Mobyle : a Web portal framework for bioinformatics analyses

B.Néron[1], P. Tufféry[2], C.Letondal[1]

[1]Institut Pasteur, Paris, France.
[2]Equipe de Bioinformatique Génomique et Moléculaire, INSERM E346, Université Paris 7, Paris, France.

Abstract

We have designed a system, Mobyle, to provide an environment for running and defining bioinformatics analyses. The objective of this environment, a successor of the Pise[3] tool, is to enable biologists to access advanced features, such as pipelines or remote services discovery, yet not having to learn complex concepts and not having to install sophisticated software.

1. Introduction

During the last years, an increasing number of tools, such as Soaplab [1], w2h[2], Pise [3] or Jemboss [4], have been developed to enable the biologists to run specific analysis software, either remote or local, within a homogeneous environment. In order to define a better integration of various services, some systems, such as myGrid [5] and biomoby [6] rely on ontologies to either describe the bioinformatics objets, or the analysis services. Software systems such as Pegasys [7], Taverna [8], w3h [9], G-Pipe [10], Panoramic [11] or Wildfire [12], not only provide a framework for defining and enacting workflows, such as biopipe [12], but also focus on the user interface aspects, generally by providing a graphical representation of the workflows. Our approach both aims at services integration and end-user access, that we address through the more limited but more widely-used Web interaction paradigm. In this poster, we first describe our user-centered studies and how they helped us to understand and address end-users issues. Next we briefly describe our framework and the early prototype of our system.

2. Methods

In order to better anticipate how biologists would access advanced mechanisms to enhance, combine, automate or configure their bioinformatics analyses, we have conducted a serie of in-situ interviews (about 30) and organised 3 video-prototyping workshops with researchers of the Pasteur Institute [14][15]. The outcome of these studies essentially warned us about the actual need of tools for building complex pipelines or accessing just-discovered web services. Furthermore, simple problems related to the data and programs' results seem to be more important for biologists not having much training in computing than sophisticated tools. Yet, we wanted to provide basic pipeline and discovery mechanisms, for we observed that they address a major need: biologists spend a lot of time copying and pasting data to combine analyses, or searching various well-know servers and assembling results in ad-hoc protocols, that they put down in their laboratory notebook.

3. Framework

The framework encompasses various components on both the client and the server sides (Fig. 1). On the client side, in addition to the Web portal, components include: a command-line interface as well as some APIs (python, perl). A biomoby access is also under construction, thanks to Y. Wong's python API [16]. On the server side, the current DTD for Mobyle services enables to either run local Unix programs, wrap remote Web servers, or call biomoby web services. A first implementation of the Web wrapper and the biomoby API is available on the RPBS Web server [16].

4. Prototype

Up until now, several components of the system have been developed. Fig. 2. shows the current state of Web user's interface prototype. The left panel is, as usual, devoted to navigation. It includes several types of navigation: looking for services, either by searching or browsing the services tree; a list of finished or currently running jobs (where finished jobs are highlighted in green) is also provided: from this list, the user can retrieve previous requests. The navigation panel also includes access to user's data files formerly analyzed: once selected, corresponding jobs and services' forms become available for re-use. Finally, tutorials are provided, since, as we were able to observe, they often provide a helpful basis for finding appropriate services. The top tabs provide access to forms according to their primary data type input: it thus also offer a way for filtering services. This area will probably give access to biomoby object ontology in a

near future, provided we can confirm a real need for this additional feature for the end-user. The data loading area includes an access to previously loaded data through the 'History' item. The center area is divided in two parts: one for data loading, and a part for services' forms, which share the data loading area corresponding to their main data input type.

