BIOINFORMATICS REVIEW

Vol. 23 no. 14 2007, pages 1713–1717
doi:10.1093/bioinformatics/btm239

Sequence analysis

Bioinformatics software for biologists in the genomics era

Sudhir Kumar[1,*,†] and Joel Dudley[1,2,†]

[1]Center for Evolutionary Functional Genomics, The Biodesign Institute and School of Life Sciences, Arizona State University, Tempe, Arizona 85287-5301 and [2]Stanford Medical Informatics, Stanford University, Stanford, CA 94305-5479, USA

Received on March 16, 2007; revised and accepted on April 26, 2007

Advance Access publication May 7, 2007

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: The genome sequencing revolution is approaching a landmark figure of 1000 completely sequenced genomes. Coupled with fast-declining, per-base sequencing costs, this influx of DNA sequence data has encouraged laboratory scientists to engage large datasets in comparative sequence analyses for making evolutionary, functional and translational inferences. However, the majority of the scientists at the forefront of experimental research are not bioinformaticians, so a gap exists between the user-friendly software needed and the scripting/programming infrastructure often employed for the analysis of large numbers of genes, long genomic segments and groups of sequences. We see an urgent need for the expansion of the fundamental paradigms under which biologist-friendly software tools are designed and developed to fulfill the needs of biologists to analyze large datasets by using sophisticated computational methods. We argue that the design principles need to be sensitive to the reality that comparatively small teams of biologists have historically developed some of the most popular biological software packages in molecular evolutionary analysis. Furthermore, biological intuitiveness and investigator empowerment need to take precedence over the current supposition that biologists should re-tool and become programmers when analyzing genome scale datasets.

Contact: s.kumar@asu.edu

1 INTRODUCTION

The scope of comparative sequence analysis in molecular biology and genetics has expanded dramatically following rather humble beginnings with datasets containing a few homologs of a few proteins in the early 1960's (Fig. 1A) (Dayhoff *et al.*, 1965; Hagen, 2000; Higgs and Attwood, 2005; Kumar, 2005). Major advances in DNA sequencing technology in the last decade have allowed for the assembly of grand datasets that include hundreds of homologous sequences from a large number of species and genes (Brown, 1999; Ciccarelli *et al.*, 2006; Eisen, 1998; Gu *et al.*, 2002; Huson *et al.*, 2007; Koonin *et al.*, 1997; Sankoff and Nadeau, 2000).

Consequently, laboratory scientists are analyzing very large datasets on their desktops, which, until recently, used to be the exclusive domain of investigators skilled in bioinformatics tools and techniques. As biologists venture into bioinformatics, they often have to trade their favorite graphical computer desktop environments for the comparatively arcane command line interfaces, and they have to learn to write patch-work programming scripts that are used to 'glue' functionality from several distinct computational tools into a coherent analysis pipeline. The growing need to learn and employ programming/scripting skills is an impediment to effective research involving genome scale datasets for many (e.g., www.oreilly.com/news/perlbio_1001.html), because it presupposes an availability of time and an interest in programming.

We see an increasing need for the development of data analysis software that provides bioinformatics functionalities to biologists without requiring prior knowledge of programming and scripting languages. These software tools should fulfill biologists' needs to apply the most advanced and sophisticated computational methods without having to learn the often cumbersome command line versions of very useful programs. At the same time, these biologist-friendly tools need to be useable on different operating systems, and they need to provide a natural language description of the results produced in order to state assumptions made in the analysis.

Given our understanding of the software usage patterns of evolutionary biologists, we present the above-mentioned considerations in the realm of modern bioinformatics software available to biologists. Our primary focus is a typical biological researcher who is not a programmer, but a scientist actively generating and testing hypotheses at the desktop or lab bench. These biologists vastly outnumber those who are bioinformaticians, and it is they who are poised to draw unique insights into the emerging deluge of genomics data due to their position at the forefront of the experimental design and sequence data generation. Such investigators generally prefer user-friendly software tools with extensive graphical interfaces (Fig. 1B) (Roberts, 2004), which are frequently written by small research teams and often come with user-friendly interfaces. These software tools now need to evolve in response to the challenges that are presented by the need to analyze an exponentially growing amount of sequence data. A new paradigm is needed in which these existing and future teams are able to easily adapt to growing research needs while staying focused on their core

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, both the authors should be regarded as joint First Authors.

