BenchDW: a generic framework for biological data warehouse benchmarking

Thomas Triplet
Centre for Structural and Functional Genomics
Department of Computer Science and Software Engineering, Concordia University
1455 De Maisonneuve Blvd. West
Montreal, Quebec, H3G 1M8, Canada
thomastriplet@gmail.com

Gregory Butler
Centre for Structural and Functional Genomics
Department of Computer Science and Software Engineering, Concordia University
1455 De Maisonneuve Blvd. West
Montreal, Quebec, H3G 1M8, Canada
gregb@encs.concordia.ca

ABSTRACT

The rapid development of *-omics* techniques have provided an unprecedented amount of data, enabling system-wide biological research. However, the success of systems biology is contingent on the ability to integrate a wide variety of types of biological data to automatically predict, assign functional annotations of proteins and perform comparative analyses. Although each biological data integration system presents to some extent a number of desirable features, none of them meets all the requirements for effective integration of system-wide data. In this paper, we present BenchDW, a generic and flexible benchmark framework that aims at facilitating the evaluation and quantification of the capabilities of those biological data warehouses. It currently comprises 22 different metrics ranging from documentation quality to accuracy and response times, which may be recorded for different hardware configurations. Each metric can be weighted to better suit the user's specific needs and compared to the gold standard. BenchDW was designed to be flexible, easy to use and offers many benefits over spreadsheets, thus presenting the characteristics required to facilitate acceptance by the scientific community. We demonstrate the utility of BenchDW by briefly reviewing three data warehouses (BioMart, BioXRT and InterMine) and by showcasing how it can be leveraged to identify the specificities of the systems of interest. BenchDW is available online at *http://warehousebenchmark.fungalgenomics.ca/benchmark/benchdw/index.html* under the GNU GPLv3 license.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

benchmark, genomics, systems biology, data warehousing, data integration

SAC'13 March 18-22, 2013, Coimbra, Portugal.

1. BACKGROUND

Life sciences techniques made significant improvements over the past decades, resulting in huge amounts of data collected over the years by the scientific community. In order to facilitate the organization and the subsequent analyses of this valuable data, databases have been developed very early. Since then, the number of databases has dramatically increased. The 2012 Molecular Biology Database Collection [7] includes nearly 1400 databases, each describing millions of biological records.

This unprecedented wealth of information originating from genomic studies represents a tremendous potential in all areas of biological science. Successful data integration is one of the keys to successful bioinformatics research [1]: scientists need an integrated view of these heterogeneous data sources with advanced data-mining, analysis and visualisation tools. Successful data integration also relies on computational techniques to automatically predict and assign functional annotations of proteins as effective integration of biological data should enable scientists to perform comparative analyses, modelling and inference of protein functions.

However, biological data present numerous challenges from the lack of standard formats to data inconsistencies resulting from experimental data variations [18]. Although each data warehousing system presents a number of desirable features, none of them meets all the requirements for effective integration of system-wide data: for example, BioMart [23] is a data federation framework that facilitates the simultaneous querying of multiple data sources but lacks sophisticated mining tools. BioXRT [22] offers a flexible database structure, despite its basic user interface. On the other hand, InterMine [13] features a customizable user interface and is helpful to track the provenance of data, but requires considerable efforts to configure. PathwayTools [12] achieves good accuracy, but the underlying data structure is not flexible as it is precomputed while building the warehouse.

1.1 Spreadsheet/Database Paradigm

In order to evaluate more accurately existing biological data warehousing systems, we developed a benchmark that comprises twenty-two important quantitative and qualitative metrics or dimensions for each warehouse, such as the accuracy of the output the warehouses could produce as an answer to a number of biologically meaningful queries, the running-time to obtain the answer or the quality of the documentation [19] (Table 1). We initially used an Excel-like

Table 1: Benchmark metrics. Weights can be dynamically adjusted by end-users according to their needs.

