Better Bioinformatics Through Usability Analysis

Davide Bolchini[1,2*], Anthony Finkelstein[1] , Vito Perrone[1] and Sylvia Nagl[3]

[1] Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, United Kingdom

[2] Faculty of Communication Sciences, University of Lugano, Via G. Buffi 13, 6900 Lugano TI, Switzerland

[3] Department of Oncology, UCL Cancer Institute and Institute of Structural and Molecular Biology Division of Biosciences, UCL, Gower Street, London WC1 6BT, United Kingdom

ABSTRACT

Motivation: Improving the usability of bioinformatics resources enables researchers to find, interact with, share, compare and manipulate important information more effectively and efficiently. It thus enables researchers to gain improved insights into biological processes with the potential, ultimately, of yielding new scientific results. Usability 'barriers' can pose significant obstacles to a satisfactory user experience and force researchers to spend unnecessary time and effort to complete their tasks. The number of online biological databases available is growing and there is an expanding community of diverse users. In this context there is an increasing need to ensure the highest standards of usability.

Results: Using 'state-of-the-art' usability evaluation methods, we have identified and characterised a sample of usability issues potentially relevant to web bioinformatics resources in general. These specifically concern the design of the navigation and search mechanisms available to the user. The usability issues we have discovered in our substantial case studies are undermining the ability of users to find the information they need in their daily research activities. In addition to characterising these issues, specific recommendations for improvements are proposed leveraging proven practices from web and usability engineering. The methods and approach we exemplify can be readily adopted by the developers of bioinformatics resources.

Supplementary Information: additional data about the usability methods and a summary of the results of the usability analyses presented in the paper is provided in a supplementary information file.

1 INTRODUCTION

Usability is a fundamental aspect of the quality of an interactive application. ISO 92491 defines it as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." The primary goal of this paper is to characterise some critical usability problems in web-based bioinformatics resources and to provide recommendations leading to improved design. As the bioinformatics community increasingly relies on the design, sharing and use of web-based resources, it is crucial to address systematically the usability of these web applications and to deliver a more rewarding user experience to researchers. By drawing on the body of knowledge gained over decades of study and practice in the usability engineering community, we can bring about an enhanced level of usability with important benefits to current and future users of bioinformatics systems.

More usable web applications would enable bioinformatics researchers to find, interact with, share, compare and manipulate important information resources more effectively and efficiently, thus providing the enabling conditions for gaining new insights into biological processes. Usability barriers, by contrast, can inhibit a satisfactory user experience and force researchers to waste time and energy in carrying out their everyday tasks. Usability 'breakdowns' can make even apparently simple tasks, although technically accomplishable, impossible to complete for the user. Removing these barriers can bring about an enhanced level of professional productivity. To make a significant step in this direction, this paper presents a characterisation of a first sample of critical usability problems, as they have emerged from two substantial usability studies, and points out recommendations for achieving a more user-centered design. The first study aimed at understanding usability issues related to navigation and information architecture design. It involved the usability analysis of a well-known *browsing-oriented* repository of protein classifications, CATH. The second study focused on analyzing usability problems related to search and consisted of a user testing on three *search-oriented* bioinformatics repositories, namely Bio-Carta, SwissProt and NCBI. The objective of the studies is not to report an evaluation of the individual resources, but rather to use them as a starting point to collect and characterize more general usability issues.

2 RELATED WORK

The focus of usability analysis is to detect, understand, mitigate, and ultimately prevent usability problems. A usability problem is an obstacle to a successful user experience, meaning by "successful" the effectiveness (feasibility) and efficiency (spending

an acceptable amount of time and energy) in carrying out tasks. For example, the fact that it takes several frustrating and time-consuming trials for a user to eventually get the desired result from a search in a biological database may be caused by a number of usability problems inherent in the way the search functionality has been conceived, designed and communicated to the user. There is evidence of an increasing awareness – at least in the research arena – of the need of usability studies in the development of biomedical systems in general (Rose, 2005). There have been some notable efforts to address the usability of bioinformatics systems. The challenge of bringing an increased awareness of usability and user-centered design to the development of bioinformatics applications can be tackled in a number of different ways. The Human-Centered Software Engineering (HCSE) at Concordia University has worked on developing integrated web-based interfaces to popular bioinformatics portals in order to provide integrated access to web resources relevant to a set of typical tasks (Javahery *et al.*, 2004). The Human-Computer Interaction Lab at the University of Maryland is investigating advanced visualisation techniques to access and manipulate large multimedia information sets in biological databases (Hochheiser *et al.*, 2003). Tackling the work context of bioinformaticians, Joan Bartlett at McGill University has been investigating the daily activities of bioinformatics researchers in order to derive a list of typical information tasks that entail the use of web-based resources to complete (Bartlett and Toms, 2005). Working in a similar vein, Robert Stevens at University of Manchester has undertaken research aimed at characterising and classifying tasks in bioinformatics and analysing their inter-relationships (Stevens *et al.*, 2001).

