W16–W20 *Nucleic Acids Research, 2007, Vol. 35, Web Server issue*
doi:10.1093/nar/gkm280

CARGO: a web portal to integrate customized biological information

Ildefonso Cases[1], David G. Pisano[2,3,*], Eduardo Andres[2], Angel Carro[2], José M. Fernández[3], Gonzalo Gómez-López[2], Jose M. Rodriguez[3], Jaime F. Vera[1], Alfonso Valencia[1,3] and Ana M. Rojas[1]

[1]SCOMPBio Group, [2]Bioinformatics Unit (UBio), Structural Biology and Biocomputing Programme, Spanish National Cancer Research Centre (CNIO), Madrid, Spain and [3]Spanish National Institute for Bioinformatics (INB), Spain

Received February 5, 2007; Revised April 5, 2007; Accepted April 11, 2007

ABSTRACT

There is a huge quantity of information generated in Life Sciences, and it is dispersed in many databases and repositories. Despite the broad availability of the information, there is a great demand for methods that are able to look for, gather and display distributed data in a standardized and friendly way. CARGO (Cancer And Related Genes Online) is a configurable biological web portal designed as a tool to facilitate, integrate and visualize results from Internet resources, independently of their native format or access method. Through the use of small agents, called widgets, supported by a Rich Internet Application (RIA) paradigm based on AJAX, CARGO provides pieces of minimal, relevant and descriptive biological information. The tool is designed to be used by experimental biologists with no training in bioinformatics. In the current state, the system presents a list of human cancer genes. Available at http://cargo.bioinfo.cnio.es

INTRODUCTION

With the implementation of large-scale analysis initiatives, the amount of information in terms of biological data availability is overwhelming, as reflected by the hundreds of databases (1) and web servers (2) described in the literature. This number is expected to grow year on year, increasing the size of resources available to the experimental research community. These resources have a great value to scientists for proposing novel hypotheses and delineating further research, but are sometimes difficult to access due to both usability and maintenance issues.

In addition, without a strategy to efficiently extract it, the information may become unusable as data availability increases. Efforts to implement integration and standardization have been developed in different frameworks. To mention a few, the BioMOBY (3) (http://www.biomoby.org) and DAS (4) (http://www.biodas.org) projects aim to leverage the retrieval and integration of biological data served from distributed resources at the machine level through commonly agreed public conventions. The open approach and the potential of those initiatives have been well accepted by the scientific community (5), but still substantial improvement is needed to present them to end users. The creation of customized bioinformatics environments would greatly facilitate end user interaction with the information, and would enable its effective use. This is particularly important when the end users are experimentalists with no training in Bioinformatics. The task of developing systems for this type of user constitutes a challenge for developers and individuals trained in computing and bioinformatics.

The current trends to create specialized user interfaces fall into two different approaches: data aggregation, or super-specialization. Examples of the first approach are illustrated by the main biological data repositories like NCBI (6) (http://www.ncbi.nlm.nih.gov/) and Ensembl (7) (http://www.ensembl.org/), or by web servers like Harvester (8) (http://harvester.embl.de/), where exhaustive information about a queried entity can be retrieved and presented simultaneously to the user. The aggregation concept is an approach that suffers from certain issues: although it simplifies the task of searching over multiple disparate resources by providing a unique entry point, it may add complexity to the analysis of the results. This is in part caused by inherent problems such as representing heterogeneous findings into a single result format, or eliminating data redundancy if the same piece of information is retrieved from more than one source.

Certain super-specialized servers address some of these issues with strategies like presenting only domain specific data obtained from selected sources (9)

*To whom correspondence should be addressed. Tel: +34-91-224-6900; Fax: +34-91-224-8006; Email: dgpisano@cnio.es

(http://www.efamily.org.uk/software/dasclients/spice/). However, the high customization required for a particular task produces a lack of flexibility. The user might deal with complex data from multiple origins and this creates a myriad of specialized applications.

CARGO represents a new generation of visualization tools that aims to circumvent these two problems. In this work we present a system that aims to facilitate the visualization and the analysis of biological data. Its philosophy is to address the problem of aggregation by presenting only slices of *core* information in a unified interface, and leaving up to the user whether to search more details by providing links to the original source.

