Databases and ontologies

BIAS: Bioinformatics Integrated Application Software

McGill Centre for Bioinformatics, 3775 University Street, McGill University, Montreal, Canada H3A 2B4

Received on December 19, 2004; revised on June 21, 2004; accepted on November 18, 2004
Advance Access publication November 30, 2004

ABSTRACT

Motivation: We introduce a development platform especially tailored to Bioinformatics research and software development. BIAS (Bioinformatics Integrated Application Software) provides the tools necessary for carrying out integrative Bioinformatics research requiring multiple datasets and analysis tools. It follows an object-relational strategy for providing persistent objects, allows third-party tools to be easily incorporated within the system and supports standards and data-exchange protocols common to Bioinformatics.

Availability: BIAS is an OpenSource project and is freely available to all interested users at http://www.mcb.mcgill.ca/~bias/. This website also contains a paper containing a more detailed description of BIAS and a sample implementation of a Bayesian network approach for the simultaneous prediction of gene regulation events and of mRNA expression from combinations of gene regulation events.

Contact: hallett@mcb.mcgill.ca

There now exists a wide range of ontologies, standards, databases and applications available to the Bioinformatics community. However, the vast majority of these objects exist independently and are not tightly integrated into a single system. This situation makes it difficult for research groups to quickly and effectively use state-of-the-art Bioinformatics tools. Furthermore, it necessitates a great effort on their part to both integrate all the relevant objects together into a usable platform and to maintain this system over time. The slow progress towards a development framework for Bioinformatics is due to a complex set of reasons including the heterogeneity and diversity of biological data (Lacroix, 2002). Additionally, the development of fields such as systems biology bring with it a need to integrate multiple types of information within a coherent statistical model (McMillan, 2002; Haas et al., 2003; Segal et al., 2002). However, there remains a general lack of tools available to help the community to manage these complex analyses. Existing development platforms such as [Acero genomics knowledge platform (2003), http://www.acero.com and Rosetta resolver (2004), http://www.rosettabio.com/products/resolver/] are generally not offered at a price accessible to academic laboratories.

We make available a new development platform entitled BIAS (Bioinformatics Integrated Application Software). The mandate of BIAS is to provide infrastructure to integrate many different Bioinformatics objects together into one easy to use, easy to expand and computationally powerful system. Although the system currently has infrastructure for gene expression, transcription factor binding and a sample implementation of a Bayesian network approach for the simultaneous prediction of gene regulation events and of mRNA expression from combinations of gene regulation events, BIAS is the seed of a coherent statistical model (McMillan, 2002; Haas et al., 2003; Segal et al., 2002). However, there remains a general lack of tools available to help the community to manage these complex analyses. Existing development platforms such as [Acero genomics knowledge platform (2003), http://www.acero.com and Rosetta resolver (2004), http://www.rosettabio.com/products/resolver/] are generally not offered at a price accessible to academic laboratories.

We make available a new development platform entitled BIAS (Bioinformatics Integrated Application Software). The mandate of BIAS is to provide infrastructure to integrate many different Bioinformatics objects together into one easy to use, easy to expand and computationally powerful system. Although the system currently has infrastructure for gene expression, transcription factor binding site and protein–protein interaction studies, BIAS is the seed of a coherent statistical model.

*To whom correspondence should be addressed.

Bioinformatics development framework. We have made solid, pragmatic design decisions and the OpenSource aspect of our system should encourage research groups to tailor this kernel to their specific needs rather than ‘re-creating the wheel’. We provide an intuition of the main components of BIAS. Figure 1 depicts an overview of BIAS.

Data sources: internal. The notion of an object-relational (OR) model is essentially an amalgamation of relational and object-oriented models [The apache DB project (2004), http://db.apache.org; Java 2 platform, enterprise edition (2003), http://java.sun.com/j2ee; Stonebraker et al. (1999)] that tries to provide the advantages of both systems. In particular, it allows programmers to work with a modern object-oriented programming environment while preserving the rich toolkit of reliable functionality offered for persistent storage by relational database management systems. BIAS offers a pragmatic OR model that maps objects in the object-oriented world to relations in the database and vice versa. When a programmer defines a new class in Java, the class is reflected as a table in the database. When an instance of the class is created, the instance is reflected as a tuple (row) in the corresponding table. The system guarantees that modifications to classes or instances of classes are automatically reflected in the database (and vice versa). The strengths of our system are its ease of use in comparison with other OR systems (http://db.apache.org and http://java.sun.com/j2ee), and the ease with which many important Bioinformatics objects can be integrated into the model.

© The Author 2004. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oupjournals.org

Fig. 1. Basic overview of BIAS.
**Data sources: external.** Any Bioinformatics software must consider questions regarding the importation and exportation of data from foreign sources. There are three ways with which this is achieved: (1) parsing external files, (2) distributed access queries and (3) via a Java API. (1) BIAS provides a factory of programs for data import/export and data quality control. These routines provide a skeleton which other research groups can tailor to their specific needs. (2) BIAS also provides tools for using distributed annotation systems (Stein, 2003). The purpose of distributed annotation systems is to facilitate the de-centralization of the curation process of large multiple data resources. When a user queries a DAS server, the server retrieves the relevant information from multiple servers. (3) BIAS provides routines for exploiting Java APIs offered by third party databases. Currently, it is possible to connect to Ensembl directly through a Java API. In all cases, the imported data are automatically placed into BIAS Java objects.

**Libraries.** One of the greatest strengths of BIAS is its ability to incorporate third party software as external libraries. This allows the developer to use the BIAS data model and routines but at the same time make use of high-quality, specialized software packages. We have integrated the statistical package R (and Bioconductor) due to its use in the analysis of gene expression data (R Development Core Team, 2004). BIAS provides a library of basic functions, data structures, graphical user interface libraries and algorithms that are commonly used in many Bioinformatics analyses. In fact, the functionality offered by the BioJava package (http://www.biojava.org) is incorporated within the system.

**Modules, clients and servers.** A module is an application written by a developer in BIAS. It can make use of the BIAS internal and external libraries, data source access routines and data model. Essentially, modules are the basic means by which developers add Bioinformatics functionality to the framework. Modules can be used as a stand-alone package where the end-user of the system (i.e. the biologist) need not necessarily know that the application is written in BIAS. When an end-user uses BIAS, they execute the client process on a machine of their choice. The client is responsible for all communication to and from the server. The server component acts as a middleware layer between the modules/clients and the data sources/libraries components of the system and receives requests from the clients. It is primarily responsible for accessing the BIAS libraries and data resources of a module. At present, BIAS remains a single-user system but development of a multi-user client/server architecture is underway.

**Discussion and future directions.** BIAS currently contains infrastructure for: (1) gene expression studies (cDNA, Agilent and Affymetrix) compliant with the MIAME standard; (2) handling complete genomes and annotations; (3) transcription factor binding site analyses; (4) genetic network inference; and (5) protein–protein interaction experiments. The most pressing needs to the underlying system are better graphic user interface classes, and more advanced support for distribution annotation systems. We plan to integrate BioPerl into BIAS and finish the client/server implementation in the immediate future. Recently, we have started to migrate our OR strategy to the HIBERNATE system (http://www.hibernate.org/).

**REFERENCES**

Acero genomics knowledge platform (2003).


The apache DB project (2004).