Although these contributions cover important aspects of improving the user experience of biological databases little has been done to analyse the underlying *design* characteristics of web bioinformatics resources that can lead to potential usability problems. Tackling design issues identifies the usability problems at their source, and helps to prevent the emergence of problems in current and future applications.

3   METHODS

A wide variety of methods are available for identifying the usability problems in the design of web applications. Among them, the most commonly adopted methods fall into two complementary families: user testing and inspection. In user testing, design properties are evaluated 'in action' by observing a sample of representative actual or potential users using the application or its prototype (Rosson, 2001) (Kuniavski, 2002). Usability inspection methods require expert evaluators, who examine the application in depth, following standards or widely accepted protocols (e.g. checklists of tasks or usability principles) (Nielsen, 1994). For application domains that are relatively mature and stable, usability inspection methods can detect, or rather anticipate, a wide range of problems in a complex system in a limited amount of time. For this reason, inspection methods have more recently achieved widespread use, particularly in industrial environments. User testing, by contrast, is useful for capturing how the application supports users in accomplishing complex tasks, which cannot be easily anticipated during inspection.

3.1   Inspection

In the first study, we have carried out a usability inspection of a well-known "navigation-intensive" bioinformatics website (CATH – Protein Structure Classification - http://cathwww.biochem.ucl.ac.uk) using it as an example to illustrate and characterise some critical usability issues related to navigation and information architecture designs. The intent here is not to report the findings of the inspection of CATH, but to take this application as a starting point for modelling usability issues potentially relevant for other web applications in bioinformatics. The inspection of CATH was conducted on version 3.1.0, released Jan 2007. For the analysis based on inspection, we have used the inspection protocol MILE+ (Triacca *et al.*, 2005), the latest of a set of usability inspection methods originally developed for hypermedia and content-rich interactive applications (Matera *et al.*, 2002), and which leverage common practices in usability engineering. MILE+ provides conceptual tools to anticipate the design breakdowns that have negative impact on the user experience by systematically checking the compliance of the application features with a structured set of usability heuristics. MILE+ offers a built-in library of (82) usability heuristics, coupled with a set of operational guidelines that identify the inspection tasks that must be undertaken. These heuristics address different design dimensions: *navigation* (36 heuristics addressing the usability of the information architecture and navigation mechanisms), *content* (8 heuristics addressing the general quality of the information offered to the user), *technology/performance* (7 heuristics addressing usability issues caused by technological breakdowns), and *interface design* (31 heuristics addressing the semiotics of the interface and the graphical layout).

Each MILE+ heuristic is associated with a detailed inspection protocol consisting of three basic components (see Table 1): (a) the feature or specific portion of the application relevant to the heuristics to be applied; (b) the definition of the potential usability problem; (c) one or more inspection actions (or tasks) to be carried out by the inspector.

Table 1. An example of MILE+ navigation heuristics

Feature	**Navigation within a Topic**
Heuristics	**Orientation clues**
Action	1. Identify an instance of topic in the website
	2. Check whether path visibility is present (Where can I go?): navigate from the home page to the instance of the selected topic and verify whether the path traversed (where have I been?) is communicated.
	3. Check whether status visibility is present (where am I?): navigate randomly within the topic and verify whether the current location within the information architecture is communicated.
	4. Check whether context visibility is present: navigate randomly within the topic and verify that the information context you are browsing is communicated at every location.
	5. Repeat step 1 thru 4 for other 3 instances of the topic.

The concepts and vocabulary of MILE+ heuristics are based on common concepts of information architecture and hypermedia design. In particular, the ontology underlying MILE+ borrows from well-known and main stream usability engineering methods (Rosson, 2001) and from IDM (Interactive Dialogue Model) (Bolchini *et al.*, 2006), a lightweight design model for content-intensive interactive applications..