Metric	Category	Weight	Description
Accuracy	Output quality	7	Fraction of true correct results
F1-Score	Output quality	7	Harmonic mean of the precision and sensitivity
F2-Score	Output quality	4	Weighted harmonic means of the precision and the sensitivity (emphasis on sensitivity)
F0.5-Score	Output quality	4	Weighted harmonic means of the precision and the sensitivity (emphasis on precision)
Matthews Correlation Coefficient	Output quality	0	Balanced measure of the quality of a classification, which can be used on unbalanced data sets
Precision	Output quality	4	Fraction of retrieved results that are relevant to a particular query
Sensitivity	Output quality	4	Ability to identify positive results
Specificity	Output quality	0	Ability to identify negative results
Answer-time improvement over manual	Performance	2	Time to design and run the queries for a warehouse, compared to a manual approach using existing resources
Answer-time improvement over gold std.	Performance	2	Time to design and run the queries for a warehouse, compared to the normalized database
Query-design complexity	Performance	8	Time to translate the queries from natural language into a suitable form for the warehouse
Running-time impact	Performance	3	Time for the warehouse to produce the output
Number of APIs	Development	1	Variety of computer languages that may be used to programmatically interact with the data warehouses
Build time	Development	1	Time to build the system once it is properly configured
Configuration time	Development	5	Time to configure the warehouse
Customization level	Development	1	Flexibility of the warehouse
Number of dependencies	Development	1	Number of software on which the warehouses rely on to function
Installation time	Development	2	Time to install the warehouse framework and its dependencies
Open-source	Development	1	Source code is freely available
Clarity	Documentation	3	Difficulty to understand the documentation
Comprehensiveness	Documentation	5	Description of all the parameters
Support	Support	9	Evaluation of the support provided by developers

spreadsheet to record our measurements. Spreadsheets are indeed broadly used by the scientific community. Their intuitive and easily understandable user interface is a significant advantage. They are also visually appealing and feature a number of tools to visualize data using charts.

1.1.1 Scalability Issue

However, we found that spreadsheets did not scale up well as more dimensions were introduced into the benchmark. Spreadsheets might be sufficient when one needs to organize simple data. As reported in previous studies [11, 14, 5, 8], spreadsheets do not scale up well and, as the spreadsheet will expand to accommodate a growing number of records of increasing complexity, data handling will become increasingly cumbersome, hence reducing the utility of potentially valuable information. For example, it is not possible in Excel to generate dynamic pivot tables to aggregate data without using more complex Visual Basic macros.

1.1.2 Quality Control Issue

Besides the scalability issue, spreadsheets are subject to data redundancy and consequently data integrity loss. For example, if a hardware configuration or a query needs to be displayed in different spreadsheets, it will most likely be duplicated in each document. When the entry is updated in one place, all occurrences elsewhere may not be properly updated, which will result in multiple inconsistent versions of the same data. Moreover, unlike databases, spreadsheets do not enforce referential integrity: they do not check that resources referenced somewhere in the spreadsheet are still valid, which may be critical, in particular when those resources are frequently updated, as it is the case here.

In this paper, we present BenchDW, a generic and flexible alternative as a general benchmarking framework. BenchDW aims at facilitating and standardizing the evaluation and quantification of the capabilities of those biological data warehouses. Our framework currently comprises 22 different metrics, ranging from documentation quality to query response precision and response times, which may be recorded for different hardware configurations. Each metric can be weighted to better suit the user's specific needs and compared to the gold standard. They are also used for the computation of the overall grade that is assigned to each data warehouse. Our framework also logs all data entries so that the history of each record may be checked for future reference. BenchDW has been successfully used for the past two years to benchmark four biological data integration frameworks using 14 typical biological queries on 4 different hardware configurations.

2. IMPLEMENTATION

Despite their numerous benefits over spreadsheets, database management systems still lack satisfactory user interfaces for data analysis [3] whereas Excel spreadsheets do provide intuitive and well-known interfaces for data analysis and consolidation, provided the issues mentioned above are addressed.

Web-based applications are dynamic and interactive websites that offer a rich user interface comparable to standard desktop programs [9, 10]. They can be executed on any connected workstation, without software installation or specific requirements and have the major advantage of being always up-to-date wherever they are being accessed, thereby eluding the need for multiples copies of the same document on the various workstations used for the benchmark, effectively solving synchronization issues between local copies.

