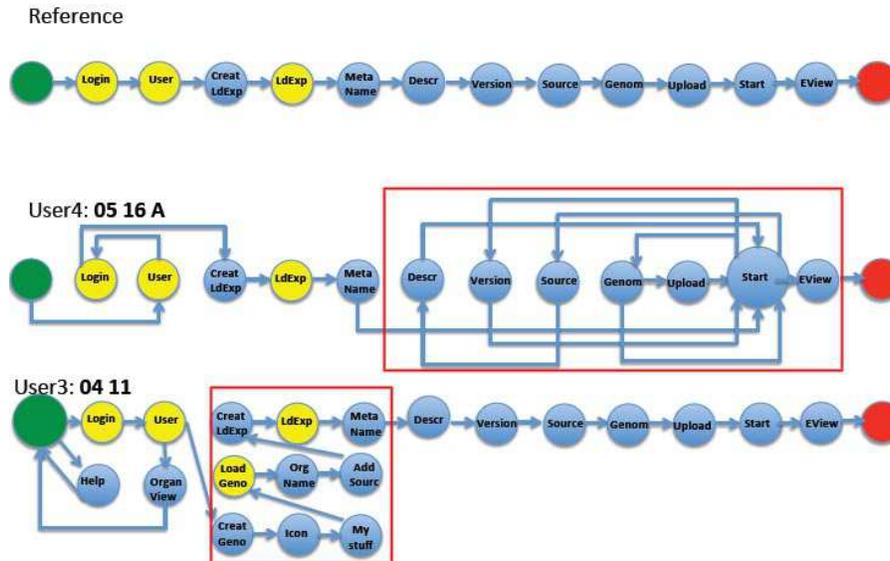


Fig. 2. Here we show the “workflow graph” for users completing Task 2. In the middle and bottom we show the steps two of our users required to complete the task, and where specifically they diverged from the ideal path (top). User 4 (in the middle) made a number of errors, first selecting attempting to use a tool before logging in to the system, and then, later on being extremely sidetracked when trying to load an experiment. User 3 (on the bottom) made a number of “wrong turns” when trying to load an experiment, mistakenly loading a genome instead. The areas in red boxes show areas where a user was particularly confused and either took a long time to complete or required multiple hints.

and mitigated the potential for arbitrary interpretation of a user’s actions. Specifically, in our analyses, we kept track of four variables for each interaction (i.e. mouse click):

- **Tool:** The page for a particular tool within the *CoGe* application that the user was currently visiting.
- **SubTask:** The current step being executed while using that tool.
- **ClickedOn:** An indication of whether or not the user performed their intended action, along with a description of the incorrect action when he or she did not.
- **Hint:** Coded as “1” if a hint was given to the user on this step.

Additionally, we tracked the total time it took a user to complete each of the four tasks.

B. Workflow Graph Visualization

Given the potential for a wide variety of complex workflow configurations across a number of users, we designed *workflow diagrams* to enable a visual representation of user interactions within the *CoGe* system. Previous research suggests that visual representations can significantly increase an analyst’s ability to understand complex information [24], [25], and more specifically, various path visualizations have been used to represent workflows [26], [27], [28]. Our workflow visualization builds upon and complements the quantitative and qualitative evaluation of video recordings of user interactions. In our graphical representations (see Figs. 1 and 2), the starting point of each task is represented as a green circular node, pages within the *CoGe* application are shown in yellow, sub-tasks are shown in blue, and the target node– the final step in the task– is shown in red. Each user interaction is represented by an

edge between one sub-task or page and another. Our graphical representation is thus comprised of a directed graph in which the arcs constitute a multiset of ordered pairs, meaning that loops and multiple arcs may be present. We use straight lines, as research has indicated that they perform better than curved arcs in many situations [29], but future work will explore different visual designs for the layout and encoding of the workflow diagrams.

A “reference” graph illustrates the optimal path through the *CoGe* system for a given task, and acts as a visual reference comparison to the user’s attempts. For example, in Fig. 1, which includes user graphs for Task 1, the graph of actions performed by User 3 includes two additional loops (shown in red) as compared to the reference graph. In this case, the user had visited the “CoGe Blast” and “Organism View” pages before they were able to successfully find the “Synmap” page. This visual representation helped us to interpret where this user struggled in their completion of a task, and in particular highlights that they had difficulty finding the proper tool (“SynMap”). Fig. 1 also illustrates similarities across users, as User 2 had a similar set of errors as User 3, visiting “Organism View” before finding the “SynMap” page. Overall, for Task 1 six out of eight users had similar issues, though once they found the proper tool for genome comparison the task was completed easily. Using these visual representations to analyze quantitative and qualitative data recorded for these tasks, we compiled a list of issues and recommended changes that were given to the *CoGe* application developers.