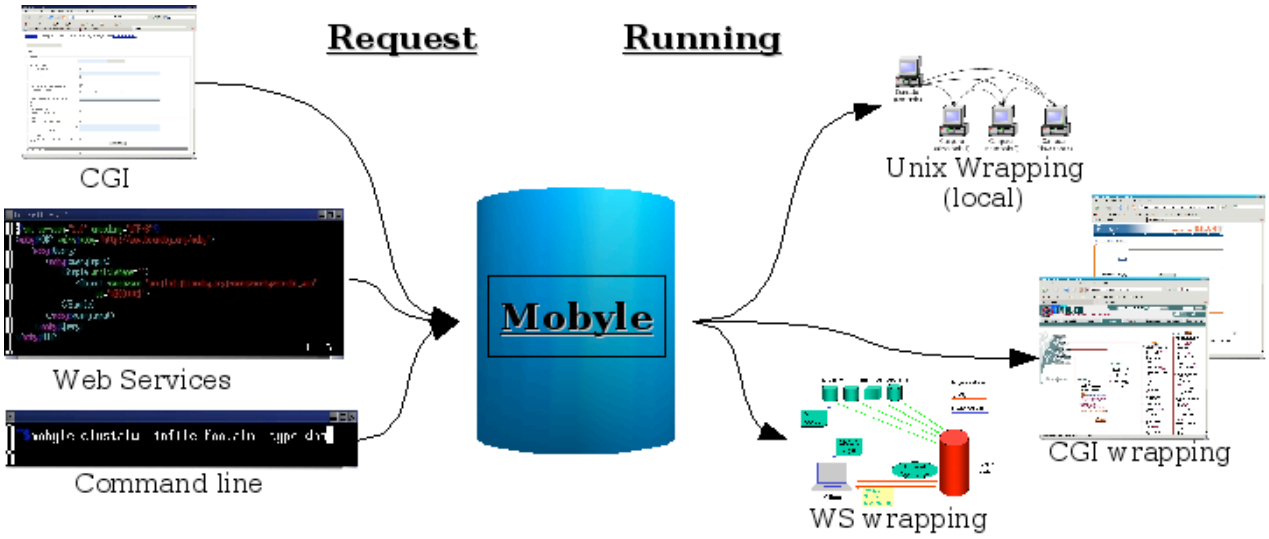

Fig. 1. Mobyle framework.

The web user's interface relies on several techniques, such as DHTML, template processing with the SimpleTAL python package, and Slither, a framework for developping interactive Web applications [17]. We also have adopted the Ajax approach [18], which enables very interactive Web interfaces development, by providing server-side dynamic document generation through javascript XmlHttpRequest objects.

5. Pipelines

Our design for providing pipelines deliberately seeks simplicity. We only address linear pipelines, and we opted for a familiar textual representation and simple interaction mechanisms. As shown in Fig. 2, services' and jobs' lists may refer to pre-defined pipelines, visually indicated by a list of services separated by '>>'. The form area displays a pipeline ready for use, which includes 3 steps: blast, clustalw and dnapars. The second step's form is currently displayed; other steps can be selected by clicking on their name in the tab (note that the data area defines input only for the *first* step). New pipelines may be defined in two ways (not shown in the figure): either by interactively chaining programs at run time, like in Pise [3], or by connecting programs before any submissions, like in G-Pipe [10] thanks to a “Pipe” menu. Finally, a “Repeat” button in the data loading area indicates to the server that the analysis should be performed for each data item (e.g: sequence) in the submitted dataset. Although not as general as a real loop statements, this feature corresponds to 90% of the loops found in current scripts.

6. Conclusion

The system still needs heavy coding, as well as some remaining design tasks regarding the Web interface. We intend to proceed to a first evaluation by organizing usability tests in the next monthes. The poster goes together with a demonstration of the current prototype.

7. References

1. M. Senger, P. Rice and T. Oinn, "Soaplab - a unified Sesame door to analysis tools", Proceedings, UK e-Science, All Hands Meeting 2003, Editors - Simon J Cox, p.509-513, ISBN - 1-904425-11-9, September 2003.
2. Senger,M., Flores,T., Glatting,K.-H., Ernst,P., Hotz-Wagenblatt,A. and Suhai,S. (1998) W2H: WWW interface to the GCG sequence analysis package. Bioinformatics, 14, 452-457.
3. C. Letondal (2001) A Web interface generator for molecular biology programs in Unix. Bioinformatics, 17(1): 73-82.