BenchDW was designed to present the five acceptance characteristics – relative advantage, compatibility, complexity, trialability and observability – as defined by Rogers [16] to maximize its utility. It consists of a modular and integrated collection of online spreadsheets accessible over the Internet and backed-up by a relational database for efficient data management. Spreadsheets can be used to record all aspects of the benchmark, from the list of data warehouses to benchmark and their software dependencies to output quality measurements. Furthermore, BenchDW is open-source. The main benefit for end-users is that they can easily extend our framework to record new metrics that better suit their specific needs. With minimal programming skills, it is possible for example to add a simple spreadsheet to accommodate new types of data in a few minutes.

We implemented the online spreadsheets as a set of highly dynamic web pages implemented using Asynchronous JavaScript and XML (AJAX) web technologies [10], which enable a web application to communicate with a server in the background using JavaScript and *XMLHttpRequest* objects, without interfering with the current state of the page. AJAX technologies provide an effective means to create dynamic web pages that can interact with the user. To make BenchDW more accessible, our framework features a *fluid* layout that automatically fits the content to the screen definition of the user. We successfully tested BenchDW on various screen definitions up to $3840 * 1080$ (dual Full-HD configuration). Our fluid approach will become increasingly beneficial for users as the sizes of monitors have significantly increased in the past few years and large definitions ($> 1024 * 728$) now account for over 85% [20].

The implementation of BenchDW also relies on a number of open-source programming libraries. The web user interface (see section 3.1 below for an overview) of the framework was implemented using ExtJS v4.0 [17], the general open-source AJAX framework from Sencha. It is backed-up by the freely available PostgreSQL 9.1 relational database management system [15]. The server-side code was implemented using PHP framework CodeIgniter 2.0 [6] enhanced with the HMVC [21] extension. The Model-View-Controller (MVC) architecture of CodeIgniter and HMVC enables our framework to adopt a very flexible and modular design. It is therefore possible to easily extend BenchDW with new features by implementing new modules.

To facilitate the configuration of BenchDW, we configured a virtual machine image for VirtualBox that comes with everything that is needed, so that BenchDW may be ready to use in a few clicks.

3. RESULTS

3.1 User Interface Overview

Figure 1 gives an overview of the graphical user interface (GUI). Most pages are composed of two panels: the main menu is on the left and a spreadsheet or a graph as the main panel, which is the primary means to enter to enter data. Several spreadsheets and/or graphs can be opened at the same time. The main menu is dynamically generated and can be easily extended to accommodate new modules that may be create by the community. The menu can be collapsed in one click so that the spreadsheets can be displayed in full screen mode to maximize the usable working space thanks to the fluid-layout of our system. The look-and-feel and color schema of BenchDW may be also customized to match the color conventions in use in each laboratory. Spreadsheets may also be customized by displaying, hiding, reordering and resizing columns as needed so that only the most relevant data are displayed. Where appropriate, small histograms are displayed within cells to give a more visual representation of the data and quickly compare different records.

3.2 Data Entry

3.2.1 Overview

The spreadsheet is the primary means of entering data in BenchDW. Each cell is associated with an editor whose format depends on the data within the cell. Most cell editors are simple text fields. More advanced editors are provided where needed. In particular, cross-references to other tables are typically associated with a combo box, whose content is dynamically generated after the content of the referenced table. Combo-boxes facilitate data entry by suggesting entries as the user types. They also have the added benefit of limiting data entry mistakes, in particular when users enter data that do not exist in the referenced table. Specific editors are also available for Boolean flags and dates. BenchDW also supports rich text editors with advanced text formatting capabilities, which are mainly used for comments and free-text cells.

Data may also be imported programmatically, using JavaScript and RESTful requests. Furthermore, users have the possibility to export BenchDW data to various formats in one click – in particular as Excel spreadsheets, and JSON and XML formats – for sharing and further analysis.

3.2.2 Data integrity and validation

To further reduce entry errors, each cell editor can be associated with a *validator*. Validators check the correctness of data types and send immediate feedback to the user in case of an error. They are usually based on regular expressions or more advanced customized functions as needed.

