A Bioinformatics Reference Model: Towards a Framework for Developing and Organising Bioinformatic Resources

Hong Liang Hiew* and Matthew Bellgard*

* *The Western Australian Centre for Comparative Genomics, Murdoch University, South St, Murdoch WA 6150, Australia*

Abstract. Life Science research faces the constant challenge of how to effectively handle an ever-growing body of bioinformatics software and online resources. The users and developers of bioinformatics resources have a diverse set of competing demands on how these resources need to be developed and organised. Unfortunately, there does not exist an adequate community-wide framework to integrate such competing demands. The problems that arise from this include unstructured standards development, the emergence of tools that do not meet specific needs of researchers, and often times a communications gap between those who use the tools and those who supply them. This paper presents an overview of the different functions and needs of bioinformatics stakeholders to determine what may be required in a community-wide framework. A *Bioinformatics Reference Model* is proposed as a basis for such a framework. The reference model outlines the functional relationship between research usage and technical aspects of bioinformatics resources. It separates important functions into multiple structured layers, clarifies how they relate to each other, and highlights the gaps that need to be addressed for progress towards a diverse, manageable, and sustainable body of resources. The relevance of this reference model to the bioscience research community, and its implications in progress for organising our bioinformatics resources, are discussed.

Keywords: bioinformatics, reference model, resources

INTRODUCTION

The growth of bioinformatics tools, software and online resources has been at an unprecedented pace, and the importance of such resources in life science research is now beyond dispute [1-4]. The Bioinformatics Links Directory (http://bioinformatics.ca/links_directory/) featured in the Nucleic Acids Research journal's annual bellwether *Web Servers* special issue now lists over 1000 web servers hosted in over 35 countries with web-based bioinformatics tools [5]. The equally important Molecular Biology Database Collection from the same journal (http://217.169.56.209/nar/database/c/) as reported in its 2007 update [6] now lists 968 different databases. The growth of such lists is mirrored by the rate of increase in available resources at major international bioinformatics sites such as NCBI (http://www.ncbi.nlm.nih.gov/), EMBL-EBI (http://www.ebi.ac.uk/) and ExPASy (http://expasy.org/). Further demands from the expected data deluge in the near future from major sequencing projects utilising new high-throughput technologies, from the

CP952, *Computational Models for Life Sciences—CMLS '07, 2007 International Symposium*
edited by T. D. Pham and X. Zhou

demands for ongoing *in-silico* analyses, and from the needs for rapid production of outcomes from research. The rate of growth of bioinformatics resources will likely accelerate in the coming years.

The bioinformatics field now faces the important issue of how to organise its resources to efficiently and effectively support life science research. There is a growing body of evidence that better approaches than those currently employed are needed [7-11]. The main difficulty in making progress on this front is that the development of bioinformatics resources by its nature is faced with competing considerations such as usability versus feature richness, innovation versus interoperability, and research relevance versus technical excellence. All such considerations have their merits, and one should not be arbitrarily discarded over another without due attention. Therefore, a natural solution for this would be to develop a community-wide coherent framework to integrate these competing considerations into a productive environment. A coherent framework would be able to assist in alleviating many common problems in bioinformatics, such as i) the complexity of bioinformatics resources that keeps growing faster than available mechanisms to properly organise them, ii) standards that are developed and adopted in unstructured ways, iii) tools that are built for one area of research are not easily ported and used in research of a related area, and iv) communication gaps where research needs are not translated to relevant tool features.

The problems of competing considerations outlined above, and the evolution towards a common framework, are natural paths for most technologies as they grow to reach critical mass. All technologies experience these phenomena. Although not necessary explicitly articulated, all mature technologies have such frameworks in place. For example in the automotive industry, there is an implicit understanding regarding the components of an automobile, how they functionally inter-relate, which parties are responsible for developing which components, how the components will integrate to form an automobile, and how all components will be maintained post production. Note that different stakeholders in the automotive industry serve different functions, have different issues to manage, and use different approaches in their tasks. Not all parties agree on everything, and not all components following exactly the same standards. However, there is an implicit framework linking the functions of all components: from the raw minerals, to processed metals, to car parts to, to manufactured cars, to service stations, to car accessories, to transports logistics, and so forth. Ultimately, all these components operate in a synergistic and integrated way to service the needs of all stakeholders in the automotive industry. This is the automotive industry's implicit "framework". It is conjectured in this paper that the bioinformatics field have arrived at a point in its evolution where stakeholders should be participating in actively developing a similar shared framework. Although there have been some efforts in building frameworks in bioinformatics for specific functions such as workflows [12], data semantics [10], service descriptions [13] and tool generation and customisation [14], a coherent community-wide framework covering all bioinformatics functions has yet to emerge.

