

## EnzymeTracker: A Web-based System for Sample Tracking with Customizable Reports

Thomas Triplet<sup>\*§</sup>, Justin Powlowski<sup>†§</sup>, Adrian Tsang<sup>‡§</sup> and Gregory Butler<sup>\*§</sup>

*\*Department of Computer Science and Software Engineering, Concordia University  
1455 De Maisonneuve Blvd. West, Montreal, Quebec, H3G 1M8, Canada  
Email: {thomastriplet@gmail.com} {gregb@encs.concordia.ca}*

*†Department of Chemistry and Biochemistry, Concordia University  
7141 Sherbrooke Street West, Montreal, Quebec, H4B 1R6, Canada  
Email: powlow@alcor.concordia.ca*

*‡Department of Biology, Concordia University  
7141 Sherbrooke Street West, Montreal, Quebec, H4B 1R6, Canada  
Email: tsang@gene.concordia.ca*

*§ Centre for Structural and Functional Genomics  
7141 Sherbrooke Street West, Montreal, Quebec, H4B 1R6, Canada*

**Abstract** — In many laboratories, researchers store experimental data on their own workstation using spreadsheets. However, this approach poses a number of problems, ranging from versioning or sharing issues to inefficient data-mining. Standard spreadsheets are also error-prone as data do not undergo any validation process. In this paper, we propose the EnzymeTracker, a web-based laboratory information management system for sample tracking, as a robust and flexible alternative that aims at facilitating entry, mining and sharing of experimental biological data. The EnzymeTracker features online spreadsheets and tools for monitoring numerous experiments conducted by several collaborators to identify and characterize samples, from their basic functional annotations to their complete enzymatic activity. It also provides libraries of shared data such as protocols, and administration tools for data access control using OpenID and user/team management. Our system relies on a database management system for efficient data indexing and management and a user-friendly AJAX interface that can be accessed over the Internet. The EnzymeTracker facilitates data entry by dynamically suggesting entries and providing smart data-mining tools to effectively retrieve data. It also features a number of tools to visualize and annotate experimental data, and export customizable reports. The EnzymeTracker is available online at <http://cubique.concordia.ca/enzymedb/index.html> under the GNU GPLv3 licence.

**Keywords** — *laboratory information management; enzyme; data warehousing; data integration*

### I. INTRODUCTION

Spreadsheets (like Excel) are 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. Hence, spreadsheets are currently the primary means to store both experimental and manually curated genomics/proteomics data in most laboratories.

#### A. Spreadsheet/database paradigm

Spreadsheets might be sufficient when one needs to organize simple data. However, this approach raises a number of problems as spreadsheets present numerous well-known deficiencies compared to databases when dealing with involved data. As reported in previous studies [1], [2], [3], [4], spreadsheets do not scale up well and, as the spreadsheet will expand to accommodate a growing number of records of increasing complexity, data handling — from data entry to data mining and analysis — will become increasingly cumbersome, hence reducing the utility of potentially valuable information.

Besides the scalability issue, spreadsheets are subject to data redundancy and consequently data integrity loss. For example, if protein annotations should be displayed in different spreadsheets, they will most likely be duplicated in each document. When an annotation is updated in one place, all occurrences elsewhere may not be updated, which will result in multiple inconsistent versions of the same data. In this case, one does not know which versions are obsolete and which version is correct. 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 or deleted.

Spreadsheets are also error-prone and do not facilitate data entry. Typically, any cell can contain any type of data and validation is optional at best. Spreadsheets may even incorrectly infer a data type based on the data, in particular numbers and dates in Excel. Spreadsheets are also inefficient to handle sparse data, both in terms of storage and performance. Storage is less of a concern nowadays as costs have dramatically decreased in the past few years. However, it should still be taken into consideration when handling millions of records, as is often the case in bioinformatics and large-scale studies in general. In contrast, optimized databases lead to speed improvements.

Furthermore, sharing data using spreadsheets proved to be difficult, when possible. For example, a shared Excel spreadsheet can be checked-out and edited by only one user at a time. Other collaborators can only display a read-only copy of the document until changes are committed by the first user. Neither waiting for a user to complete his work or duplicating resources is a practical satisfactory solution in larger work groups.

Finally, spreadsheets provide little — if any — security or access control mechanisms. Spreadsheets can be password-protected. However, the password of the spreadsheet is unique and known by many users, and they do not offer the possibility to select what users or groups of users can see/edit in the document: once opened, any record can be displayed by the user. The password is also embedded within the document and it is therefore not possible to revoke access remotely. Databases on the other hand provide advanced access control mechanisms, and enable system administrators to precisely grant or revoke permissions to users or groups of users to create, view, update or delete resources as needed.

### B. Technology acceptance issue

Despite their deficiencies, spreadsheets have been heavily used by biologists because they offer an intuitive and generic user interface that is applicable to most of their projects. Upgrading from spreadsheets to a more sophisticated laboratory information management systems (LIMS) is not trivial. To be broadly accepted by the scientific community as a valuable replacement for spreadsheets, LIMS need to present the five acceptance characteristics defined by Rogers [5]:

- *relative advantage*: the extent to which the LIMS offers improvements over spreadsheets,
- *compatibility*: its consistency with social practices and norms among its users,
- *complexity*: its ease of use or learning,
- *trialability*: the opportunity to try an innovation before committing to use it,
- *observability*: the extent to which the technology's gains are clear to see.

In this paper, we propose the EnzymeTracker, a generic web-based laboratory information management system for sample tracking, as an efficient and user-friendly alternative that aims at facilitating entry, mining and sharing of samples and experimental biological data. Our system was designed to present the above acceptance characteristics to maximize its utility and features advanced yet intuitive annotation and visualization tools as well as a flexible and customizable report designer.

