Let the shoemaker make the shoes – An abstraction layer is needed between bioinformatic analysis, tools, data, and equipment: An agenda for the next 5 years

Tariq Segal*‡ and Ross Barnard†

* Biotechnology Program, University of Queensland,
† Department of Biochemistry and Molecular Biology, University of Queensland,
Brisbane, Queensland, 4072 Australia

‡ tariq@segal.com.au

Abstract

Bioinformatic tool development is being driven by individual efforts which while extending the boundaries of what is possible, are constrained by the framework in which the tools are being defined. This is resulting in a slower development process, as well as tools that operate independently of other tools, and suffer inconsistent interfaces in terminology, layout, level of standards compatibility, stability, etc. The utility of these tools could be increased with better planning and development during this growth stage of bioinformatics tool development but this requires the adoption of a 'grander plan' of how the architecture of bioinformatics should be laid out.

There are a number of themes to this discourse.

It is economically attractive to allow specialists to focus on their respective scientific specialities rather than on building a computing framework and the associated tools necessary to pursue their respective specialities.

There is a body of knowledge in how to best utilise individual tools or combinations of tools and this knowledge is not being captured and codified and consequently not being exploited to its fullest.

It is possible to build problem solving strategies around a tool or combinations of tools and codified knowledge about how to best utilise the tool/s.

It should be possible to mix, match, combine, and exclude individual tools as components in a toolset employed by user defined problem solving strategies.

Finally, it should be possible to automate the execution of user-defined strategies utilising tools and tool expertise.

Keywords: Bioinformatics, tools, abstraction layer, framework

1 Introduction

With the excess of news about the sequencing of the human genome and the proliferous and readily available search tools, sequence translation tools, homology modelling tools, and the amount of money being invested by traditional IT companies and technology venture capitalist, one begins to expect that biology is now simply a number crunching exercise and with enough computing power we will be able to run a couple of programs and understand how life functions. Unfortunately, we have not yet reached this panacea and the current generation of tools are not up to this task.

At this point in the development of bioinformatic infrastructure we see the researcher working hard to be both an expert in his or her field as well as managing a strategy to stay abreast of the tools available, develop or enhance these tools, as well as utilise emerging tools at a sufficiently expert level of operation.

And this is not a static domain. The number of search and modelling tools continues to grow every year. Manzetti (2002) emphasized that it is becoming increasingly difficult to determine the best tools to use and the order in which to use them. Choices may be determined by habit subsequent to serendipitous discovery, rather than by more objective selection criteria related to fitness for the task. There is no guarantee that we are using the best tools for the task and, if by chance we are, then there is no guarantee that we are using the tools in the most effective fashion.

Without the boundaries of defined standards or a dominant development framework individual researchers continue to create tools based on their individual preferences and concepts of utility. Contrast this with common computer application development. No-one would consider writing their own word processor and if they did would be stopped by analysing the product that could be delivered for the equivalent price of buying MS-Word.

Economists have been stating for decades that utility is best served by having resources work where they contribute the most return and this is a major pillar of the argument for comparative advantage. There is no benefit in making your own shoes unless you happen to be a shoemaker. In other words researchers should spend their time where their expertise is most valuable.

There is exponential growth occurring in available biological data and a widening gap between data and knowledge. Finding information in this overload of data requires volume processing and biologists need to better utilise computing. Unfortunately, most biologists are not experts in the disciplines of mathematics, statistics nor computing. The bio-information industry requires a framework within which the numerous bioinformatic tools can operate such that researchers can start using these tools as their 'tools of trade' rather than their 'fruits of labour'.

Copyright © 2003, Australian Computer Society, Inc. This paper appeared at First Asia-Pacific Bioinformatics Conference, Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol. 19. Yi-Ping Phoebe Chen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

There is now a need to create a framework able to be used as a bioinformatic backplane in which tools, equipment, and data connect, in much the same way as additional functions can be inserted into software applications and that will allow the automatic or manual execution of problem solving strategies utilising resources from across the entire resource pool.

2 An Example Problem

Bioinformatics combined with structural genomics shows the potential to generate structural information reliable enough to assist in a variety of tasks from function prediction to drug target identification. The task of generating structural information usually begins with sequence search tools such as BLAST, FASTA, WU-BLAST, that have gone through numerous evolutions, are well accepted and well understood and produce results that become inputs to family membership search tools such as CLUSTAL-X. These tools are complemented by template prediction tools that utilise initial sequence searches or a chemistry analysis leading to structure prediction and finally function prediction.