In addition, to minimize data entry, cells are automatically computed whenever possible. For example, the sensitivity, specificity, precision and accuracy of responses, as well as F-scores are automatically calculated based on the number of true/false positives/negatives. Calculated fields in BenchDW are also used to facilitate data entry. For instance, the hardware configuration used to measure performance metrics should appear on several related spreadsheets. Using standard spreadsheets, the user needs to copy and paste the name of the configuration wherever needed.

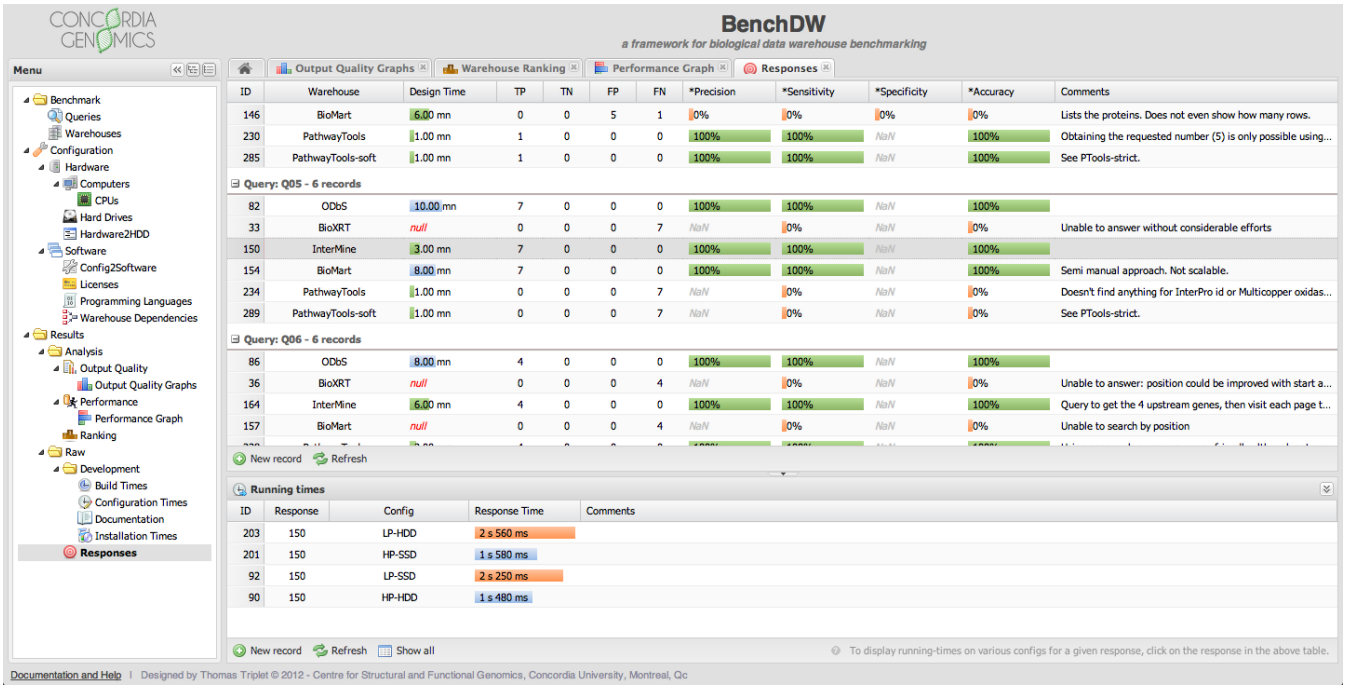

Figure 1: Overview of the graphical web user interface.

In BenchDW, the underlying relational database is leveraged to display the short description of the configuration in all tables where it is needed. The main benefit is that changes to the description of the configuration are automatically reflected in all tables and data in the various online tables are therefore always consistent and up-to-date.

3.3 Versioning and backups

Our system is supported by a PostgreSQL relational database, which efficiently handles versioning and backups. Unlike in standard spreadsheets, when a user updates or deletes a record in BenchDW, modifications are always logged for future reference as part of the record's history, which may be simply accessed by right-clicking on that record and select *Show modification history*. As a consequence, while updating a spreadsheet is always possible, no data are ever deleted and restoring a record to a previous state or accessing the complete data modification log in case an error is made while updating a spreadsheet is always possible. It should be noted that the log mechanisms are automatically triggered at the database level using the PL/TCL procedural language, and not at the application level. This ensures that log mechanisms are always triggered whenever data are modified, even when data is entered without using the regular GUI (programmatic RESTful requests or command-line for example) so that the log is always in consistent state.