The system uses a gene-based querying procedure that activates a number of small software agents (called widgets) to retrieve, relate and display concise, accurate and relevant information. As CARGO deals with disparate sources, and allows a great technological diversity, the problem of super-specialization is also addressed. In its current state, it provides updated information obtained from reliable data sources such Ensembl (7), CPATH (10), dbSNP (6), PDB (11), OMIM (12) or iHOP (13) via DAS (4) or Web services (3).

THE SYSTEM

The system has three main modules. The first one constitutes the *query module* (Figure 1a), where the user can search Ensembl (v41—October 2006) (7) gene descriptions by either writing a free text option, or by choosing results from a precompiled lists of genes. To illustrate this, we have included a list containing cancer candidate genes for breast and colon cancer (14) as an example. The second part of the system is the *core module* (Figure 1b), which connects the results, in the form of a normalized gene list, to widgets. When the user chooses a gene in the results list, this module broadcasts the gene identifier to the open widgets, and activates their internal logic. The last element is formed by the *widgets* themselves, which retrieve and assemble the information (Figure 1c) in a process transparent to the user. A widget is just a small web page, designed to show information in a visual and concise manner. This web page is produced by a server-side program (a CGI, for instance), which is called by the core module via an HTTP GET request, passing an Ensembl identifier as the only parameter. The server-side part of the widgets that retrieves and displays the information is technology-agnostic, and currently implements many technologies including DAS (4), specialized APIs, specialized Web services (3), and other web