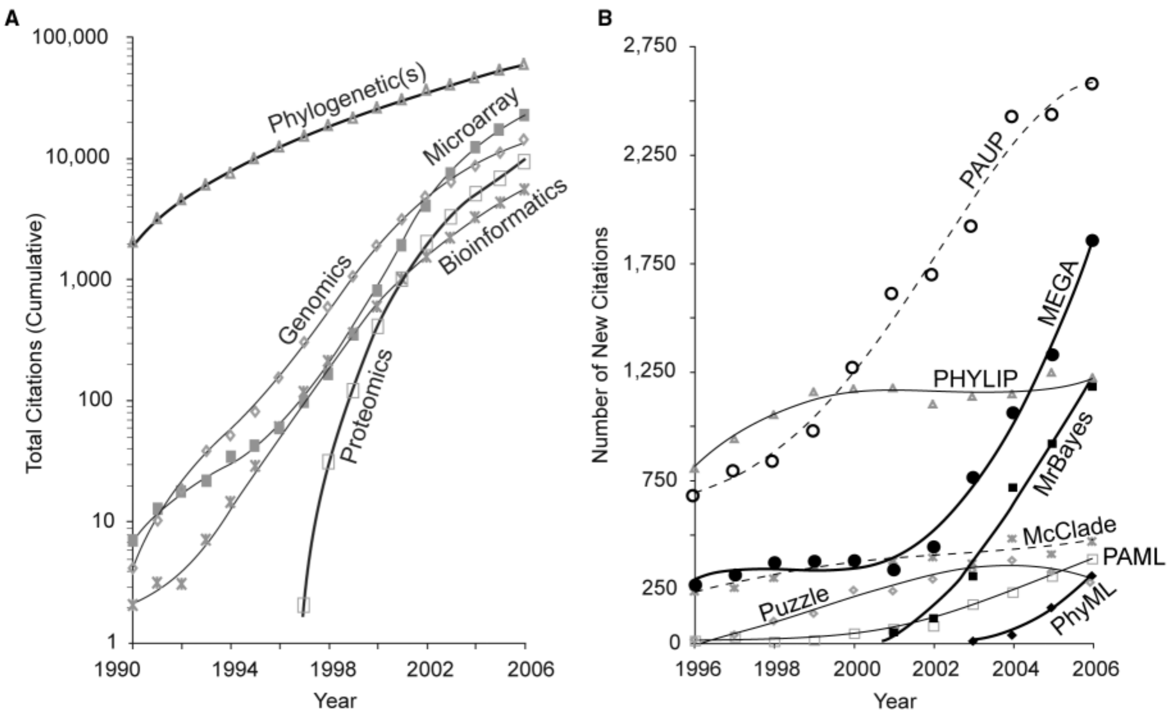

Fig. 1. (**A**) Expanding scope of phylogenetic analyses as reflected in the number of scientific articles that use terms, such as Phylogenetics and Genomics in their titles, abstracts or keywords. Modified and extended from Higgs and Attwood (2005). (**B**) Relative impacts of evolutionary analysis software packages over the last 10 years. Only non-commercial software packages available on-line (without fee) are included, except for two available for a nominal fee (shown with dashed line). Data for both panels were obtained from the Web of Science (February 2007 edition). For panel B, the numbers of new citation were generated using the 'Cited References' facility with the search arguments for author name, cited work and citation year kindly provided by Joe Felsenstein for MEGA (www.megasoftware.net), PAUP (paup.csit.fsu.edu), PHYLIP (evolution.genetics.washington.edu/phylip.html), MrBayes (mrbayes.csit.fsu.edu), Puzzle (www.tree-puzzle.de), PhyML (atgc.lirmm.fr/phyml) and PAML (abacus.gene.ucl.ac.uk/software/paml.html).

competencies. In the following, we discuss a set of themes for the new biologist-centric paradigm with a focus on the tools that facilitate comparative analysis of sequence data, which is our primary area of expertise.

2 APPLYING THE SAME ANALYSIS FOR DIFFERENT DATA SEGMENTS

A survey of the published literature in genome research and evolutionary genomics clearly underscores the need to develop user-friendly functionalities that are geared toward enabling biological researchers to apply the same analysis across a large number of sequences, genes or other biological units of data.

This is because both an understanding of genome- or taxonomy-wide patterns of change and the ability to test whether a given observation is part of the norm or an exception is done by such repeated analysis. Traditionally, this objective is reached with the application of high-level scripting languages. For example, consider the fact that biologists frequently compare the non-synonymous (amino acid altering, K_a) and synonymous (silent, K_s) sequence divergence between species for a given gene, which can be easily accomplished with the help of many different software packages (Kumar *et al.*, 2004; Yang, 1997). The analysis assumes a bioinformatics dimension when a biologist wants to compare K_a and K_s for thousands of genes, and to contrast the observed patterns across genes. Currently, a biologist has to either compute K_a and K_s for each gene manually in most desktop software, or to write batch files or procedural scripts. Both alternatives are awkward and inefficient.