The reliability and strength of the heuristics rely on the fact that they have been collected and iteratively refined over the years as a crystallization of the experience of website usability experts and through a constant

alignment to state-of-the-art usability guidelines and patterns (Van Duyne, 2002). In this study, MILE+ heuristics have been used by one inspector on CATH to *detect* usability problems (when a heuristic is infringed), to *characterise* them (describing the various design aspects involved in the problem), to investigate the potential *causes* and to *specify* recommendations for improvement. The overall inspection process took 15 person-days to complete. Some of the representative classes of problems discovered are reported in the next section. The problems have been described in a more general and abstract fashion, to illustrate their potential relevance to a wide range of other bioinformatics web-based resources.

3.2 User Testing

To capture a wider range of potential usability issues, we have carried out a second study - through user testing – on three large, well-known "search-intensive" repositories: BioCarta (www.biocarta.com), Swiss Prot (www.expasy.ch/sprot) and NCBI (www.ncbi.nlm.nih.gov). The user study aimed at analysing how users interact with and use well-known bioinformatics repositories to fulfil search tasks in the context of a typical research situation they might face in their work. It is important to note that the purpose of the search-related usability study was to investigate the nature of the usability problems encountered while searching bioinformatics repositories, and not to perform a comparative usability evaluation across the three web repositories.

We have recruited 10 subjects across various fields related to the bioinformatics, biology and medical domain. We included users whose expertise with bioinformatics repositories ranges from beginners to intermediate with 2 to 8 years of experience in the relevant biological field (see Supplementary Information for more details about the characteristics of the users). All users were unfamiliar with one of the web repositories to be evaluated (BioCarta), while they had already used the other two repositories (SwissProt and NCBI) at least once.

As is typical in user testing activities, one detailed scenario has been elaborated, by the usability experts together with a group of domain experts (bioinformaticians), in order to frame the tasks assigned in a plausible and, hopefully, motivating setting, meaningful to the subjects. The scenario was designed to be as close as possible to a realistic situation that might occur in the work activity of the researchers.

The scenario includes a contextual narrative, two goals for the researcher and, in the frame of these goals, 5 specific search actions (tasks) to be undertaken on the repositories during the test. Each task asks the user to find specific information using one or more repositories. Each task was directed to the exploration of one of the three resources, although its execution involved a peripheral use of other 'ancillary' websites outside the design of the study (e.g. to retrieve information useful to formulate the query).

The test was not aimed at specifically evaluating the usability of each system *per se* (for which a systematic inspection or a more focused user testing would be more appropriate) but rather at assessing how the accomplishment of a complex, realistic scenario of use is supported by a mix of different web resources, as this is the typical situation occurring in bioinformatics.

The user testing was performed by having a person sitting beside the subject acting as facilitator and observer, posing questions and taking notes when appropriate in order to gain as much insight as possible into the perception and reasoning of the users in trying to complete the tasks ("think aloud" technique). The scenario, goals and tasks are summarized below.

Scenario: A 51-year-old, premenopausal woman comes to the hospital with a newly diagnosed breast carcinoma and a strong family history of breast cancer. A mass in the right breast has been found and an excisional biopsy has revealed a 0.2 cm invasive lobular carcinoma. This type of breast cancer originates from epithelium and is the second most common form of invasive breast cancer.

Goal 1: Understand the role of Oestrogen Receptor (ER). The invasive tumor expressed oestrogen receptor (ER) but did not overexpress HER2/neu. Successful therapies to date include inhibition of proliferation via the oestrogen receptor and HER2 pathways. Use the information you can find on BioCarta to:

Task 1: List three genes whose expression is regulated by ER. *Task 2*: Find the biological function of ER.

Goal 2: Explore the relationship between p53 protein and BRCA2 and the function of telomerase. The patient's family history raised the possibility of a genetic susceptibility to breast cancer through germ-line mutations in BRCA1 or BRCA2. The potential function of BRCA2 is the maintenance of chromosomal integrity as inactivation of this key protein results in an accelerated rate of mutation throughout the genome. Some of the frequent effects of this process are already known, such as mutations in p53 and abnormal expression of telomerase.

Task 3: Use Biocarta to find out what is the downstream effect of p53 in the pathway it participates in together with BRCA2.

Task 4: Use SWISS PROT to find what is the function of telomerase.

Goal 3: Explore the role of Twist protein. Metastasis is a multi-step process during which cancer cells disseminate from the site of primary tumours and establish secondary tumours in distant organs. In a search for key regulators of metastasis in a murine breast tumour model, it was found that the transcription factor Twist, a master regulator of embryonic morphogenesis extensively studied in Drosophila, plays an essential role in metastasis.