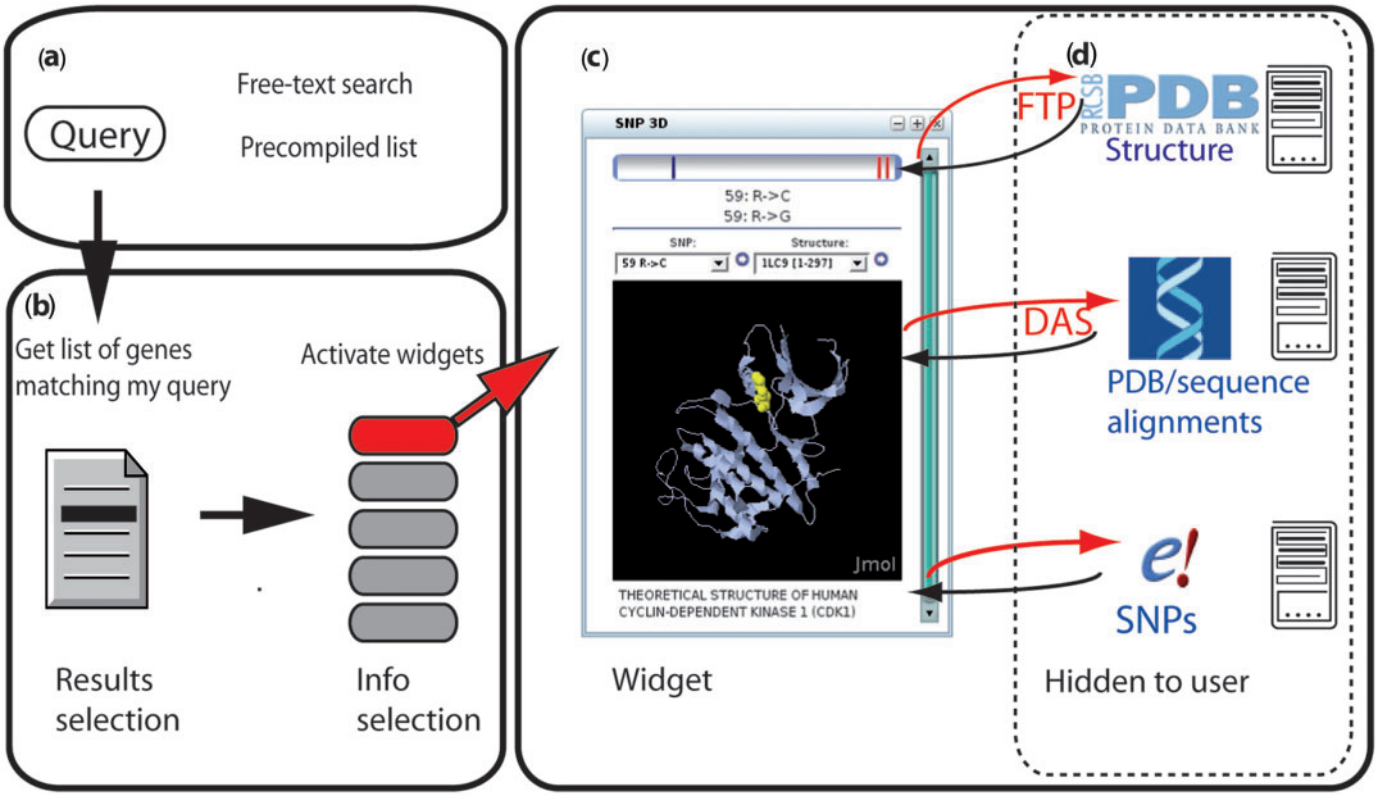

Figure 1. The Cargo System. (**a**) Query module: available options are free-text search and pre-compiled lists of genes. (**b**) Core module: retrieves potential genes matching the input query. The user selects a gene from the results obtained, and the system dispatches the information to all the available widgets, which refresh accordingly. (**c**) Widgets: The widgets are activated by the user. Each widget provides concise information at different levels. The technologies and procedures used by the widget are hidden from the user. (**d**) In the example, the system fetches PDB files using FTP, then extracts information about sequence/structure alignments from The Sanger Institute (http://das.sanger.ac.uk/das/msdpdbsp/) using DAS (4), and conducts queries about SNPs using the Ensembl Variation API (http://www.ensembl.org/info/software/variation/index.html).

technologies like RSS. Since the widgets are called by standard HTTP requests, any widget can be hosted remotely with respect to the core module. Thus, third-party developed widgets can easily be included into CARGO. Developers willing to display their widgets in CARGO, need only to provide a URL, and make their server-side program respond to the HTTP GET request '?ensemblid=%', where '%' means the Ensembl identifier selected by the user. The core module in CARGO calls the URL of the opened widgets, and loads the valid HTML document provided by the widget into its window.

We have developed a set of widgets that addresses different biological problems to represent the concept of CARGO's widget implementation. The *literature mining widget*, or iHOP widget (Figure 2), is the CARGO implementation of the popular iHOP tool (13), which provides information from PubMed abstracts. The widget uses Web services technologies (José M. Fernández, in press) to query iHOP. The output is a comprehensive collection of sentences mined from the literature, either defining a gene or its interactions with other genes. *The 3D Coding SNPs* (Figures 1c and 2), maps the non-synonymous SNPs found in proteins onto PDB structures (11). The widget retrieves information from dbSNP database using Ensembl APIs, and maps the gene sequence to retrieved PDB structures using a DAS alignment server (http://das.sanger.ac.uk/das/msdpdbsp), but the process is transparent to the end user. The *Disease Information* module (Figure 2) is the widget implementation of the OMIM (12) database, providing summarized information about diseases associated to the gene. The *Interactome* widget extracts information from the protein–protein interactions database cPath (10), along with experimental evidence of the interaction and literature references. The *Transcript annotations* widget shows functional predictions for the query using the FunCUT pipeline (15). Due to the large computational resources used by this pipeline, predictions have to be pre-calculated.

USAGE

CARGO is designed to be extremely simple and clear to use. To best show its capabilities we propose biological questions and answer them on-the fly (Figure 2, background coloured boxes).