Each task illustrated a different set of problems that were shared by a number of users. Task 1 helped demonstrate that many users did not know which tools (e.g. “Synmap”) should be chosen, and suggestions were given to the application developers to add short descriptions of each tool (e.g. text that displays on hover). Thus, Task 1 illustrated a lack of

knowledge of tool functionality among the *CoGe* users. In Task 2, as shown in Fig. 2, User 4 repeatedly tried to submit a form while mandatory fields were not filled in, leading us to suggest that mandatory form fields needed to be clearly highlighted. Also in Task 2, User three had difficulty locating certain buttons, such as the “share” button to share experimental data. These results suggested modifications in the placement and styling of important buttons. In analyzing Task 3, we found that users often clicked on a list item’s title— opening a new tab— when they should have clicked on the checkbox to the left of the title for that item. Users also had trouble understanding the icons placed above the list of items, meant to be used to perform actions on groups of documents. Based on these results, we suggested the use of larger icons, as well as the re-structuring of the behaviors involved in item selection. The workflow graphs also enabled us to recognize when there were no major issues. For example, in Task 4 all users were able to complete the necessary steps quickly, mostly aligning with the expected flow as described by the application developers.

V. IMPROVEMENTS TO *CoGe*

Following our analysis, we suggested several changes to the *CoGe* user interface in order to make it easier for users to more efficiently complete some of the complex tasks enabled by the *CoGe* web application. These suggestions were determined based on how frequently they were mentioned by our subjects and also by our observation of how many times a particular issue led to a misstep. By prioritizing these issues we highlighted the changes we believed would have the most impact. That is, though our results were not necessarily comprehensive, we expected that the suggestions based upon their analysis would nonetheless improve the usability of the application, and specifically that they would reduce the number of steps and the “wrong turns” taken when the user is executing a workflow.

A. Design Improvements

Some of the main problems that caused issues turned out to be relatively minor, and for the most part were not expected by the application developers. Users repeatedly indicated a number of issues, including: their inability to understand the meaning of various icons; the confusing placement of text, buttons, and tabs, especially when they changed position in different tools; the frustration of not being able to move backwards in history in order to undo a step or to repeat a step; and their confusion about which fields in a form were required, among others. Thus, a good portion of our suggestions included fairly straightforward changes that the application developers were able to implement quickly and that made a large impact on the overall efficiency of the users. For example, the meaning of several icons, particularly on the “User” page, was unclear to many users. A button that allows one to share data with other users was represented as glyph meant to represent the silhouette of a person’s head and shoulders. However, the glyph was so small that several users could not discern its meaning— more than one user remarked that the glyph looked “like a sunset.” *CoGe* was updated to include much larger icons and glyphs were re-designed to more clearly represent the functionality of the button. In addition, tooltips were added which display descriptions of each button’s functionality.

Several users chose the wrong tools on the *CoGe* home-page, where the names of several different tools are listed. One user commented that once they “knew where [they] needed to go, it wasn’t that bad [...] understanding where to start was the problem.” *CoGe* originally included a list of buttons labeled with the names of several available tools, such as “OrganismView,” “CoGeBlast,” etc. However, these tool names alone did not effectively convey their functionality to an inexperienced user. *CoGe* was updated to replace these textual buttons with glyphs designed to illustrate the basic functionality of each tool. In addition, a short description of each tool is now included next to each glyph.

Many tools in the *CoGe* system require the user to fill out a form, often with many options and fields, and then click a button that will, for example, run an analysis. In the original version of *CoGe*, some form-completion buttons were unexpectedly located at the top of the page rather than the bottom, requiring users to return to the top of the page after completing the form. The updated version of *CoGe* implemented a consistent form layout, with all form-completion buttons located at the bottom of the page.

Throughout the original version of *CoGe*, when a user enters a new page in the application, a new browser tab is opened. When a new tab is opened, the browser history for that tab is empty, meaning the “back” button in the browser is disabled. Many users had difficulty going back to previous sections of the application, not realizing (or forgetting) that the sections of the application that they had previously visited were in separate browser tabs. At the same time, after visiting many sections of the application, many tabs would remain open simultaneously, requiring the user to visit each tab in an effort to locate their desired content. In the updated version of *CoGe*, links now open in the same tab, so that the browser “back” button works as expected. In addition, some sections of the site (such as the “My Profile” page) now open content in a nested pop-up window located within the same page.