This paper presents an analysis of the bioinformatics field to determine what is needed for a community-wide framework for organising bioinformatics functions and components. A *Bioinformatics Reference Model* is proposed as the basis for such a

framework. A reference model [15, 16] is a template model that separates different functions and components into independently manageable components and layers. The reference model outlines the relationship between research usage and technical aspects of resources. It separates important functions into multiple structured layers, clarifies how they relate to each other, and highlights the gaps that need to be addressed. To achieve the aim of an efficient and effective body of resources, these gaps will need to be addressed.

The next sections will discuss the concept of a *resource*, and then analyse the current problems with diverse considerations in resource handling. The proposed reference model solution is presented following from this analysis. The paper concludes with a discussion on the issues related to the model, and the relevance of the model to developing bioinformatics support for life science research outcomes.

A REFERENCE MODEL FOR BIOINFORMATICS

Building a resource framework

In the area of bioinformatics, the term *resource* can refer to a variety of entities. Some common examples of resources include software programs, online tools, databases, datasets, documents, web servers, workstations, protocols and standards. To develop and organise such a vast body of resources is a complex task. The solution for such an undertaking may be found in the area of software engineering, where the primary concern is the development and organization of a vast complex body of software resources. One of the key developments in software engineering to tackle the complexity problem is the focus on *architecture* for large projects [16-18]. In particular, the area of computer networking have benefited significantly from the use of architectural ideas. Computer network development follows the concept of *reference models*. A networking reference model [15, 16, 19, 20] is a template specifying layers of components. Each layer is responsible for certain functions and issues, and it interacts with layers above and below them to produce a working networking environment. Similar to the automobile industry example described earlier, the benefits of such a model to the stakeholders is considerable. For example, network users today have a choice of which internet service provider to connect to, without needing to become involved in network programming, service exchanges, devices, and technicians all the other intricacies they are not suppose to be responsible for. This situation can provide useful lessons for bioinformatics. Equivalent structured frameworks can help the community organise and develop the informatics resources used for life-science research.

Principles of an effective model

To derive a reference model, we adapt the basic principles of effective architectural layering to some of the basic needs within the bioinformatics field. These adapted principles are given in Table 1.

TABLE 1. Principles of Effective Layering. This set of principles are synthesized from the ISO standard for Open Systems Interconnection Model [20], and adapted in this paper for suitable use in bioinformatics.

Principle		Description
1	Appropriate separation of functions	Collect similar functions into one layer. Separate manifestly different functions into different layers. Organise layers into linear hierarchy so that functions in one layer only interact with functions in layers above and below them. But be wary of creating too many layers that the task of describing and integrating the layers are more difficult than necessary.
2	Appropriate interfaces between layers	Interfaces between layers should be demonstrably successful from past experiences, or will have reasonable chance of success in the future; Create interfaces as small as possible so that interactions across layers are minimized.
3	Planning for future	Localise functions in the layers so that the layer could be totally redesigned to take advantage of new advances in science and technology without changing the adjacent layers
4	Sub-layering	Create sub-layers where further grouping of functions based on principles 1-3 are necessary, but the groupings may be by-passed to allow functions to interact beyond their adjacent layers.

The principles in Table 1 are the foundations to effectively managing a complex environment of diverse components with different functions. Components at each layer can concentrate on their functions, while interfacing with other components in the environment in a clearly structured way.

Bioinformatics stakeholders

The next step to developing an effective reference model is to determine how resource functions are to be separated. An analysis of stakeholders' needs can be invaluable in this regard. In bioinformatics, there are many different categories of stakeholders. Some examples are *Life Science researchers*, *tool developers*, *technical support staff*, and *bioinformatics service providers*. Below is a brief analysis of each of those categories.

The *researchers* principally view bioinformatics resources as parts of their scientific methods. Their priorities are research and analysis, and the resources are only means to an end. Their demands are therefore related to the functionality of the resources and how it fits into research steps. Researchers also demand quality in resources such as interface ease-of-use, reliability and accessibility, that will enable them to effectively and efficiently use the resources in scientific work.