This structure modelling problem demonstrates the need to codify tool usage expertise. The major source of deceptive alignments in BLAST searches is the presence within proteins of regions that have highly biased amino acid compositions and these errors can be amplified in an iterative profile search using PSI-BLAST (Altschul & Koonin, 1998). An experienced researcher may or may not be aware of this limitation but new users of these tools would most probably be unaware. Altschul & Koonin (1998) note that while filters can be used to eliminate most biased regions, PSI-BLAST can still generate compositionally rooted artefacts, usually identifiable by 'inspection', especially when sequences such as myosins or collagens are retrieved, and hence favour filters such as SEG and COILS for sequence pre-processing before submission.

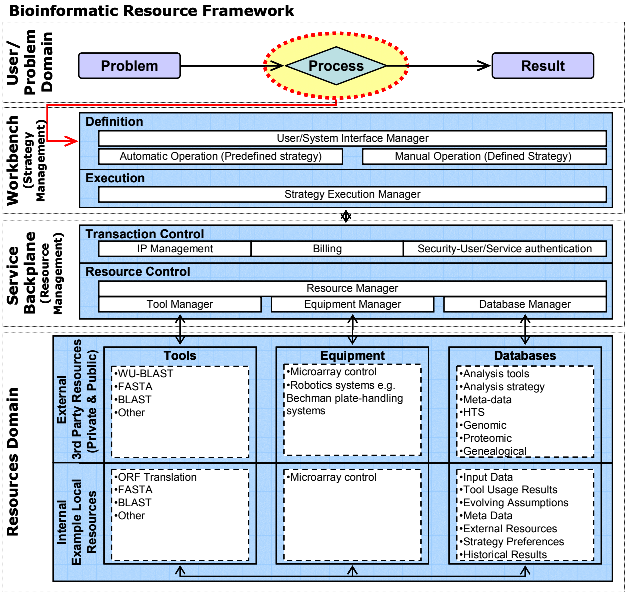
This issue highlights two issues. Unless tool usage is understood results can be worthless and, often the expertise already exists to overcome these usage caveats.

3 What would a framework look like?

There are many ways that such a framework could be constructed and this report develops one such model. It is improbable that the current system of disparate and independent tools will evolve into an integrated system as each individual tool is evolving to satisfy a different set of needs and consequently, determined interference in the evolution of this system is necessary.

The proposed framework introduces a workbench and a backplane. The workbench answers the “What do I want to do?” question while the backplane provides the “Go and do it” management process.

Diagram 1 depicts the proposed framework highlighting the main tasks required and identifying the information flows that would exist in such a system. The four layers of the framework are explained in the following subsections.



Source: Design by Tariq Segal 2002

Diagram 1: Distributed System Architecture

3.1 The Problem Domain

The top layer of the diagram shows the problem domain. Moving from problem to result requires that it be possible to define a problem solving strategy. Currently, strategy execution is performed manually by a researcher interacting with the resources shown in the bottom layer. While a strategy can be managed and executed manually, the framework proposed aims to capture and codify these strategies and then make them available for either manual or automated execution as shown in the workbench layer.

3.2 The Workbench

The first function of the workbench is the user interface. In addition, the workbench is the tool that captures the user’s expertise and utilises and exploits previously defined strategies and expertise. The workbench allows strategy definition, and the manual or automatic execution of this locally defined or externally retrieved strategy, to present a scored and ranked result in response to the user’s problem.

3.3 The Services Backplane

The services backplane is a resource management system designed to manage and share information between tools, equipment and databases, both internal and external, as part of strategy execution. The ability of tools to influence the input and thus processing of similar or complementary tools increases the value of the entire resource pool.

For example, a backplane may interface with a local BLAST tool, an external FASTA tool, a CLUSTALW tool and the NCBI database as part of a sequence family membership search. The strategy execution manager would dialog with the services backplane to manage the BLAST and FASTA searches, using the combined ranked results as inputs into the CLUSTAL tool.

The services backplane must also handle all administrative services, including the management of paid services, using secure resources, billing systems and payment gateways, and intellectual property management systems.

3.4 The Resource Domain

The final layer of diagram 1 shows the resource domain comprising of the resource classes and resources available. Available resources will vary by researcher as resources may be internal or external, private or public, subscription based or free. Individual researchers will utilise a subset of the entire resource pool based on personal preferences, domain specific needs, areas of interest, required equipment, and provisioned access.

The major resources classes are tools, databases, and equipment. Strategies may themselves be considered resources but are described separately in later sections.

3.4.1 Databases

Enormous amounts of biological data already exists spread across various databases, often incompatible with each other and often designed for a single type of analysis. Many bioinformatic tools are locked into specific data sources and abstraction is required at this interface also.