3.4 Visualization tools and Data Analysis

3.4.1 Data-Mining

Each table in BenchDW is fully searchable and each column is associated with a flexible filter that depends on the type of data the column represents. Five different types of filters can be configured: textual, multi-selection, numerical, calendar and Boolean. Numerical filters let the user query for values above, below or equal to a given threshold. They are most useful to query the various metrics associated with each data warehouse. Boolean filters are typically used to retrieve records when given a flag. For instance, this filter is convenient to list all hardware configurations that have been flagged as virtual machines. Calendar filters are helpful to search for entries given a time frame. The multi-selection filter is most effective for searching for one or more items in a given list. The list may be a static enumeration or may be dynamically generated by the server based on data from other tables in case of foreign-key references.

3.4.2 Charts and Graphs

Most data in BenchDW can be viewed using tables. Where appropriate, tables were enhanced with *sparklines*, that is, small histograms within cells, so that it provides the user with a more visual overview of the data (see Figure 1). BenchDW also features a number of more advanced graphs to facilitate the visualization of more complex aggregated data (see Figures 2 and 3). The graphs are automatically updated as the underlying data are modified. Graphs are usually represented using bar charts although many other chart types (lines, scatter, pie, area and radar) are also supported and may be integrated as new functionalities to suit the specificities of each laboratory.

3.4.3 Grading Mechanisms

In addition to raw metrics and basic graph representation, we defined an aggregated pivot table (see Figure 4) that provides an overview of the results of the benchmark, thereby facilitating the choice of a data warehousing system or another. For each warehouse, the table summarizes all the metrics and gives a partial grade (derived from the rank) to the warehouse for this particular metric. The table

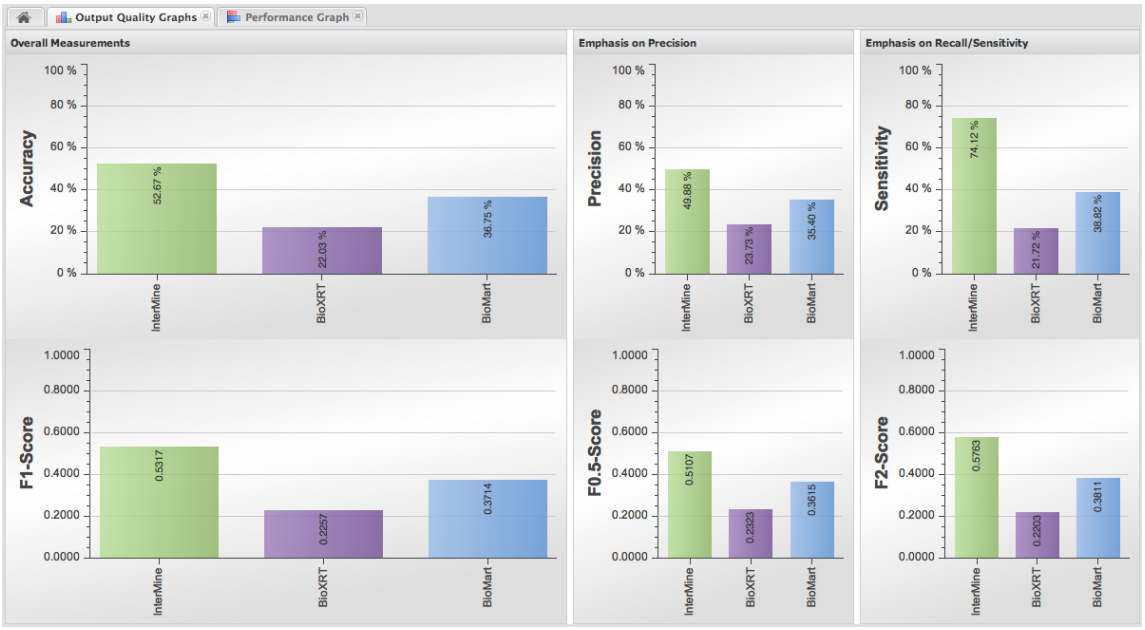

Figure 2: Screenshot of the output quality comparison graphs. The graphs include the accuracy, precision, sensitivity and F-measures.