Despite its bright color and bold text, originally many users had difficulty locating (or even noticing) a labelled button that was necessary for one of the tasks. This “Create” button was updated to appear at the very top of a list of actions, rather than at the bottom, and it was made much larger. Similarly, almost every user in the first phase of our user study tried to submit certain forms without filling in mandatory fields. The updated version of *CoGe* includes markers to denote mandatory form fields.

A few minor changes were also suggested, although our analysis didn’t indicate that they caused any specific problems. However, we felt that they would generally improve the overall usability of the application. Several users commented on the small font size throughout the application. *CoGe* was updated with dynamic fonts and buttons that are scaled proportionally to the size of the user’s window. A general “decluttering” of the website’s layout was undertaken as well. For instance, in one part of the application, a list of filter options was collapsed by default, requiring the user to click to open the list of filter options. The updated version of *CoGe* shows the options as expanded by default. In addition, certain optional (and rarely used) form fields were set to be hidden by default.

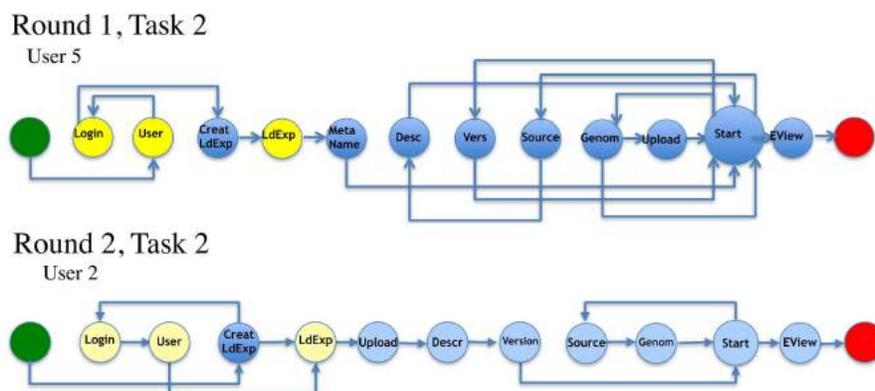


Fig. 3. Here we compare a typical workflow for a user completing Task 2 in the original study (“Round 1”), before our suggestions were implemented, versus the workflow that took the longest amount of time in the follow-up study (“Round 2”). That is, the worst case in the follow-up study is much better than the average case in the original study. (The colors have the same meaning as in Fig. 1).

B. Conceptual Reformulations

Other suggested improvements were somewhat more conceptual in nature, but they nonetheless led the developers to make relevant changes to the *CoGe* application. Some users noted a confusion about which items were *actions* and which were *selections*. We called this the “noun vs. verb” problem, and it seemed to permeate different aspects of the application. For instance, *CoGe*’s “User” page presents a list of items— such as genomes or experiments— related to a specific workflow. When interacting with a list of files, a number of different actions are possible: A user may want to *open* an item in a new view or they may want to *select* one item or several items on which they would perform some action. That is, they were not sure if the item represented something to do (a *verb*) or something to choose (a *noun*), and this led to errors as users attempting to select an item would inadvertently click on the item’s title text, opening the item instead of selecting it. For the updated version of *CoGe*, we suggested that the developers clarify what items indicate files, datasets, or other objects versus actions that can be executed on the objects.

A more important issue that clouds the *CoGe* web application is, as mentioned above, the tension between flexible functionality and guided usability. Although the original version of *CoGe* encouraged users to visit the *CoGePedia* for tutorials, the front page of the application was clearly geared toward experienced users and was confusing to newer users. Rather than simplify the functionality of the *CoGe* (which would be contrary to the entire goal of their application), we instead suggested that a clearer indication of the available functionality would be useful to users. We made the suggestion that the front page of the site feature a second set of “entries” into the application, alongside the list of tools. These are presented as a clickable list of answers to the question: “What do you want to do?” A user is able to click on a list of the most common tasks as identified by both user surveys and our analysis of user interaction. These common tasks include: “Compare two genomes,” “Load a new genome,” “Load experimental data,” “Run the RNAseq pipeline,” “Blast against *CoGe*’s genomes,” and “Browse/share data you’ve loaded.” These options do not cover all of the functionality available to users, but they do not obscure any functionality either (which is still available via the re-designed toolkit menu), and they provide an easy

way for users to find the tools needed to design a workflow for a particular data analysis.