Task 5: Use the NCBI portal to discover what is the human protein most highly related to twist in Drosophila.

4 NAVIGATION USABILITY

4.1 Navigating Through Releases

Regularity and consistency are among the key usability principles to consider when designing an effective navigation experience. For large web repositories, however, the complexity of the information and navigation structures being designed and the multiplicity of micro-design interventions over time can cause designers to lose control of what is offered to the user at any given moment.

A bioinformatics web application (such as CATH) can be seen as a federation of different sub-systems (typically related to different underlying databases), which should, ideally, seamlessly cooperate with each other to offer a full range of information services. The type of relations between these sub-systems varies significantly. Some applications are "built over" the data of another one. For example, CATH offers the "Protein Structure Classification" and the "Dictionary of Homologous Superfamilies (DHS)". The latter is derived from the data provided by the former. Other systems, although independent, cross-reference their data, to provide richer information on the same objects. See, for example, how CATH relates to GENE3D or PDBSum.

In this heterogeneous setting, each system is periodically updated and enriched as new data become available from researchers. At a given point in time therefore, a bioinformatics web application appears as an articulation of different sub-systems, each at its own *release state*. The resulting situation is that the user, accessing a web application, navigates through a composition of different subsystems, jumping back and forth over a "release timeline". The communication of these shifts on the "release timeline" has to be carefully designed, so that users can easily realize where they are, what content they are browsing and at what release status. If this is not done, users may be led unwarily to visit out of date content.

Doing the usability inspection on CATH, and in particular using the MILE+ heuristics "Consistency in Overall Navigation" we have discovered navigation breakdowns that stem from this design issue. CATH homepage is linked to the Protein Classification (in its latest release) and to the Dictionary of Homologous Superfamilies (DHS). The latter, however, is built on the data of a *previous release* of the protein classification. Let us assume now that the user moves from the homepage to the DHS. From here, the user sees a link (labelled "Top of the Hierarchy") promising to lead to the Protein Classification page. That link, however, unexpectedly takes the user to the homepage of the previous release of CATH which the DHS is built upon. What happened?
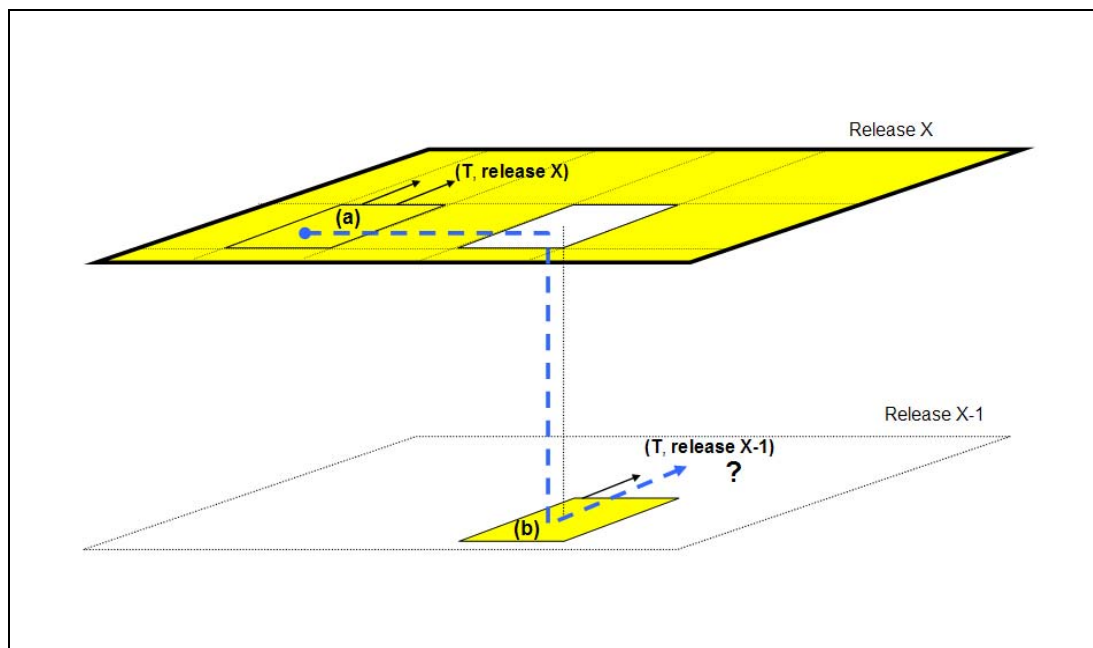
4.2 Example

Fig. 1. Falling into a subsystem in a previous release and following its navigation takes the user to non-updated content.