The *tool developers* view bioinformatics resources as something they produce. They therefore have to know what the resources are composed of, what can be used to create the resources, and what researchers (users) want to do with the resources. Like the researchers, they are also concerned with functionality, ease-of-use, reliability and accessibility. But unlike researchers, they have demands for properties not related to usage, such as portability, testability and evolvability, since it impacts on their design, production and distribution work.

Technical support staff deals with maintaining the applications in a form usable by the researchers. In that sense, they are not primarily interested in the functionality of the resources, but their maintainability, robustness and security.

Bioinformatics service providers need to manage and link resources with user needs. They are primarily interested in ensuring there is a supply of resources from tool developers, a market of demand from researcher users, and a team of staff who maintains the resources. They therefore partially share the demands of all of the stakeholders above. They also have additional interest in issues of ownership, authorship and price.

There are other categories of bioinformatics stakeholders than those described above. These include data curators, administrators, laboratory staff, vendors, educators, government agencies, policy makers, and so forth. However, for the purpose of analysis in this paper, an exhaustive review of all possible stakeholder categories is not necessary at this stage. There are already two very clearly recognisable views of bioinformatics resources from the description above. The first view, the *Scientific View*, is where research processes are of primary concern and bioinfomatics resources are secondary. The resources are of relevance only in support of research processes. The second view, the *Technological View*, is where resources are the primary objects of concern, and have issues that need to be dealt with in and of themselves. Fig. 1 shows some core components of these two views, and the relationship between the two views.

As depicted in Fig. 1, there is a gap between the scientific and technological views of bioinformatics resources. The technologies in the technological view are not in a form that can be naturally linked to the processes in the scientific view. The gap is in the form of incompatible functions, operations, requirements and components. If this gap is left unaddressed, resources will either be built to be too technically-focus and lacking in scientific relevance, or they will be built with scientific functionalities but without the necessary technical qualities to operate sustainably and effectively. The gap needs to be addressed, as illustrated in Fig. 2, by:

1. Integrating technologies into research relevant components, and
2. Delivering the components in an appropriate form for usage in research activities.

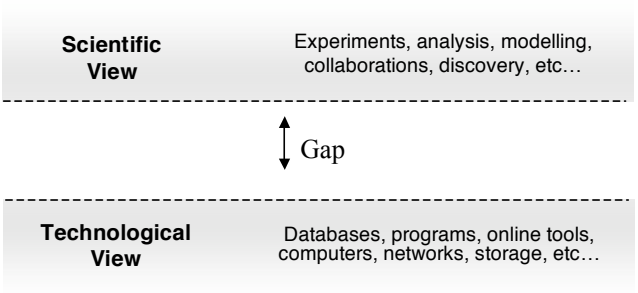

FIGURE 1. Scientific and Technological Views of Bioinformatics Resources

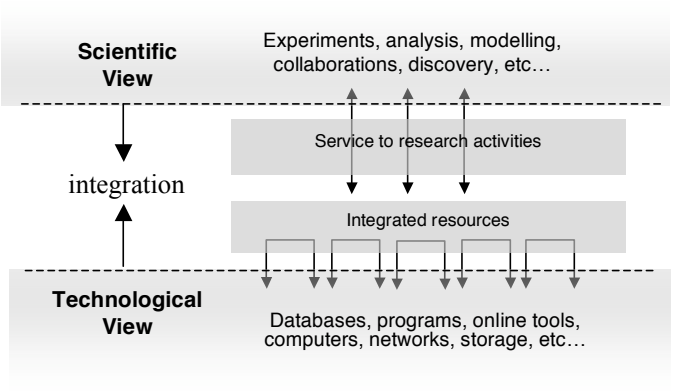

FIGURE 2. Integrating Bioinformatics Resources

A Reference Model

The previous sections lead logically to the Bioinformatics Reference Model shown in Fig. 3. It consists of 3 layers: *Research*, *Bioinformatics* and *Infrastructure*. The *Bioinformatics* layer is further divided into 3 sub-layers: *Service*, *Integration* and *Data & Tools*.