We can imagine a user asking a simple biological question: *How many SNPs of P53 are coding and can be*

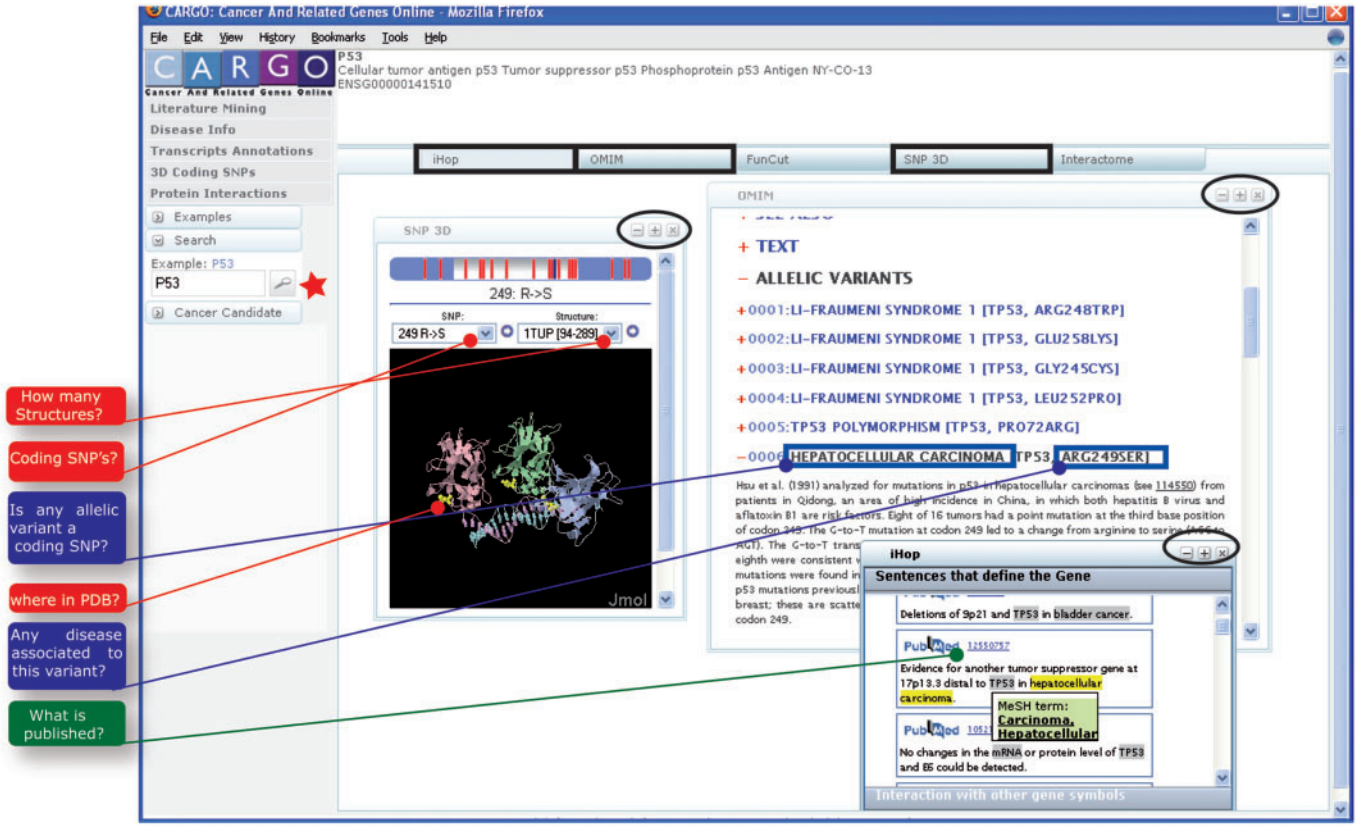

Figure 2. Use of Cargo. The left frame shows the query interface. As an example, a search for 'P53' (red star) is shown. At the top, a small description is provided to the user. The rest of the working space shows the different widgets. Black boxes indicate activated widgets (iHOP, OMIM and SNP 3D). Black circles show the widget interface for visual configuration. By activating the SNP 3D widget, information about coding SNPs, structures and correspondence between them is provided (red lines). The OMIM widget provides information regarding allelic variants and diseases (blue lines). In the example, it links a specific allelic variant to hepatocellular carcinoma, which can then be visualized in the structure. The iHOP widget provides information from literature in the context of sentences. Therefore, the terms used by the query and other important labels (such as MeSH terms), are also highlighted (green pop-up, showing 'Carcinoma, hepatocellular' in iHOP widget) and a link to the original paper is provided.

mapped onto solved structures? With the current tools available, the user would need to navigate at least two different Internet resources, PDB and dbSNP, to retrieve the information about the structure and the variations, respectively. Once this step is done, the user would have to manually find the correspondence between these two resources, which means mapping the sequence to the structure. A further step would require determining the localization of the variation mapped onto the correct structure, which is itself a tricky matter. To do this, the user would have to download and learn to use a specialized viewer.