Let us characterize the problem in more abstract terms. Let us assume (Figure 1) that users are navigating in a given subsystem - called (a) - of the overall application, which is in its latest release ("X"). From "a" there are links towards other sub-systems, such as T, which is at the same release of (a). As users navigate, however, from (a) to another sub- system (b), they "fall" in a previous release status, because (b) follows a different release policy and is currently updated up to a previous release (X-1). Unexpectedly, as users try to reach T from (b), they are not taken to T in its latest release, but rather to T in the release aligned with (b). If the users do not realize this "release shift" happened, they would continue the navigation in a non-updated system, thus accessing non-updated data. And recursively, from here, for the same design flaw, a chain of similar problems can happen, continuously shifting users across different release status of the same content.

The consequence of this problematic design is that the navigation connections which should be, and are perceived to be, constant across sub-system boundaries, unexpectedly change behavior according to the release status of the content they start from. To understand the design issue in detail, it is important to note the difference between local and global navigation. In the case of local navigation, i.e. navigation among the different parts of a given subsystem, it is acceptable to move within a previous release. In the case of CATH, navigation to different parts of the DHS obviously takes the user to content updated at a release status aligned with the DHS. In case of global navigation, however, i.e. moving from one subsystem to another, the unexpected "permanence" at a previous release (as a more current one is available) brings the user to access obsolete content.

If the release status of the *current and target* content of the navigation is not clearly and constantly communicated to the users (especially when shifting across subsystems), users risk being unexpectedly taken to non out of date content (with the risk of not even realizing it). As researchers access out of date information, they risk using obsolete information for their re-

search work and missing the opportunity to exploit the full potential of the content available.

As a recommendation to designers, it is important to consider that in shifting between two subsystems the user should be able to answer the following questions: Which release am I in? Which release can I navigate to from here? Am I moving to another release? How can I get to the latest release?

Various design solutions can be conceived to meet these requirements. An important design recommendation is to systematically ensure that *all* outgoing links from a subsystem in a previous release (which are typically the sections which tend to go out of control over time) are targeting the desired pages (i.e. the most updated ones). Besides, the communication of release change can be anticipated in many ways at the link level (through link comments and mouse-over tags) by enriching the semantics of the link with release information (e.g. Go to DHS – Release 2.6). Finally, the content being navigated can be clearly and constantly labelled with the current release status it is in, by means, for example, of richer page title information. These design recommendations are based on proven patterns of information scent design (Spool, 2004), which have demonstrated an enhanced level of usability where applied. In the new version of CATH (see Figure 2), currently under development, the design team has solved the usability problems connected with the co-existence of different releases of the application subsystems.
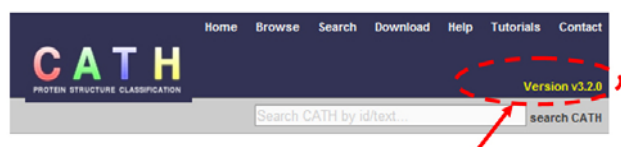

Fig. 2. An excerpt of new CATH redesign.

Page navigation has been redesigned using templates, that maintain navigation consistency across the entire application. In each template (and therefore in each page instance of the same template), an explicit indication of the current release status is communicated to the user.

4.3 Rigid Access Structures

Access Structures are navigational paths which are critical for finding the information offered by the application. They comprise all those design elements which are not *per se* pieces of content but whose purpose is to enable one to locate and reach the application content. We can easily identify two primary access structures available to the user in CATH: a hierarchical classification to access the protein domains, and an index to access the DHS data. The usability of the access structure strongly depends on the design of the proper criteria available to the user to locate and access information. Capturing the user's "way of reasoning" or the desired strategy to look for information is key to providing effective access structures, which support the user to find content in a natural fashion. A typical usability problem of access structures is their "rigidity", that is the availability of a limited number of access paths and the lack of those the user may need when attempting to locate a piece of content with a specific criterion in mind.

4.4 Example

During our CATH usability inspection with MILE+, we have discovered a limit in the access structures particularly with respect to the ways to access the DHS data. The access to the homologous superfamilies is organized as a huge index to all the 1459 superfamilies. For length reasons, this index is split into 98 pages sequentially linked, where each index page displays 15 items (Figure 3).