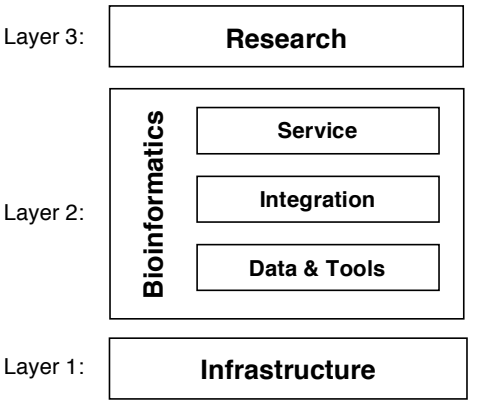

FIGURE 3. Bioinformatics Reference Model

The following sections describe each layer.

Layer 3: Research

The functions adopted within this layer are the life science research functions. These functions are provided through *research processes*. Bioinformatics resources need to contribute to these processes. Some example research processes that most commonly involve bioinformatics include data analysis, annotation, curation, modelling and prediction.

Layer 2: Bioinformatics

The *Bioinformatics* layer contains all functions that need to support research functions in layer 3. This layer consists of the three sub-layers describe below.

Sub-layer: Service

The sub-layer is defined by functions that can *directly* provide support to research. The functions transform resources into *research-service-ready* forms. For a resource to be research-service-ready, it should at least have the following *mandatory* properties:

1. The resource has an identified *name*.
2. The resource *performs an identified set of tasks*. The tasks *produces research-relevant outcomes*.
3. The resource has an *identified location and provider* (ie. the resource's users know how to get the resource).
4. The resource has an easily obtainable *description,* with information on items 1-3 above.
5. The resource has a *usable interface* to interact with the user.

The resource should also have the following *desirable* properties:

6. The resource has known *levels of performance*. It at least has specified levels of how reliable it is and how long it takes to conduct a standard task.
7. If the resource undergoes change, it has a clear description of *difference between past and present versions, and difference in levels of performance.*

To be service-ready therefore requires tools and systems take on the above properties before being exposed to the researchers operating in layer 3. There may be other extra properties certain researchers or service providers may prefer to have, but the list of 7 above should be the bare minimum.

Sub-layer: Integration

In this sub-layer, the main function is to link and connect components to form integrated resources. The integrated resource is made up of many parts, taken from the *Data & Tool* sub-layer below it. The integrated resource combines each of these

operational parts to produce something more relevant to the research activities in layer 3.

Examples of integrated resources with this sub-layer functions include genome browsers and pipelines such as Ensembl (http://www.ensembl.org/), cross-searching tools such as Entrez (http://www.ncbi.nlm.nih.gov/Entrez/), unified databases such as UniProt (http://www.uniprot.org/), online interfaces such as web sites, portals, workflows, and many others. Other widely used examples include the organism and database specific resources such as the The Arabidopsis Information Resource TAIR (http://www.arabidopsis.org/), the Comprehensive Microbial Resource (http://cmr.tigr.org/tigr-scripts/CMR/CmrHomePage.cgi) and the Online Mendelian Inheritance in Man (http://www.ncbi.nlm.nih.gov/sites/entrez?db=OMIM).

The common ways to integrate stand-alone data and tools into useful systems for research include the following (note that not all of them are mutually exclusive – most systems uses a combination of some or all):

1. Interface integration: allow the user to interact with various tools using a until a single, coherent interface. Eg. web sites.
2. Visualisation integration: present results from multiple sources in a single visual form. Eg. genome browsers.
3. Content integration: Data from multiple sources are put together. Eg. integrated datasets.
4. Input-output integration: Results from one resource are automatically fed into other resources for further processing. Eg. pipelines and workflows.
5. Processing integration: where the processing of different tools is combined for efficiency purposes.
6. Platform integration: where resources are put into a single unified platform to ease administration and support.
7. Distribution integration: where different resources are packaged, promoted and distributed in a single package.

With the growth of data and the needs for bioinformatics resources to assist in more complex tasks, it is expected that many other ways of integration will emerge.

Sub-layer: Data & Tools

In this sub-layer, the main functions are functions for specific bioinformatics tasks. Common examples of tool functions include molecular sequence searches, pair-wise and multiple sequence alignments, feature predictions, and phylogenetic analysis. Example of data functions includes storage of nucleotide and protein sequences, expressed sequence tags (ESTs), maps, mutations, enzymes, plasmids, organism-specific data, and literature references. The components that provide these functions are generally standalone tools and datasets. Components in Data & Tools are the basic building blocks of bioinformatics.