Data clarity is required much as normalisation is necessary to the relational data model. There is a need for a virtual database with a single albeit evolving database schema encompassing all data relevant to the bioinformatic domain. The data itself need not reside in a single location or database but must be managed by a single schema. This schema must be flexible enough to encompass the current data, the capability to expand in new directions and the flexibility to dynamically change the data structure itself.

This global database would be the repository of known validated data and is to be used in conjunction with local working databases holding additional local analysis data.

The objective is to leave as much data as possible external in a virtual global database while retrieving enough to allow for local processing. The local database should operate as part of the global database schema while being able to override it should the need arise.

3.4.2 Tools

One goal of this framework is to make it more possible to create tools that have access to data, equipment and other tools, and that are themselves accessible to the rest of the bioinformatic world. Tools must perform conceptually simple tasks such that they can be used as blocks in strategies. Tools must use common data exchange protocols and should not be tied in to local data sources.

While it is easy to find and operate many of the numerous existing bioinformatic tools, interpretation of the results is not such an easy task. These tools are still for the experienced biologist, even when following recommended pathways. At most, the novice user is restricted to using default options for almost all analysis. Additionally, results are only meaningful when the underlying algorithms and logic of the tools is understood and aligns with the logic and strategy of the researcher. Unfortunately, even for the experienced biologist lack of familiarity with these tools can lead to undesired or sub-optimal results.

In order to develop strategies, tools must be obvious in their execution such that it is conceptually simple for the researcher to understand the performance. We may not understand how a car engine works but can conceptualise a power source that drives a car forwards or backwards depending on whether the gear is in forward or reverse.

The framework requires both fast efficient tools that are conceptually easy to understand and that can form part of a toolset of functions, as well as a pool of expertise about tool usage that can be employed or ignored depending on a defined strategy. This could allow the knowledge to avoid local iteration traps as described previously while allowing the flexibility to use either, some, none, or all of the available search tools like FASTA and BLAST and their variants as part of a sequence search strategy.

While many tools now employ common standards, protocols, and databases, tools do not talk to each other. Consequently, each analysis step is an independent process brought together in the mind of the researcher. The utility of the entire system would increase if each tool could contribute to the operation of every other, and the ability to dynamically restructure tasks based on discoveries is likely to generate better or faster results.

3.4.3 Equipment

Lab equipment is being produced with network interfaces and remote management capability. Automated micro-array screening systems, high-throughput imaging systems, and software are becoming available. While it is impossible to trivialise the function of the lab, as ultimately *in-silico* predictions and models always need to be tested and validated, the benefits of volume production will drive the creation of HTS equipment. At some future time the lab will be virtualised and researcher presence may be optional. This framework should readily extend to include laboratory equipment in a strategy execution.

It should be possible to construct a strategy that allows a researcher to remotely access a microarray process, perform an analysis, image and analyse the results, and then adjust the inputs to a subsequent iteration of the microarray analysis based on the results of the first test.

4 How can strategies be captured?

In the simplest sense, a strategy may be considered a defined process to obtain a solution to a problem. Problem solving strategies, as sets of instructions, are a distinct body of expertise that can be captured and shared by being formalised in systems as is the presumption behind the design of computer systems (Von Neumann, 1993).

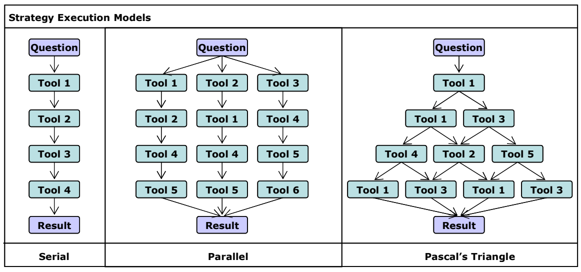


Diagram 2: Strategy Execution Models

Diagram 2 shows a number of process strategies. The left model indicates a simple sequential tool usage strategy. To some extent, this is the current process where a researcher employs a variety of tools in a sequential process. The middle model is a parallel model. In many instances researchers employ this strategy although it is executed serially. For example, a researcher may perform BLAST, FASTA, and PSI-BLAST searches, aggregate the results then best guess the input to a CLUSTAL-X search tool.

The rightmost model is a more complex strategy where multiple pathways are followed until execution is stopped or completed. Often ranking and scoring strategies are used at each level to promote or restrict courses of action. Strategies such as this are rarely used in the current environment and are likely to be a major improvement from the implantation of this framework. For example, a strategy may process both a BLAST and FASTA search and use some form of probabilistic multiplication to promote the commonly returned results to increase their attractiveness. This may then feed in parallel into a SCOP family matching tool and a superfamily matching tool and the results of this ranked to choose the next action.