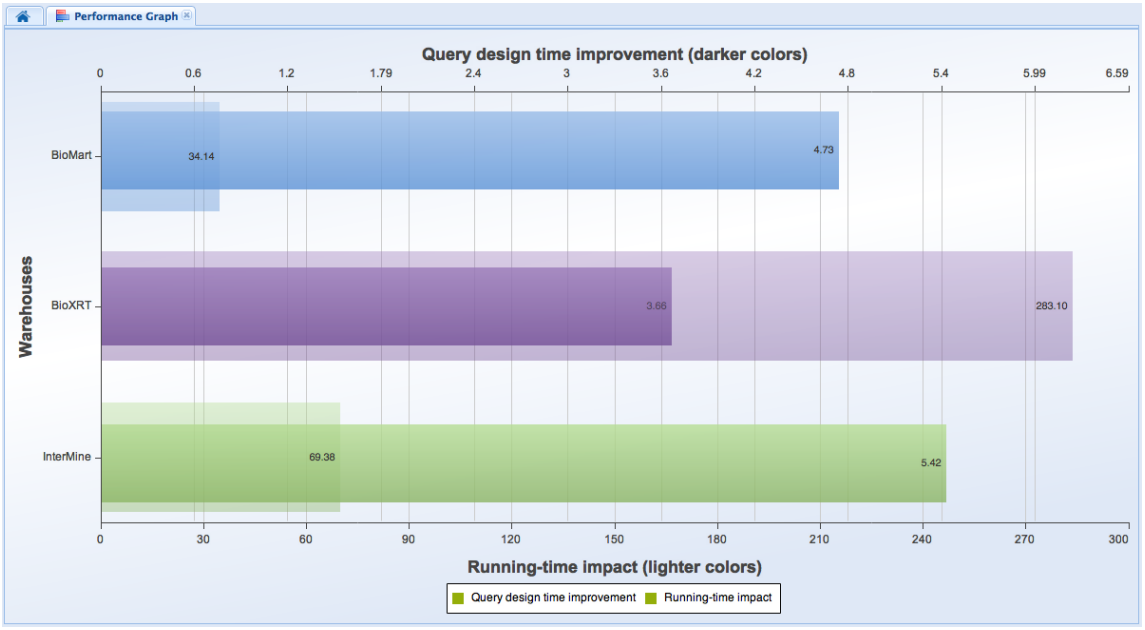

Figure 3: Screenshot of the performance comparison graph featuring the blue skin. The two series – query design-time improvement and running-time impact – may be displayed/hidden in one click.

also provides an overall grade, which is a weighted average of the individual grades. The weights may be modified dynamically by the end user so that the overall grade reflects the priorities and specific needs of each laboratory. Non-relevant metrics may be discarded by assigning the weight '0'. For example in Table 1, the *specificity* and *Matthews Correlation Coefficient*, which rely on the *true-negatives* rate, are not relevant in this case to evaluate the output quality of the data warehouses: their weights were thus zeroed.

4. EXPANDABILITY AND USE CASES

Although it was initially developed to benchmark four biological data warehouses using fourteen predefined typical queries, BenchDW was designed as a general and modular benchmarking framework that is easy to use and configure. In fact, BenchDW may be used not only to benchmark new data warehouses or with other queries, it may as well be used as a generic platform to evaluate other systems. It is for example particularly suitable to compare algorithms,