Using CARGO, the user just searches for 'P53' (Figure 2, red star) and selects this gene in the results list. Since the SNP-3D widget integrates information from dbSNP, PDB, structure/sequence equivalences and a visualizer, all of the aforementioned steps are addressed at once (Figure 2, red background boxes). This widget exemplifies the integration level design concept in CARGO: the user can access three different kinds of information and integrate them in a minimal working space, because special attention has been paid to visually represent all the relevant data in a concise way. Additional visualization tools facilitate the interpretation of the results. For instance, the blue top bar in the widget represents the length of the protein and the grey area indicates the structural coverage in the sequence by the selected structure (in this case, 1TPU). The red vertical bars indicate all the residues mapped as SNPs. By choosing a particular SNP either in the bar or using the menus, the user can view the SNP mapped onto the structure (yellow balls in the structure). The user can select alternative structures and also links are provided to the original sources if more detailed information is desired.

Once the coding SNPs are mapped into the structure, the user could ask *how many of these coding SNPs are defined as allelic variants?* To answer this, the user should navigate the OMIM database and read all the free text. By activating the OMIM widget in CARGO, a small window shows all the results for the P53 gene. Searching in the allelic variants section, we can compare them with the SNPs shown by the SNP-3D widget. For instance, the allelic variant ARG249SER is a coding SNP (Figure 2). We can ask still further questions, such as *is this allelic variant associated to any disease?* As seen in the OMIM entry, it is linked to hepatocellular carcinoma.

One straightforward task would be to retrieve literature regarding this disease or this allelic variant. The usual way would be to navigate and perform advanced text searches in PubMed. These searches usually provide lots of unwanted results. In CARGO, activation of the iHOP widget brings up a whole list of references involving the protein. Again, the visual aspect of the widget emphasizes and highlights the relevance of the term in the context of the sentence. Here, an entry relating the p53 with hepatocellular carcinoma is shown, along with links to the original article. By activating the other widgets, the user can display alternative information. Any widget can be minimized, displayed or hidden (Figure 2, black circles).

SUMMARY

CARGO is capable of integrating information at different levels, ranging from visualization to mapping. The main strength of CARGO, as compared to alternative tools (8) (http://harvester.embl.de/), is its 'go-to-the-core' capabilities. At the same time it aims to facilitate the interpretation and retrieval of information in a second-generation visualization framework. It features technology-agnostic widget creation, which allows the inclusion of as many disparate resources as desired to completely address a range of questions.

By definition, CARGO is an open platform, since it can be extended just by the addition of new widgets and searching modules. At the same time, the usefulness of CARGO depends strongly on the diversity and quality of the available widgets. We are already developing new widgets to mediate further information, such as the visualization of protein annotation at the residue level, or the visualization of the tissues in which a gene is expressed, as reported by the GNF Gene Atlas (16). Additional query modules are also under development, including one that allows the graphical generation of gene lists from genome coordinates. We hope CARGO will grow by community effort, so we are preparing documentation to facilitate other parties to develop their own widgets or searching modules, which can be easily integrated into the CARGO structure. Given the simplicity and flexibility of the CARGO concept, we expect that many of these widgets will be incorporated in the near future.

SUPPORTED PLATFORMS

A selected community of experimental researchers is currently testing the system. CARGO requires Firefox 1.5 and Java 1.4.1, and has been tested in Windows, Mac OS X and Linux. Additional support for alternative browsers is currently in progress.