Today, biologists need to employ scripting languages, because existing software packages do not provide facilities for iteratively carrying out such computations automatically. Attempts have been made to render these scripting languages friendlier for biologists through projects such as BioPERL (Stajich *et al.*, 2002). While these efforts have had a revolutionary impact on bioinformaticians' endeavors (Kell and Oliver, 2004), the use of these frameworks entails significant learning curves for non-programmers. The actual process of developing, debugging and maintaining scripts can be laborious and time-consuming, and even basic scripting languages require biologists to express their analysis in terms of variables and operators, which is excessively technologically involved. Hence, we see an emerging need for the development of graphical computing environments that are anchored in biological context. Such environments would enable biologists to visually define and execute large-scale, iterative analyses in terms of biological domain concepts (e.g. site-by-site and gene-by-gene analysis), and to present the results using biologist-centric results explorers, rather than large, cryptic ASCII text files.

3 HOSTING SOPHISTICATED TOOLS IN USER-FRIENDLY PLATFORMS

Biologists at the forefront of experimental design and research want to utilize sophisticated and powerful computational and statistical methods developed by statisticians and computational biologists (Baxevanis and Ouellette, 2005; Felsenstein, 2004; Higgs and Attwood, 2005; Li, 1997; Nei and Kumar, 2000; Nielsen, 2005). However, many theoretical developments are slow to reach mainstream biologists due to the specialized (command-line) interfaces of the software implementing new statistical and computational methods. This creates a gap between the developers of computational and statistical methods and the laboratory scientists, which is reflected in the citation impact statistics of easy-to-use programs as compared to a vast majority of other programs (Roberts, 2004). Citation impacts of many of these software packages far exceed those of more celebrated frameworks in bioinformatics, including BioMOBY (Wilkinson and Links, 2002) and PISE (Letondal, 2001).

Within the context of the user-friendly software, we favor a solution where the existing implementations of computational methods can be incorporated 'as is', without requiring any significant effort from the developer of the program that is being incorporated. We refer to this approach as 'Application Linking', which is similar to 'wrapping' (Spitznagel and Garlan, 2003). The aim of Application Linking is to allow existing user-friendly applications to seamlessly host third-party scripts and applications through its graphical interface, such that the user is abstracted from the intricate nuances of the hosted application's non-visual execution requirements (e.g. process control, system I/O and control files). In effect, biologists will be able to employ the user-friendly applications for data assembly, data handling and result visualization system, as specialist programs often frustrate biological users due to their lack of such amenities. This would ensure that the established user-friendly software function as platforms that integrate and unify the diversity of computational and statistical method implementations under familiar interfaces and analysis environments, e.g. (Che *et al*., 2005). As new tools are integrated into the standard platforms, they can be easily combined with other analytical tools to realize analyses that may not have been previously feasible for non-bioinformaticians.

The Application Linking approach is favored over the alternative plug-in approach, because existing tools can be integrated without changing their source code to conform to a plug-in API (Application Programming Interface). Secondly, Application Linking does not impose many restrictions concerning the programming language and other technologies that were used to develop the application being linked, because the linked application will execute just as the user would in a stand-alone fashion.

However, within the Application Linking system, the onus of providing built-in mechanisms for software extensibility will fall on the developers of the widely used software packages. While this would require some additional work, it is desirable because these developers will now be able to avoid having to program new computational algorithms to keep their software up-to-date and relevant. Furthermore, their efforts will provide recourse for those investigators who have developed tools for their own research purposes, but suddenly find their tools to be in high demand among a particular research community. This approach is also more amiable for developers of niche computational tools, since the hosting application does not constrain the linked application's development through the imposition of a linking API or data exchange format. However, it is expected that such standard interfaces would emerge for the mutual benefit of both the hosting and linked application.

4 USER-FRIENDLY SOFTWARE ACROSS MULTIPLE PLATFORMS

While Microsoft Windows is the most widely used desktop operating system today, MacOS has been an historical favorite of many biologists (and is gaining popularity), and Linux is fast becoming more widely used because of its open-source nature and stable graphical user interface. How to make widely used software on one platform available on the others? Of course, an obvious course of action is to port the source code using cross-platform software frameworks, such as Java, QT (www.trolltech.com/products/qt/) or wxWidgets (www.wxwidgets.org). However, given that most popular software tools are developed and maintained by small teams of biologists/developers, such porting efforts are unlikely to be feasible, because they will require years of development and debugging efforts. Such cross-platform solutions also require sacrificing capabilities owing to limitations intrinsic to cross-platform frameworks and programming resource scarcity.