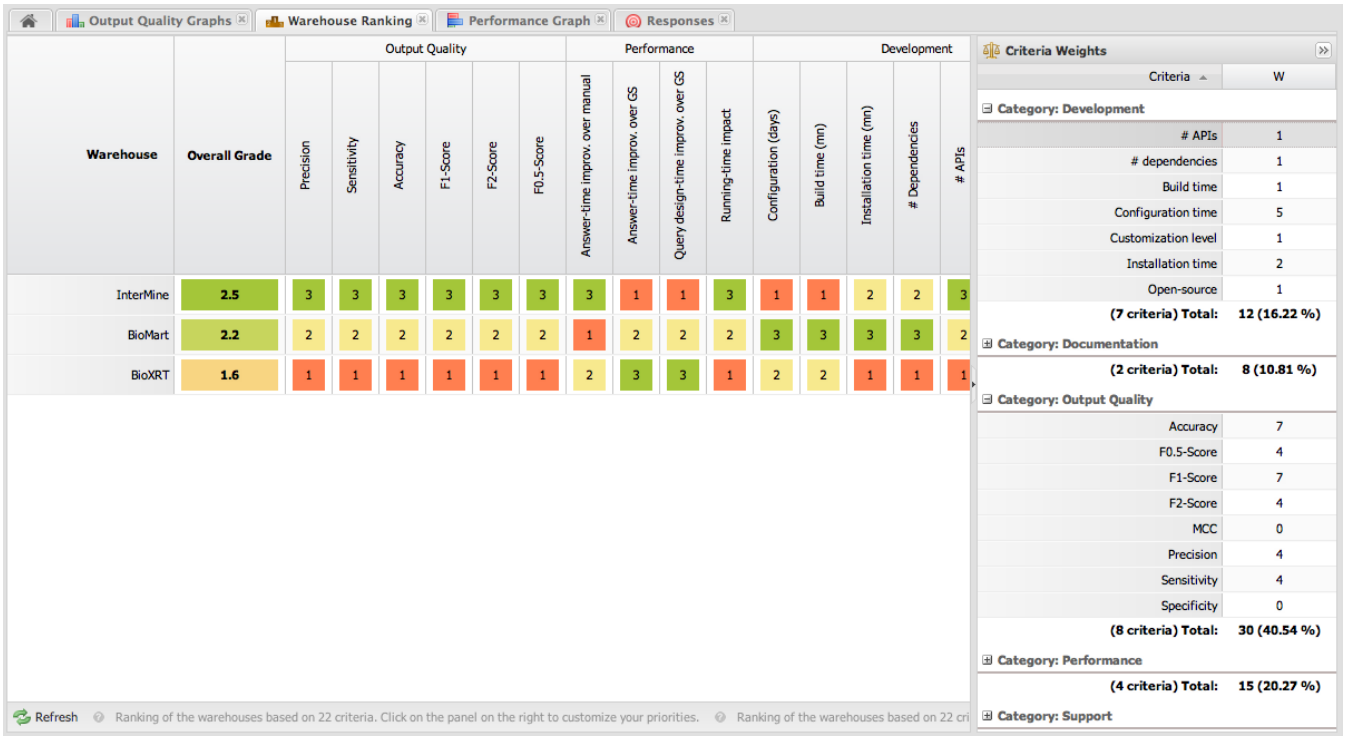

Figure 4: Screenshot of the ranking table which provides an overview of all metrics relevant to a particular benchmark. The criteria in the right panel can be dynamically adjusted to the user's requirements.

such as functional annotation prediction pipelines or other classification and prediction algorithms. Toward that end, one only needs to configure BenchDW by simply editing the two spreadsheets that list the systems to benchmark and the queries to use.

The set of metrics used to analyse and compare the systems can be modified by updating the corresponding database table. In addition, BenchDW is released with an open-source licence. It is thus possible to implement new features with minimal web development skills in a few minutes. BenchDW is also compatible with the PL/R procedural language [4], which may be used by advanced users in place of PHP and standard SQL to implement more sophisticated statistical analysis packages using the R programming language [2]. We plan in the future to setup a public repository to encourage developers to implement and share new modules with the community.

Finally, BenchDW may be helpful to compare hardware configurations and thus identify potential hardware bottlenecks. For example, while it is generally expected that database applications will benefit most from the significantly higher bandwidth of solid state drives, the improvement – if any – is not trivial when evaluating algorithms where performance may be limited by the central processing unit. It is possible to create a new module that will aggregate and compare performance measurements between the two major hard drive technologies – traditional spinning magnetic disks and solid state drives. BenchDW can therefore be used to quantify the benefits that may be gained by hosting the chosen system on an optimized piece of hardware, thereby avoiding unnecessary purchase expenses.