This large index of 1459 superfamilies is ordered by superfamily code (e.g. 1.10.10.250 before 1.10.10.350). Plus, there is the possibility of using a "local" search engine, specifically devoted to DHS search. Given these access structures, when looking for a specific superfamily name (e.g. Urate Oxidase), without knowing the code, this design poses substantial obstacles in accomplishing the task. Users have no way to retrieve the superfamily details by name efficiently. They have to browse 49 pages one-by-one, and in each page read a list of 15 items to look for the desired superfamily. During inspection, we have optimistically estimated that if a user takes 20 seconds to carefully scan an superfamily index page, the overall time necessary to locate the superfamily "Urate Oxidase" by name – which is proportional to the number of index pages the user has to scan – is 17 minutes (page 51 out of 98).
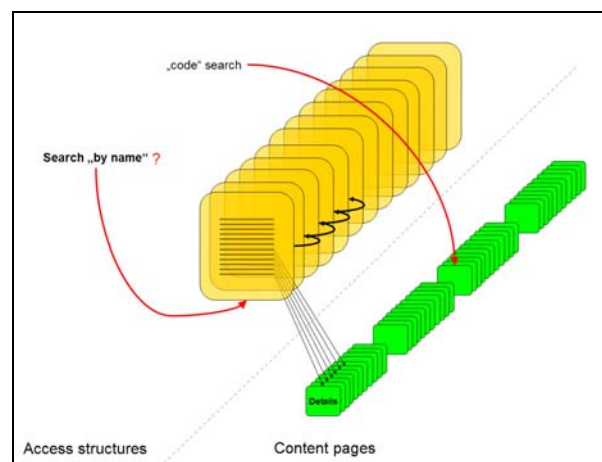

Fig. 3. An example of rigidity of the access structures when trying to locate an item by name.

Overall, it is clear that this access structure is some way from supporting an efficient retrieval of the desired superfamily, because it does not give the possibility to the user to find superfamilies by a natural criterion (their name). The fact that designers wanted to capture browsing "by name" is demonstrated by the fact that they organized the access to the supefamilies by an index clearly showing the name as a primary and well visible attribute (besides the code). Simple requirements for improvement can be recommended to overcome the described problem. First of all, multiple criteria to sort the index of superfamilies could be offered (e.g. by name, by code, etc.). Mechanisms to sort superfamilies by alphabetical order (of the name) can greatly speed up the retrieval process when trying to locate a specific superfamily by name.

Various design solutions are available to meet these requirements, which are not discussed here in detail and are under

evaluation for the current redesign of CATH. Just to pick an example, links pointing to sets of superfamilies grouped by name initial (A to Z) or to a set of possible initials (A-D, E-H, I-O, etc) would enable browsing the index more efficiently. Finally, the local search engine devoted to the DHS search should support the keyword search by superfamily name. As to the usability of the search engine, however, the next section will highlight more general usability aspects to consider, based on the results of a further usability study.

5 SEARCH AND THE USER EXPERIENCE

While browsing is a suitable access paradigm for exploring and interacting with structured information collections based on controlled taxonomies, search is often the first choice for users and is typically used to support complex tasks in less-structured and larger information repositories. To start understanding the nature of the usability problems involved in search tasks, we have carried out a user testing using three large bioinformatics web repositories: BioCarta (www.biocarta.com), Swiss Prot (ftp.expasy.org/sprot) and NCBI (www.ncbi.nlm.nih.gov).

5.1 Results and Discussion

The limited number of tasks and users involved in the study precluded an entirely satisfactory meaningful quantitative analysis of the results. A summary of the quantitative results of the user testing, however, is reported for reference in the Supplementary Information. A qualitative analysis of the user experience problems, however, did produce valuable insights. In fact, the analysis of the test logs revealed that users found significant usability barriers to fully accomplish their tasks. The nature of these problems concerned two aspects: *formulating search query* and *interpreting search results*.

When *formulating a search query*, all users had problems in hitting the "right" terminology supported by the search mechanisms offered by the repositories. Note that this emerged from all users across all the three applications. A recurring obstacle to an effective search formulation is that the system recognizes *only one* spelling for a given information element (e.g. estrogen), even in cases where other spellings are commonly used among researchers and in the literature (e.g. oestrogen, œstrogen). can be readily imagined, when the user searches using an unsupported spelling, the system returns no results (without further explanation or hint). Observing and soliciting the users to "think aloud" while performing the test, we have noticed that most of users did not realise that the cause of "no result" is the unsupported spelling. Some of them left the repository and tried another one which they were more familiar with. An extreme case of the same problem occurs when a repository forces the use to search using a repository-specific object identifier (e.g. in SwissProt) in order to get relevant results. During user testing, the consequences of this problem have been of two kinds: first of all, the users do not realize - even after several trials - that a specific object ID needs to be used to yield accurate results. Secondly, even when they such an object ID is required for the search, they do not have any idea where and how to find it in order to formulate the query.