In addition to following this suggestion, the application developers also decided to move some of their tutorials from the *CoGePedia* onto the front page. Specifically, they chose tutorials that demonstrated how to answer a particular question, rather than providing tutorials that presented an overview of the full functionality of a single tool. While it is unlikely that a video would demonstrate an answer to the exact question a user had, it was felt that this approach would in fact allow a user to more quickly understand the possible workflows through the web application.

VI. FOLLOW-UP STUDY

We conducted a follow-up study with 4 different participants after the suggested changes were implemented. We were specifically targeting new users who were not very familiar with *CoGe*, but who were domain experts familiar with comparative genomics tasks. The subjects included an undergraduate student in biology, a genomics researcher, a bioinformatics IT administrator, and a graduate student in applied life sciences. Because there were changes in the structure of the website, the steps do not line-up precisely with the original study for all tasks, otherwise the follow-up study was conducted in the same way as the original study. However, it is immediately clear that the “ideal” workflows more closely aligned with the paths the users took through the website, and all users were able to complete tasks more quickly. Comparing the results of our follow-up study to our original study, we found that, on average, it took users much less time to complete Tasks 1, 3, and 4. They improved by 51%, 58% and 77%, respectively. Task 2 took almost the same time, on average, in both user studies. However, Task 2 did show a decrease in the number of hints required, and, subjectively, there was less confusion expressed by users during the follow-up study. The most dramatic difference was that our suggested improvements to the web application in every case resulted in users successfully avoiding the loops that plagued even experienced users in Tasks 1, 2, and 3. Furthermore, users required far fewer hints and users were quickly able to get back “on track” when they did make a mistake. Fig. 3 shows a representative workflow visualization

using the new version of the website after implementing our suggestions.

VII. CONCLUSION AND FUTURE WORK

The development of research-oriented web applications that enable effective interactions with “Big Data” is a complex undertaking. Balancing competing goals of functionality, flexibility, and usability in a rapidly evolving ecosystem will continue to be a challenge. We have shown that a relatively straightforward and pragmatic evaluation methodology can quickly and easily highlight the main bottlenecks hampering the timely completion of complex tasks embedded across multiple workflows. Moreover, our visual representation of these workflows provides an effective tool to highlight the main issues with the tasks and to emphasize the divergence between a developer’s expectations and the real-world use of the system. Somewhat surprisingly, even minor modifications to the content, layout, and navigation of a web application produce dramatic improvements. Furthermore, our evaluation methodology provides a means for the developer of the web application to make reasonable guesses as to what conceptual reformulations could be useful to improve the system. The primary focus of our study was on observing (and suggesting improvements for) the overall usability of the web application in supporting the creation of custom workflows. In the future, we would also like to evaluate other aspects of research-oriented web applications, including especially interactive visualization tools that are often featured to aid scientific analysis, many of which are novel or are being applied to very large datasets in novel ways. Although this work focused on an in-depth analysis of a single website, we believe that our initial findings are generally applicable to any website that provides a scientific workflow through complex datasets. Future work will more thoroughly investigate analyses of research-oriented websites in a range of scientific domains.

ACKNOWLEDGMENT

Our research was supported by the USDA Agriculture and Food Research Initiative (AFRI), Grant #2013-67015-21231.