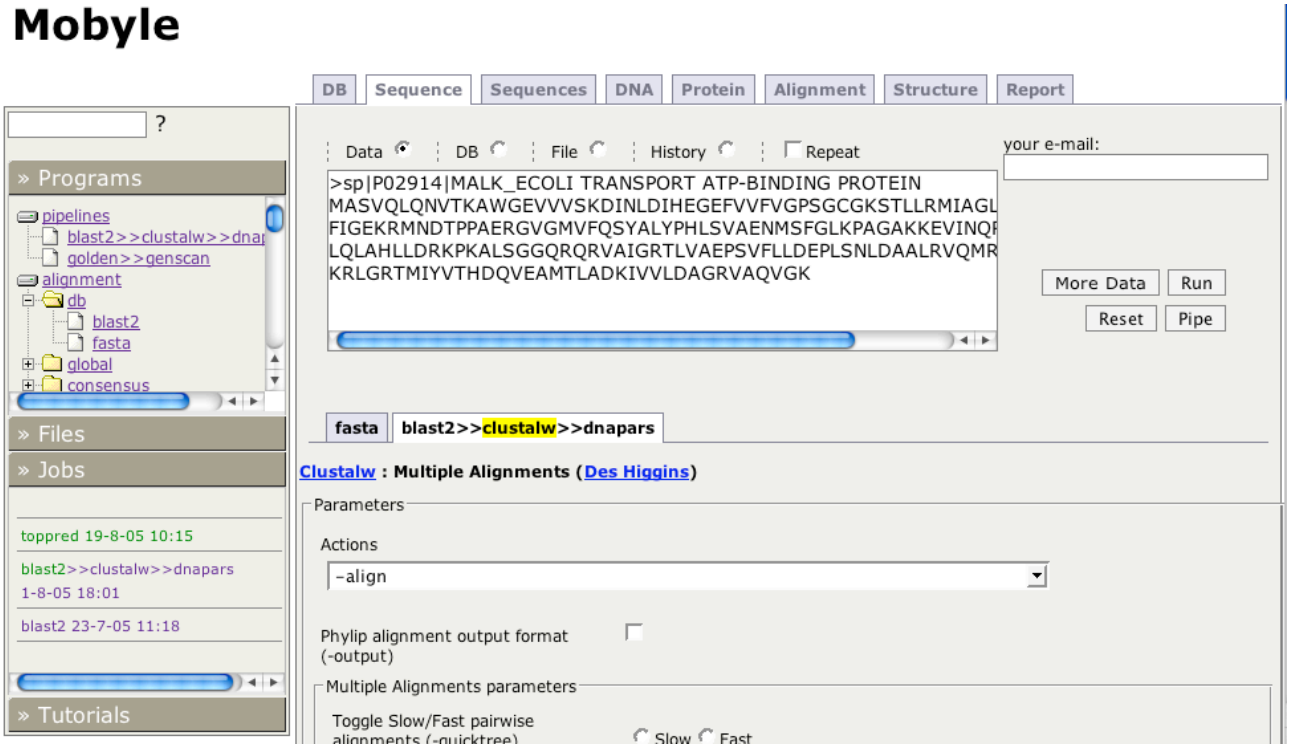

Fig. 2. Web interface prototype

4. Carver,T.J. and Mullan,L.J. (2002) A new graphical user interface to EMBOSS. Comp. Funct. Genomics, 3, 75-78
5. R. Stevens, A. Robinson and C. Goble, myGrid: personalised bioinformatics on the information grid, Bioinformatics, 19(1):i302-i304, 2003
6. Wilkinson MD (2004). BioMOBY - the MOBY-S Platform for Interoperable Data Service Provision in Computational Genomics Theory and Application. Richard P. Grant (ed) Horizon Bioscience, Wymondham, U.K
7. Sohrab P Shah, David YM He, Jessica N Sawkins, Jeffrey C Druce, Gerald Quon, Drew Lett, Grace XY Zheng, Tao Xu, BF Francis Ouellette. Pegasys: software for executing and integrating analyses of biological sequences. BMC Bioinformatics 2004, 5:40
8. Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock MR, Wipat A, and Li P (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20:3045-3054.
9. Ernst,P., Glatting,K-H and Shuai,S. (2003) A task framework for the web interface W2H. Bioinformatics, 19, 278-282.
10. Alexander Garcia Castro, Samuel Thoraval, Leyla J Garcia and Mark A Ragan (2005) Workflows in bioinformatics: meta-analysis and prototype implementation of a workflow generator. BMC Bioinformatics 2005, 6:87
11. T. Rose, F. Valentin. Panoramic http://www.pasteur.fr/recherche/panoramic/
12. F. Tang , C. Lian Chua , L-Y Ho , Y. Ping Lim , Praveen Issac and Arun Krishnan (2005) Wildfire: distributed, Grid-enabled workflow construction and execution, BMC Bioinformatics 2005, 6:69
13. S. Hoon, K. Kumar Ratnapu, J. Chia, B. Kumarasamy, X. Juguang, M. Clamp, A. Stabenau, S. Potter, L. Clarke, and E. Stupka, Biopipe: A Flexible Framework for Protocol-Based Bioinformatics Analysis, Genome Research, 13:1904-1915, 2003
14. C. Letondal, O. Amanatian (2004) Participatory Design of Pipeline tools and Web services in bioinformatics, Requirements Capture for Collaboration in e-Science Workshop. National e-Science Center, Edinburgh, January 2004.
15. C. Letondal and W. E. Mackay (2004), Participatory Programming and the Scope of Mutual Responsibility: Balancing scientific, design and software commitment, in Proceedings of PDC 2004 (Participatory Design Conference), July 27 -31, 2004 - Toronto, Canada.
16. C. Alland, et al (2005) RPBS: a web resource for structural bioinformatics Nucleic Acids Res., July 1, 2005; 33(suppl_2): W44 - W49.
17. G. K. Thiruvathukal, T. W. Christopher, J. P. Shafaee (2001) Web Programming in Python: Techniques for Integrating Linux, Apache and MySQL, Prentice Hall, 768 pages.
18. Garrett, J. "Ajax: A new approach to web applications." http://www.adaptivepath.com/publications/essays/archives/000385.php