In fact, the undertaking of porting efforts by academic software development teams would undoubtedly hinder the expansion of their functionalities. Another possibility is to consider dictating that all future bioinformatics software developments make use of cross-platform programming frameworks. This paradigm is not sensitive to the manner and circumstances under which novel bioinformatics tools are developed. Academic software tools are born out of a specific research need and are often implemented using the programming language (or technology) with which the investigator is the most comfortable or familiar.

At least for software tools written for the Microsoft Windows platform, it is possible to use commercial emulation software (e.g. VirtualPC) as a solution for those desiring to use Windows applications on Mac or Linux operating systems. Rather than the use of emulation software or the recoding of the source code, we favor the use of Application Compatibility Layers, which is a newly emerging alternative for making Windows applications useable on Apple and Linux machines. With Apple Computer's move to Intel-based hardware architectures, the x86- architecture has become standard among all three major computer workstation operating systems. This fact makes it feasible to develop and distribute Windows software installations with native software compatibility layers for all major operating systems.

The application compatibility layers, which are neither hardware nor software emulators, enable the native execution of Windows applications on Linux and Macs by providing an implementation of the Win32 API. Applications running on top of these layers can interact with the computer just as would any native Linux/MacOS application. Our tests of Molecular Evolutionary Genetics analysis (MEGA) running on Linux using one such system [Wine; www.winehq.org] have shown the display, stability and performance to be highly satisfactory and comparable to the native Windows system (Tamura *et al.*, 2007). In the case of programs that have been written to run exclusively on UNIX-based operating systems, the Cygwin (http://www.cygwin.com) compatibility layer, which provides Linux API compatibility, can be used to execute such programs natively on Windows-based workstations.

In our view, the essence of addressing the cross-platform issue is that bioinformatics software should continue to be developed using the platform and development technology that enables the developer to express the software design with maximum biological relevance, as any software can now be reasonably executed across all major platforms using existing software compatibility layers. In this approach, Application Linking can still be used to unify bioinformatics tools across major platforms. This is a far cry from the cross-platform utopia traditionally envisioned, but it is both pragmatic and tractable given the current state of bioinformatics software development.

5 RESEARCHER-ACCESSIBLE RESULT DESCRIPTIONS

As biologists increase their reliance on advanced software and complex computational methods to conduct their research, they need to be provided with all assumptions made and the values of all parameters used during analysis in their user-friendly software. These descriptions should be in biologically relevant, natural language text that is easily comprehensible. If possible, the software should also provide full citations for each identifiable method used during the analysis, along with information on each of the third party tools embedded. This design principle is important because it aids the investigator in the interpretation of the results. Availability of detailed descriptions of the results will prove extremely useful for new and expert users, as it will promote a better understanding of the underlying assumptions and finer details of the results presented. We have already implemented such a facility in our own software package MEGA. MEGA now contains a Caption Expert, which is a system that generates natural language descriptions of the result produced, along with full citations of the scientific literature (Tamura *et al.*, 2007). Even in its early test phase, we received very positive response from the biological community.

6 CONCLUSIONS

Given the need of laboratory scientists to conduct the analysis of large numbers of genes and sequences, we have advocated an enhanced attention for the adoption of a new paradigm in efforts aimed at providing user-friendly sequence analysis software. We have also advocated a pragmatic approach to the evolution of computational tools in which popular user-friendly software packages can become integrative platforms that serve as a bridge between developers of computational and statistical methods and laboratory scientists.

With these goals in mind, we need to consider issues about the support and the long-term fate of the widely used computer software. Because these tools are often built by small academic teams with limited resources, the expansion of their user base over time creates significant overhead for their authors, who are inundated with requests for documentation, technical support and persistent calls for further developments in response to community needs. Many authors are neither trained to develop documentation, nor equipped with the resources for providing technical support. At present, no dialog seems to exist about how to provide support to authors and users of popular software tools. Similarly, a serious dialog is needed regarding the fate of widely used academic programs when their primary authors cannot continue to support or update them due to time constraints or lack of extramural funding.

Rather than fall into disuse and go extinct, a National Consortium and associated infrastructure needs to be established for the preservation of high-impact academic software (and databases). This National Consortium may be given the charge of providing open-source-project hosting in the spirit of sourceforge.net in which the source code, documentation and other systems for orphaned software can be preserved for existing users. It could fall within the purview of this consortium to find development groups interested in taking the lead to further the development of highly successful orphaned software tools. Such a consortium might spearhead (or guide) the integration of functionality from popular tools into existing and established user-friendly software packages, and also might publish the best practices for biologist-centric software development.