Observing the users struggling in formulating an appropriate search query, we have noticed a pattern of iteratively "narrowing" and "broadening" the search space. In fact, when trying to formulate more complex queries and constraining the search scope, all users in at least one task tried to combine two keywords together. However, they all have problems in finding the correct, supported syntax to use. Some tried with "AND", others with "+", others by simply juxtaposing keywords (but not being confident in doing in that) or apexes. In all cases, the results of the combined keywords search did not yield better results. An alternative search strategy adopted by most users and well supported by the three repositories involved, is the explicit *a priori* definition of the search scope for the query. The search scope is the domain of data to which the search is narrowed. For example, NCBI explicitly enables the user to scope the search to Proteins, Structure, Genomes, CancerChromosomes, and many other domains. Although this possibility is present, in some cases the user had difficulties in properly using it and it took several trials to yield accurate results. The main problem lies in the way this functionality is communicated to the user. SwissProt, for example, uses names of databases to communicate the search domains: SwissProt/trEMBL, SwissProt/tremble (full), SwissProt/tremble (beta), PROSITE, NWT-Taxonomy, SWISS-2D Page, just to name a few. Instead of being able to select the "content domain" to search for, the user is faced with a list of technical names of databases they may not be familiar with.

To systematically address these issues and provide recommendations for improvement, it is necessary to abstract from the specific occurrences of the problem, as they have been observed, to more general usability issues in search design. Although designers can assume that search is primarily executed by users who have a good knowledge of the domain (e.g. proteins) and are looking for specific information, they cannot at all assume that all users know in advance how the system expects them to carry out the search. Three crucial concerns need to be addressed to support the user when approaching a search task:

1. What content base am I searching over (*search scope*)? Which portions of content are covered by the search? For example, can I search protein domains, chains or genomes? An explicit communication of the information domains to which the search can be scoped is a great help to enable users to more efficiently retrieve relevant and accurate results. To indicate the search scope, a plain, widely-accessible terminology referring to the *content* to be searched must be used and transparently communicated to the user. Even recurrent users might find problems in understanding technical or too specific terms.

2. Which vocabulary should I use to query the system (*search ontology*)? For example, should I use the names of the protein or their codes, or both? Which code system should I follow? From this perspective, given the ample fragmentation of the knowledge domain, automatic alternative spelling recognition should be systematically supported. Design solutions may vary. For example, when searching for "oestrogen", in case that "estrogen" is supported, a Google-like message can display "Do you mean "estrogen"?", which would lead to automatically search for "estrogen". An extensive use of search mechanisms relying on ontologies for synonyms and alternative spelling is encouraged to improve

the search experience. SwissProt, for example, makes an attempt to do so by capturing synonyms of protein names.

3. Which syntactic formulation should I use to express my query best (*query syntax*)? Are logical operators supported for more articulated query? Which ones? How do I express them? To support this need, the explicit communication of examples of queries is very useful. This would save the user time spent trying to guess the supported syntax for more complex queries (most users have tried "AND"-like query formulations).

The usability study we have conducted has shown that, if not properly addressed, these questions cause users – even when strongly motivated and committed – to run into a frustrating loop of "trial and error" search queries, in the attempt to guess the answers to those questions and eventually get useful results.

When *interpreting search results*, the main problem encountered by the users was the difficulty in managing long lists of results. A list of more than 100 results has occurred in most of the formulated queries in all tasks. Users typically have adopted two alternative reactions: a) intimidated by the long list of items, they do not explore further and try to reformulate the query; b) they focus on the first, second or third results, hoping the first few results to be the most relevant ones (which is not always the case). In the former situation, the user encounters the formulation problems discussed before. In the latter, the user misses potentially relevant results which are displayed further on the list. Given the nature of the domain and the search tasks, retrieving large sets of results is a typical and expected situation in bioinformatics. Our user study shows, however, that the way in which these long lists of results are designed intimidates users rather than encouraging them to further explore the repository. The main design problems emerged are three.

First of all, users found it difficult to make sense of the whole set of results at a first glance. The visual organization of the result items makes the search results appear confusing to users, and does not guide their eyes to easily master the complexity of the results at a glance. A more user-centred design of the search results would entail a re-conception of the layout in which the results are displayed (properly spacing and highlighting key elements) not to overwhelm the user in trying to get an overview of the large set of elements retrieved.