Tools can come in many forms. Currently the most common are programs directly installed and executed under a particular computing platform. Data sources are also stored in many different database platforms and formats. They generally require an interface tool to access or manipulate the data within them.

There is an important distinction between functions at this sub-layer, and the *Integration* sub-layer above this. Functions in this layer cannot be further decomposed into pieces that still produce results of life-science relevance. They are the lowest level in decomposing bioinformatics functions. It is proposed here that the *Integration* layer is kept separate from the *Data & Tool* layer for the reason that the bioinformatics field needs enough resources growing concurrently at both levels. If the emphasis is too heavily on providing standalone tools, then there will not be proper integrated systems that enable ever-more complex research activities. At the same time, if the emphasis is too much towards building integrated resources, it will mean at some stage components to use as parts for our integrations will be exhausted. There needs to be simultaneous development and evolution at both these levels to form a synergistic environment to feed off each other. A similar reason exists in keeping layer 2 separate from the layer 1 *Infrastructure* discussed in the following section.

Layer 1: Infrastructure

Functions in this layer do not contribute directly to life science research, but indirectly through higher layers. They are the building blocks used to build the components in the layer 2. Such infrastructure building blocks come in many forms and from many fields. They include functions like:

- Programming support, such as those provided by software libraries, web services, and development toolboxes;
- Theoretical foundations, such as algorithms, computational frameworks and theories;
- Analysis and statistical techniques from Mathematics and Statistics;
- Processing and storage through computers, operating systems, networks and storage devices.
- High-performance computing such as those from computational clusters, data federations, and grid resources.

DISCUSSION

The structuring of functions and components in the Bioinformatics Reference Model is based on the effective layering principles as outlined in Table 1. The principles cover the necessary grouping and linking of functions, the appropriate use of sub-layers, and the proper planning for future changes. By keeping to these principles, the model is able to provide a framework for how to effectively develop and organise a complex set of resources. Components relate and interoperate in a defined way. Components are then free to be developed in any form or adopt any properties as long as they maintain the functional relationships. The roles of stakeholders to develop, organise and maintain the components are therefore also defined as a by-product.

The decision to have three layers in the reference model emerged naturally from the analysis of the field. Research components should never have to interface with Infrastructure components, and vice-versa. Bioinformatics components can act as

appropriate intermediaries. The decision to further structure the three sub-layers in layer 2 also provide a clean way to organise bioinformatics resources: start with components providing basic tool and data functions, then linked them together to form integrated resources, and finally deliver through service components in service-ready form for use in research processes. However, this clean three sub-layers are not always practicable in the current environment (eg. integrated resources may need to access high-performance computing infrastructure resources directly rather than depending on tools to do it themselves). Therefore, the bioinformatics functions are kept as layer 2 sub-layers rather than as full layers.

The model construction also takes into account likely future changes and improvements to scientific processes, bioinformatics development, and available infrastructure. The functions in each layer can be re-designed without heavy impact to components in adjacent layers. However, the basic functional structure of the three layers is unlikely to change in the near future.

The major gaps that currently exist are mainly in the upper sub-layers of the Bioinformatics layer 2. There is an abundance of *Data & Tool* components, but a lot less effective integrated resources in the *Integration* sub-layer that takes basic data and tool components to cater directly to specific research processes. There is even less work in developing service functions as outlined in the *Service* sub-layer. These gaps will need to be addressed.

One of the natural downstream implications of the reference model is the development of a proper resource ontology, one that outlines what we need to define for our resources. Catalogues of bioinformatics resources can then be created. Ontology research is an integral part of the life sciences today with the emergence of key major projects such as the Gene Ontology [21] (http://www.geneontology.org/) and the Systems Biology Ontology (http://www.ebi.ac.uk/sbo/). However, these ontologies are biological ontologies used mainly as controlled vocabularies for data annotations, and are not *resource* ontologies. Developments in bioinformatics resource ontologies have begun to emerge in recent years in projects such as myGrid [13] and the defunct TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) Ontology [22, 23]. Developments such as these, integrated with more generic online resource frameworks such as the World-Wide-Web Consortium's (W3C) projects into resource description and resource discovery standards (http://www.w3.org/RDF/), are important progress towards proper bioinformatics resource ontologies. However, before dealing with ontology design and implementation issues, the role and function of ontologies to the whole bioinformatics environment has to be clarified. The development of frameworks such the reference model presented in this paper is an essential step in that evolution.