Using the example problem, diagram 3 shows how a strategy can be defined as a flowchart where sequence searching is performed in parallel with multiple sequence alignment and sequence composition tools and the aggregated results are ranked and scored as inputs into the family analysis process. Family and superfamily analysis are then run in parallel and are subsequently ranked and scored. The most promising results at this point are run through a barrage of feature detection programs.

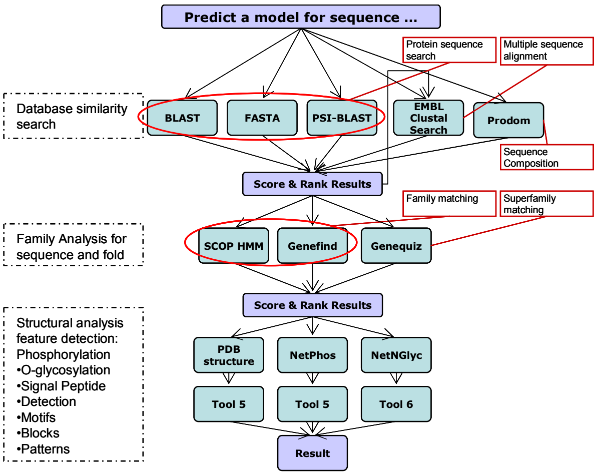


Diagram 3: Example Strategy Execution

Ranking and scoring will become areas of expertise and may use knowledge from other domains. The minimax strategy (Michie, 1997) famously employed by IBM's "Big Blue" for chess may provide insights into strategy pathway selection based on interim scoring. In this strategy, potential forward courses of action are extrapolated out a defined number of steps, whereupon the potential outcomes are scored and ranked and used in the selection of the subsequent move.

Researchers have developed valuable problem solving strategies but it is not easy to compare, rate, rank and share this expertise. Researchers could be defined by the quality of the tools they have built, the quality of the strategies they develop, the quality of the results they retrieve and the quality of the conclusions they develop.

This framework can evolve further to include self-tuning systems using recursive brute force computing and/or researcher result evaluations. Strategies can be run against historical data sets to determine optimal predictive strategies by comparing predictions against known data. This process underlies the annual CAFASP (Moult et al., 2001) and CASP (Moult et al., 1995) competitions, competitions designed to further protein prediction by comparing prediction strategies against known structures.

With such a framework, many of the emerging articles being written detailing search strategies could be codified and compared across large search sets and historical data.

5 Conclusion

The problem of an increasing population of bioinformatics tools and the lack of an integrated and systematized interface for their selection and utilization is becoming widely acknowledged. This has led some workers (Manzetti, 2002) to suggest specific flowcharts to enable users to navigate from raw sequence to their desired end point of functional, structural and relational information.

However the informatics web has reached a level of complexity that a more general approach is needed. The problem has shifted from tool development to architecture development and the optimal path forward is for tools to evolve from stand alone solutions to components of a more complex environment. New integrative and recursive interfaces will be necessary to address the issues of optimal use and accessibility of engines and synthesis of their outputs and this development will itself be a major project in logistics and collaboration.

6 Acknowledgements

The authors wish to thank Dr. John Fuerst and Mr. Rakesh David, both of University of Queensland for valuable discussions and insights.

7 References

Altschul, S.F. and Koonin, E.F., (1998), Iterated profile searches with PSI-BLAST- a tool for discovery in protein databases, *TIBS* 23, 444-447.

Buttler, D., and Critchlaw, T., (2001), Using Meta-Data to Automatically Wrap Bioinformatic Sources, available at http://www.cc.gatech.edu/~buttler/DAML/OOPSLA_01.pdf

Manzetti, S. (2002) Taking the complexity out of protein sequence analysis, *Drug Discovery Today*. 7:172-175.

Michie, D., (1997), Slaughter on Seventh Avenue, *New Scientist*, no. 2085 June 7th, pp 26-29.

Moult, J., Fidelis, K., Zemla, A., Hubbard, T., (2001), Critical assessment of methods of protein structure prediction (CASP): Round IV, *PROTEINS: Structure, Function, and Genetics*, 2-7, Suppl. 5.

Moult, J., Pedersen, J.T., Judson, R., Fidelis, K., (1995), A large-scale Experiment to Assess Protein-Structure Prediction Methods, *PROTEINS: Structure, Function, and Genetics*, 23 (3): R2-R4 NOV.

Von Neumann, J., (1993), First Draft Report on the EDVAC, *IEEE Annals of the History of Computing*, vol. 15, no. 4, 1993, pp. 27-75.