CONTRIBUTIONS

I.C., J.M.R., J.M.F. and J.F.V. developed the widgets. A.C. and E.A. programmed the search modules. A.C. and J.F.V. worked out the main framework. J.F.V. designed the website. G.G.-L. provided scientific support and wrote the documentation. I.C., D.G.P. and A.M.R. developed the general concept and coordinated the project. D.G.P., A.M.R. and A.V. assembled the manuscript. A.V. did the general coordination.

ACKNOWLEDGEMENTS

I.C. is a recipient of a 'Ramon y Cajal' programme, J.F.V. is supported by EU Marie Curie MIRG-CT-2005-016499, G.G.-L. is funded by the Biomedical Foundation of Vigo Hospitalary Complex (FICHUVI). This work has been partially financed by EU BIOSAPIENS (LSHC-CT-2003-505265) and EU EMBRACE (LSCG-CT-2004-512092), and by the National Institute for Bioinformatics (www.inab.org), a platform of 'Genoma España'.

Funding to pay the Open Access publication charges for this article was provided by EU BIOSAPIENS (LSHC-CT-2003-505265).

Conflict of interest statement. None declared.

REFERENCES

1. Bateman,A. (2007) Editorial. *Nucleic Acids Res*., **35**, D1–D2.
2. Bateman,A. (2006) Editorial. *Nucl. Acids Res*., **34**, W1.
3. Wilkinson,M.D. and Links,M. (2002) BioMOBY: an open source biological web services proposal. *Brief Bioinform*., **3**, 331–341.
4. Dowell,R.D., Jokerst,R.M., Day,A., Eddy,S.R. and Stein,L. (2001) The distributed annotation system. *BMC Bioinformatics*, **2**, 7.
5. Stein,L. (2002) Creating a bioinformatics nation. *Nature*, **417**, 119–120.
6. Wheeler,D.L., Barrett,T., Benson,D.A., Bryant,S.H., Canese,K., Chetvernin,V., Church,D.M., DiCuccio,M., Edgar,R. *et al*. (2007) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*., **35**, D5–D12.
7. Hubbard,T.J., Aken,B.L., Beal,K., Ballester,B., Caccamo,M., Chen,Y., Clarke,L., Coates,G., Cunningham,F. *et al*. (2007) Ensembl 2007. *Nucleic Acids Res*., **35**, D610–D617.
8. Liebel,U., Kindler,B. and Pepperkok,R. (2005) Bioinformatic "Harvester": a search engine for genome-wide human, mouse, and rat protein resources. *Methods Enzymol*., **404**, 19–26.
9. Prlic,A., Down,T.A. and Hubbard,T.J. (2005) Adding some SPICE to DAS. *Bioinformatics*, **21**(Suppl 2), ii40–ii41.
10. Cerami,E.G., Bader,G.D., Gross,B.E. and Sander,C. (2006) cPath: open source software for collecting, storing, and querying biological pathways. *BMC Bioinformatics*, **7**, 497.
11. Berman,H.M., Bhat,T.N., Bourne,P.E., Feng,Z., Gilliland,G., Weissig,H. and Westbrook,J. (2000) The Protein Data Bank and the challenge of structural genomics. *Nat. Struct. Biol*., **7**(Suppl), 957–959.
12. Hamosh,A., Scott,A.F., Amberger,J.S., Bocchini,C.A. and McKusick,V.A. (2005) Online mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res*., **33**, D514–D517.
13. Hoffmann,R. and Valencia,A. (2004) A gene network for navigating the literature. *Nat. Genet*., **36**, 664.
14. Sjoblom,T., Jones,S., Wood,L.D., Parsons,D.W., Lin,J., Barber,T.D., Mandelker,D., Leary,R.J., Ptak,J. *et al*. (2006) The consensus coding sequences of human breast and colorectal cancers. *Science*, **314**, 268–274.
15. Abascal,F. and Valencia,A. (2003) Automatic annotation of protein function based on family identification. *Proteins*, **53**, 683–692.
16. Kent,W.J., Sugnet,C.W., Furey,T.S., Roskin,K.M., Pringle,T.H., Zahler,A.M. and Haussler,D. (2002) The human genome browser at UCSC. *Genome Res*., **12**, 996–1006.