For over five years, the National Institutes of Health has already emphasized the need for contemporary software to be interoperable, well-documented, open source and easily modifiable and extendable (e.g. http://grants.nih.gov/grants/guide/pa-files/PAR-05-057.html). In our opinion, it is now time to take the next step to build a National Consortium for high-impact software for the purposes of promoting the development of biologist-centric software tools, and for extending the lifespan of widely used research frameworks. Such a consortium would serve to establish a framework through which expertise and resources can be developed and exchanged for the betterment of the participating software development teams, their software projects and, ultimately, the biological community.

ACKNOWLEDGEMENTS

We thank Drs Alan Filipski, Sonja Prohaska, and Jonathan Wren for many critical comments on an initial version of the manuscript, and Dr Joe Felsenstein for providing the search arguments used to generate citation impact for various software packages in Figure 1B. We also thank Ms Kristi Garboushian for editorial support and Ashly Ruttman for help with generating figures. This work was supported in part by a research grant from National Institutes of Health (S.K.).

Conflict of Interest: none declared.

REFERENCES

Baxevanis,A.D. and Ouellette,B.F.F. (2005) *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. Wiley, Hoboken, NJ.

Brown,T.A. (1999) *Genomes*. Bios Scientific Publishers, Wiley-Liss, New York.

Che,D. *et al*. (2005) BEST: binding-site estimation suite of tools. *Bioinformatics*, **21**, 2909–2911.

Ciccarelli,F.D. *et al*. (2006) Toward automatic reconstruction of a highly resolved tree of life. *Science*, **311**, 1283–1287.

Dayhoff,M.O. (1965) *Atlas of Protein Sequence and Structure*. *National Biomedical Research Foundation*. Silver Spring, MD.

Eisen,J.A. (1998) Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Res*., **8**, 163–167.

Felsenstein,J. (2004) *Inferring Phylogenies*. Sinauer Associates, Sunderland, MA.

Gu,X. *et al*. (2002) Age distribution of human gene families shows significant roles of both large- and small-scale duplications in vertebrate evolution. *Nat*. *Genet*., **31**, 205–209.

Hagen,J.B. (2000) The origins of bioinformatics. *Nat*. *Rev*. *Genet*., **1**, 231–236.

Higgs,P.G. and Attwood,T.K. (2005) *Bioinformatics and Molecular Evolution*. Blackwell Publishing, Malden, MA.

Huson,D.H. *et al*. (2007) MEGAN analysis of metagenomic data. *Genome Res*.

Kell,D.B. and Oliver,S.G. (2004) Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era. *Bioessays*, **26**, 99–105.

Koonin,E.V. *et al*. (1997) Comparison of archaeal and bacterial genomes: computer analysis of protein sequences predicts novel functions and suggests a chimeric origin for the archaea. *Mol*. *Microbiol*., **25**, 619–637.

Kumar,S. (2005) Molecular clocks: four decades of evolution. *Nat*. *Rev*. *Genet*., **6**, 654–662.

Kumar,S. *et al*. (2004) MEGA3: integrated software for molecular evolutionary genetics analysis and sequence alignment. *Brief*. *Bioinformatics*, **5**, 150–163.

Letondal,C. (2001) A web interface generator for molecular biology programs in Unix. *Bioinformatics*, **17**, 73–82.

Li,W.-H. (1997) *Molecular Evolution*. Sinauer Associates, Sunderland, MA.

Nei,M. and Kumar,S. (2000) *Molecular Evolution and Phylogenetics*. Oxford University Press, New York.

Nielsen,R. (2005) *Statistical Methods in Molecular Evolution*. Springer, New York.

Roberts,J. (2004) New growth in phylogeny programs. *The Scientist*, **18**, 22.

Sankoff,D. and Nadeau,J.H. (2000) *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families*. Kluwer Academic, Dordrecht, Boston.

Spitznagel,B. and Garlan,D. (2003) A compositional formalization of connector wrappers. In *Proceedings of the 25th International Conference on Software Engineering*. Portland, Oregon.

Stajich,J.E. *et al*. (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res*., **12**, 1611–1618.

Tamura,K. *et al*. (2007) MEGA4: Molecular evolutionary genetics analysis (MEGA) software version 4.0.. *Mol*. *Biol*. *Evol*., DOI:10.1093/molbev/msm092.

Yang,Z. (1997) PAML: a program package for phylogenetic analysis by maximum likelihood. *Comput*. *Appl*. *Biosci*., **13**, 555–556.

Wilkinson,M.D. and Links,M. (2002) BioMOBY: an open source biological web services proposal. *Brief Bioinformatics*, **3**, 331–341.