Secondly, users do not understand the ranking criterion used for ordering the documents found. They hope the order is based on relevance, but how this relevance is obtained is not communicated to them and they do not find the elements to understand it. Without having a clue of the order criterion used, users feel helpless in formulating a strategy for reading the large set of results. To address this problem, it is important to explicitly communicate to the user the actual ranking criteria used for displaying the results (as results are displayed) and, possibly, to allow sorting the obtained results by multiple, additional attributes (e.g. by publication/release date, by alphabetical order).

Finally, all users found severe obstacles in understanding the actual content of the documents (before clicking on it) on the basis of the representative information appearing in the list (e.g. the document title or the document body excerpt did not help much in identifying relevant content). In some cases, users found that specific symbols were associated with each list item

and could not find any explanation for them. For example, letters "H" and "M" are displayed aside each BioCarta search result without any legend or textual clue.

6 CONCLUSIONS

The usability of web bioinformatics applications can be significantly improved by leveraging proven practices in usability and web engineering.

This work is the first step of a wider effort aiming at surveying a larger number of biological databases to collect and characterize the typical design breakdowns which pose severe obstacles to a successful user experience. Future work will comprise a more complete and deeper characterization of the typical contexts and usage scenarios of bioinformatics repositories. This is essential to understand the role that web resources play in the daily work of bioinformatics research and to analyse further the nature of the usability problems as they emerge from the natural context of use. This will potentially lead to discovery of more opportunities for improvement and to a general framework of usable design recommendations that can be used by the bioinformatics community at large.

The ultimate goal is to make available proven design patterns (i.e. proven solutions that work) and conceptual tools in order to promote a more aware human-centered development process of bioinformatics applications.

REFERENCES

Bartlett,J.C. and Toms,E.G. (2005) 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.

Bolchini,D., Paolini,P. (2006) Interactive Dialogue Model: a Design Technique for Multi-Channel Applications. *IEEE Transactions on Multimedia*, 8, 3, 529-541.

Bolchini,D., Garzotto,F. (2007) Quality of Web Usability Evaluation Methods: an Empirical Study on MILE+, in *Proceedings of WISE (Web Information Systems Engineering) 2007*, Workshop on Web Usability and Accessibility, Lille, France, Springer.

Hochheiser,H., Baehrecke,E.H, Mount,S.M. and Shneiderman,B. (2003) Dynamic Querying for Pattern Identification in Microarray and Genomic Data, in *Proceedings of 2003 IEEE International Conference on Multimedia and Expo*.

Javahery,H., Seffah,A. and Krishnan,S. (2004) Beyond Power: Making Bioinformatics Tools User-Centric. *Communications of the ACM - Special Issue on Bio-Informatics*, 47, 11, 58-63.

Kuniavski,M. (2002), Observing the User Experience – A Practitioner's Guide to User Research, Morgan Kauffmann.

Matera,M., Costabile,M.F., Garzotto,F., Paolini,P. (2002) SUE Inspection: An Effective Method for Systematic Usability Evaluation of Hypermedia, *IEEE Transactions on Systems, Man, And Cybernetics - Part A: Systems And Humans*, 32, 1, 93-103.

Nielsen,J., Mack,R.L. (1994), Usability inspection methods, John Wiley & Sons, New York.

Rose,A.F., Schnipper,J.L., Park,E.R., Poon,E.G.; Li,Q., Middleton,B (2005) Using qualitative studies to improve the usability of an EMR, *Journal of Biomedical Informatics*, Volume 38, Issue 1, 1 February 2005, 51-60.

Rosson,M.B., Carroll,J.M. (2001), Usability Engineering: Scenario-Based Development of Human Computer Interaction, Morgan Kaufmann.

Spool,J., Perfetti,C. (2004), Designing for the Scent of Information, www.uie.com.

Stevens,R., Goble,C., Baker,P., Brass,A. (2001) A classification of tasks in bioinformatics. *Bioinformatics*, 17, 180-188.

Triacca,L., Inversini,A., Bolchini,D. (2005) Evaluating Web Usability with MiLE+, in *Proceedings of Seventh International Symposium on Web Site Evolution (WSE 2005)*, Budapest, September, 22-29.

Van Duyne,D.K., Landay,J.A., Hong,J.I (2002), The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience, Addison-Wesley Professional.