REFERENCES

- [1] E. Lyons and M. Freeling, “How to usefully compare homologous plant genes and chromosomes as DNA sequences,” *The Plant Journal*, vol. 53, no. 4, pp. 661–673, 2008.
- [2] S. Jung and D. Main, “Bioinformatics as a tool for stone fruit research,” *Genetics, Genomics and Breeding of Stone Fruits*, p. 355, 2012.
- [3] C. Zheng and D. Sankoff, “Practical aliquoting of flowering plant genomes,” *BMC Bioinformatics*, vol. 14, no. 15, pp. 1–8, 2013.
- [4] H. Chen and X. Wang, “Crusview: A Java-based visualization platform for comparative genomics analyses in Brassicaceae species,” *Plant Physiology*, vol. 163, no. 1, pp. 354–362, 2013.
- [5] F. McCarthy and E. Lyons, “From data to function: Functional modeling of poultry genomics data,” *Poultry Science*, vol. 92, no. 9, pp. 2519–2529, 2013.
- [6] J. C. Schnable, M. Freeling, and E. Lyons, “Genome-wide analysis of syntenic gene deletion in the grasses,” *Genome Biology and Evolution*, vol. 4, no. 3, pp. 265–277, 2012.
- [7] E. Lyons, B. Pedersen, J. Kane, M. Alam, R. Ming, H. Tang, X. Wang, J. Bowers, A. Paterson, D. Lisch, and M. Freeling, “Finding and comparing syntenic regions among arabidopsis and the outgroups papaya, poplar, and grape: CoGe with rosids,” *Plant Physiology*, vol. 148, no. 4, pp. 1772–1781, 2008.
- [8] H. Tang and E. Lyons, “Unleashing the genome of Brassica rapa,” *Frontiers in Plant Science*, vol. 3, 2012.
- [9] E. Lyons, M. Freeling, S. Kustu, and W. Inwood, “Using genomic sequencing for classical genetics in *E. coli* K12,” *PLoS One*, vol. 6, no. 2, p. e16717, 2011.
- [10] H. Tang, E. Lyons, B. Pedersen, J. C. Schnable, A. H. Paterson, and M. Freeling, “Screening syntenic blocks in pairwise genome comparisons through integer programming,” *BMC Bioinformatics*, vol. 12, no. 1, p. 102, 2011.
- [11] E. Lyons, “Quick and easy 2nd order genome assembly: Syntenic path assembly using CoGe,” in *Plant and Animal Genome XX Conference*. Plant and Animal Genome, 2012.
- [12] M. Bomhoff, “Load your own: Loading and sharing private genomes in CoGe,” in *Plant and Animal Genome XXI Conference*. Plant and Animal Genome, 2013.
- [13] E. Lyons, “EPIC-CoGe: Functional and diversity comparative genomics,” in *Plant and Animal Genome XXII Conference*, 2014.
- [14] B. W. Herr, W. Huang, S. Penumathy, and K. Börner, “Designing highly flexible and usable cyberinfrastructures for convergence,” *Annals of the New York Academy of Sciences*, vol. 1093, no. 1, pp. 161–179, 2006.
- [15] N. C. Goodwin, “Functionality and usability,” *Communications of the ACM*, vol. 30, no. 3, pp. 229–233, 1987.
- [16] J. Heer, J. D. Mackinlay, C. Stolte, and M. Agrawala, “Graphical histories for visualization: Supporting analysis, communication, and evaluation,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 6, pp. 1189–1196, 2008.
- [17] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang, “Recovering reasoning processes from user interactions,” *IEEE Computer Graphics and Applications*, no. 3, pp. 52–61, 2009.
- [18] J. Zhang, Q. Liu, and K. Xu, “FlowRecommender: A workflow recommendation technique for process provenance,” in *Proceedings of the Eighth Australasian Data Mining Conference*, 2009, pp. 55–61.
- [19] H. Guo, D. Tran, and D. H. Laidlaw, “Incorporating GOMS analysis into the design of an EEG data visual analysis tool,” in *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, 2012, pp. 223–224.
- [20] J. Freire, C. T. Silva, S. P. Callahan, E. Santos, C. E. Scheidegger, and H. T. Vo, “Managing rapidly-evolving scientific workflows,” in *Provenance and Annotation of Data*. Springer, 2006, pp. 10–18.
- [21] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale, “Collaborative coupling over tabletop displays,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 1181–1190.
- [22] A. Kobsa, “User experiments with tree visualization systems,” in *Proceedings of the IEEE Symposium on Information Visualization*, Washington, DC, 2004, pp. 9–16.
- [23] M. Tory and S. Staub-French, “Qualitative analysis of visualization: A building design field study,” in *Proceedings of the 2008 Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV)*, New York, NY, 2008, pp. 7:1–8.
- [24] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds., *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1999.
- [25] I. Herman, G. Melançon, and M. S. Marshall, “Graph visualization and navigation in information visualization: A survey,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 24–43, 2000.
- [26] A. Basu and R. W. Blanning, “A formal approach to workflow analysis,” *Information Systems Research*, vol. 11, no. 1, pp. 17–36, 2000.
- [27] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, “Scientific workflow management and the Kepler system,” *Concurrency and Computation: Practice and Experience*, vol. 18, no. 10, pp. 1039–1065, 2006.
- [28] D. Teng, S. Song, H. Yang, C. Ma, H. Wang, and G. Dai, “An approach to visual analysis for task flow management,” *Science China Information Sciences*, vol. 56, no. 5, pp. 1–12, 2013.
- [29] D. Holten and J. J. van Wijk, “A user study on visualizing directed edges in graphs,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 2299–2308.