CONCLUSION

This paper deals with the problems of bioinformatics resource development and organisation. Without proper coordination, it is difficult to properly balance the competing needs of different stakeholders in bioinformatics resources. This paper presents a reference model for such a purpose. The reference model outlines the functional relationship between research usage and technical aspects of resources. It

separates important functions and components of into multiple structured layers, and it allows a synergistic and sustainable growth in the body of bioinformatics resources. The model has wide-ranging benefits to increasing productivity in scientific research. For example, there is a clear definition of what functions a researcher, bioinformatics service provider, and software developer are responsible for. This reduces duplication and allows the different stakeholders to concentrate on their primary functions. Another example of benefit is when developing a software system, it is clear how to architect its different functions (eg. service interface, data processing, integration, computation and storage functions, etc). All these benefits emerge from a clear framework outlining how to organize and develop bioinformatics resources.

There are many issues yet to be resolved in the community-wide development and organization of bioinformatics resources. The Bioinformatics Reference Model may serve as a starting point for open discussions as we progress towards a diverse, manageable, and sustainable body of resources.

REFERENCES

1. T. Reichhardt, *Nature* **399** (6736), 517-20 (1999)
2. F.S. Collins, E.D. Green, A.E. Guttmacher, M.S. Guyer and the US National Human Genome Research Institute, *Nature* **422** (6934), 835-47 (2003)
3. J.D. Wren, *IEEE Eng Med Biol Mag* **23**, 87–93 (2004)
4. B. Di Ventura, C. Lemerle, K. Michalodimitrakis and L. Serrano, *Nature* **443** (7111), 527-3 (2006)
5. J.A. Fox, S. McMillan and B.F. Ouellette, *Nucleic Acids Res*. **35** (Web Server issue), W3-W5 (2007)
6. M.Y. Galperin, Nucleic Acids Res. **35** (Database issue), D3-D4, (2007)
7. L. Stein, *Nature* **417**, 119-120, (2002)
8. L. Grivell, *EMBO Rep*. **3**, 200–203 (2002)
9. A.J. Cuticchia and W. S. Gregg, *News@Nature* **429**, 241 (2004)
10. E. Neumann, *Sci STKE* **283**, pe22 (2005)
11. N. Cannata, E. Merelli and R.B. Altman, *PLoS Comp Biol* **1** (7), e76 (2005)
12. A. Konagaya, *Proceedings of the NETTAB2006* (Santa Margherita di Pula, Sardinia, 2006), pp75-82.
13. C. Wroe, R. Stevens, C. Goble, A. Roberts and M. Greenwood, *Int. J. Coop. Inf. Syst.***12** (2), 197-224 (2003)
14. M.A. Swertz and R.C. Jansen, *Nat Rev Genet*. **8** (3):235-43 (2007)
15. H. Zimmermann, *IEEE Trans. Comm*. **28** (4), 425-432 (1980)
16. L. Bass, P. Clements and R. Kazman, "Architectural Patterns, Reference Models, and Reference Architectures" in *Software Architecture in Practice*, edited by L. Bass, P. Clements and R. Kazman, Addison-Wesley (2003)
17. M. Shaw and D. Garlan, *Software architecture: perspectives on an emerging discipline*, Prentice Hall (1996)
18. L. Bass, P. Clements and R. Kazman, *Software Architecture in Practice*, Addison-Wesley (2003)
19. A.S. Tanenbaum, *Computer Networks (1st edition),* Prentice Hall (1981)
20. ISO 7498 Open System Interconnection Model, International Organization for Standardization, Geneva, Switzerland. (1994), http://acm.org/sigcomm/standards/iso_stds/OSI_MODEL/ accessed 15 August 2007
21. The Gene Ontology Consortium, *Nature Genetics* **25**, 25-29 (2000)
22. S. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N.W. Paton, C.A. Goble and A. Brass, *Bioinformatics* **16** (2), 184-185 (2000)
23. P.G. Baker, C.A. Goble, S. Bechhofer, N.W. Paton and R. Stevens, A. Brass, *Bioinformatics* **15** (6), 510-520 (1999)