5. ACKNOWLEDGMENTS

This work was supported in part by Genome Canada and Genome Quebec.

6. REFERENCES

[1] 3rd Millennium Inc. Practical Data Integration in Biopharmaceutical R&D: Strategies and Technologies. Technical report, 2002.

[2] J. M. Chambers. Facets of R. *The R Journal*, 1(1):5–8, May 2009.

[3] E. Codd, S. Codd, and C. Salley. Providing OLAP to User-Analysts: An IT Mandate. Technical report, Codd & Date, Inc, San Jose, CA, USA, 1993.

[4] J. Conway. PL/R v8.3 - R Procedural Language for PostgreSQL, 2011.

[5] E. V. Denardo. The Science of Decision Making: A Problem-Based Approach Using Excel. *OR/MS Today*, 28(4), 2001.

[6] EllisLab Inc. CodeIgniter 2.1, 2011.

[7] M. Y. Galperin and X. M. Fernández-Suárez. The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic acids research*, 40(Database issue):D1–8, Jan. 2012.

[8] B. B. Gansel. About the Limitations of Spreadsheet Applications in Business Venturing. In J. Kalcsics and S. Nickel, editors, *Operations Research Proc.*, volume 2007 of *Operations Research Proceedings*, pages 219–223, Berlin, Heidelberg, 2008. Springer.

[9] J. J. Garrett. *The Elements of User Experience: User-Centered Design for the Web*. Peachpit Press, Berkeley, CA, 2002.

[10] J. J. Garrett. Ajax: A New Approach to Web Applications, Aug. 2005.

[11] K. J. Gordon. Spreadsheet or database: Which makes more sense? *Journal of Computing in Higher Education*, 10(2):111–116, Mar. 1999.

[12] P. Karp, S. Paley, and P. Romero. The Pathway Tools Software. *Bioinformatics*, 18:S225–S232, 2002.

[13] R. Lyne, R. Smith, K. Rutherford, M. Wakeling, A. Varley, F. Guillier, H. Janssens, W. Ji, P. Mclaren, P. North, D. Rana, T. Riley, J. Sullivan, X. Watkins, M. Woodbridge, K. Lilley, S. Russell, M. Ashburner, K. Mizuguchi, and G. Micklem. FlyMine: an integrated database for Drosophila and Anopheles genomics. *Genome biology*, 8(7):R129, Jan. 2007.

[14] J. Pemberton and A. Robson. Spreadsheets in business. *Industrial Management & Data Systems*, 100(8):379–388, 2000.

[15] PostgreSQL Global Development Group. PostgreSQL 9.1, 2011.

[16] E. Rogers. *Diffusion of Innovations, 5th Edition*. Free Press, New York, NY, USA, 2003.

[17] Sencha Inc. Web Application Development with Ext JS 4.0, 2011.

[18] T. Triplet and G. Butler. Systems Biology Warehousing: Challenges and Strategies toward Effective Data Integration. In *3rd International Conference on Advances in Databases, Knowledge, and Data Applications*, pages 34–40, St. Maarten, 2011. IARIA.

[19] T. Triplet and Gregory Butler. A benchmark of biological data warehouses. *Briefings in Bioinformatics*, (To be submitted), 2012.

[20] W3schools. Browser Display Statistics, 2011.

[21] Wiredesignz. HMVC: Modular Extensions for CodeIgniter, 2011.

[22] J. Zhang, G. E. Duggan, R. Khaja, and S. W. Scherer. BioXRT: a novel platform for developing online biological databases based on the Cross-Referenced Tables model. In *3rd Canadian Working Conference on Computational Biology*, Markham, Canada, 2004.

[23] J. Zhang, S. Haider, J. Baran, A. Cros, J. M. Guberman, J. Hsu, Y. Liang, L. Yao, and A. Kasprzyk. BioMart: a data federation framework for large collaborative projects. *Database : the journal of biological databases and curation*, 2011:bar038, Jan. 2011.