Sections III and IV give an overview of the unique features of the EnzymeTracker and its web-based graphical user interface respectively. Section V briefly presents a number of visualization tools embedded within our system. Section VI describes data-mining and the generation of reports based on templates. Finally, Sections VII and VIII gives some implementation details and future directions respectively.

## II. RELATED WORK

To overcome spreadsheets limited capabilities, a number of proprietary LIMS have been developed. However, their expensive license reduces their audience to bigger laboratories or to the industry and very few systems are freely available to the general scientific community.

### A. iLAP

Stocker et al. [6] recently developed iLAP, a workflow-driven software for experimental protocol development, data acquisition and analysis. iLAP relies on a relational database and a web-based interface to effectively manage complex work flows derived from biological experimental protocols. Integration of external programs using Java Applets is also possible, in particular the popular image processing library ImageJ [7]. However, iLAP does not manage biological data directly, as data remain in files that should be uploaded and associated with a specific experiment or protocol. It is therefore not possible to search for a particular piece of biological data. iLAP does not provide tools for annotating pictures from experimental results such as SDS-PAGE (Sodium Dodecyl Sulfate PolyAcrylamide Gel Electrophoresis) gels or microplates, nor does it provides facilities to generate reports.

### B. SLIMS

Daley et al. [8] developed SLIMS, a Sample-based Laboratory Information Management System. SLIMS is a web application that provides members of a laboratory with an interface to view, edit, and create sample information. Unlike iLAP, SLIMS leverages the relational database to store and manage biological data. However, its web-interface does not utilize recent advances in web technologies. For example, most data are displayed to the user as static HTML tables, which cannot be dynamically mined nor customized. SLIMS also features a microplate annotation tool. Microplate pictures, though, cannot be uploaded nor

visualized along with their annotations. Similarly, SLIMS supports SDS-PAGE gels, which can be downloaded as plain text files, but may not be properly visualized using the picture of the gel. Reports can be generated and exported, but cannot be customized.

### III. OVERVIEW OF THE ENZYMETracker

Despite their numerous benefits over spreadsheets, database management systems still lack satisfactory user interfaces for data analysis [9] whereas Excel spreadsheets do provide intuitive graphical 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 [10], [11]. They can be executed on any connected workstation, without software installation nor specific requirements besides a recent web-browser and an active Internet connection to remotely access data. Web applications 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 different workstations, effectively solving synchronization issues between local copies.

The EnzymeTracker was thus designed as an integrated collection of online spreadsheets accessible over the Internet and backed-up by a relational database for efficient data management. It features a number of novel online tools to facilitate data entry and visualization. The EnzymeTracker also provides a library of shared records such as experimental protocols for sample assays and a comprehensive set of reporting and system administration tools.

Figure 1 gives an overview of the graphical user interface (GUI). Most pages are composed of three panels: the main menu (A) on the left, a spreadsheet (B), which is the primary means to enter to enter data, and a panel at the bottom (C), whose content depends on the data to display. Others data entry means are presented later in sections V and IV-B. Panels A and C can be dynamically collapsed and resized to customize the workspace as needed. Spreadsheets may also be customized by displaying, hiding, reordering and resizing columns as needed so that only the most relevant data are displayed.

The content of the lower panel (C) varies with the data being shown. On most pages, the panel displays the record selected in the spreadsheet in a more readable format. Depending on the spreadsheet, it can provide links to cross-referenced databases such as the Gene Ontology [12] or the Clusters of Orthologous Groups of proteins (COG) database [13]. It also automatically fetches complete references from the literature using PubMed's public API (<http://eutils.ncbi.nlm.nih.gov/>) given the PMID of an article and jobs for nucleotide or protein sequence alignment can be submitted to NCBI's BLAST server in one click (F).

### IV. DATA ENTRY

The spreadsheet (Figure 1B) is the primary means of entering data in the EnzymeTracker. 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. Figure 1D illustrates the utilization of a combo box to select a clone in the page for *Annotations*. 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 provided for Boolean flags and dates. The EnzymeTracker also supports rich text editors with text formatting capabilities, which are mainly used for comments and free-text cells.

#### A. Data integrity and validation

To further reduce entry errors, each cell editor can be associated with a *validator*. Validators, which are usually based on regular expressions or more advanced customized functions, check the correctness of data types and send immediate feedback to the user in case of an error. Validators are also useful to enforce data entry conventions and consistency within a group of users.

In addition, to minimize data entry, cells are automatically computed whenever possible. For example, the length of a protein sequence and its molecular weight (E) are automatically calculated when one enters a protein sequence. Calculated fields are also used to reduce data redundancy compared to standard spreadsheets. For instance, the name of a protein should appear on several related spreadsheets. Using standard spreadsheets, the user will copy/paste the name of the protein wherever needed. This will lead to inconsistencies between spreadsheets during their update. In the EnzymeTracker, the underlying relational database is leveraged to display the name of the protein in all tables where it is needed. The first benefit is that the protein name is automatically displayed whenever there exists a relation between proteins and the current spreadsheet. Second, changes to the protein data are automatically reflected in all tables. Data in the various online table are therefore always consistent and up-to-date.

#### B. Data importation/exportation

In some cases, the different enzyme assays and characterization of samples were already being recorded using Excel spreadsheets. We therefore implemented importation routines to facilitate the migration process to the EnzymeTracker. From experience, basic data importation by uploading and parsing files is error-prone as files formats and layouts tend to vary between files. For example, one column

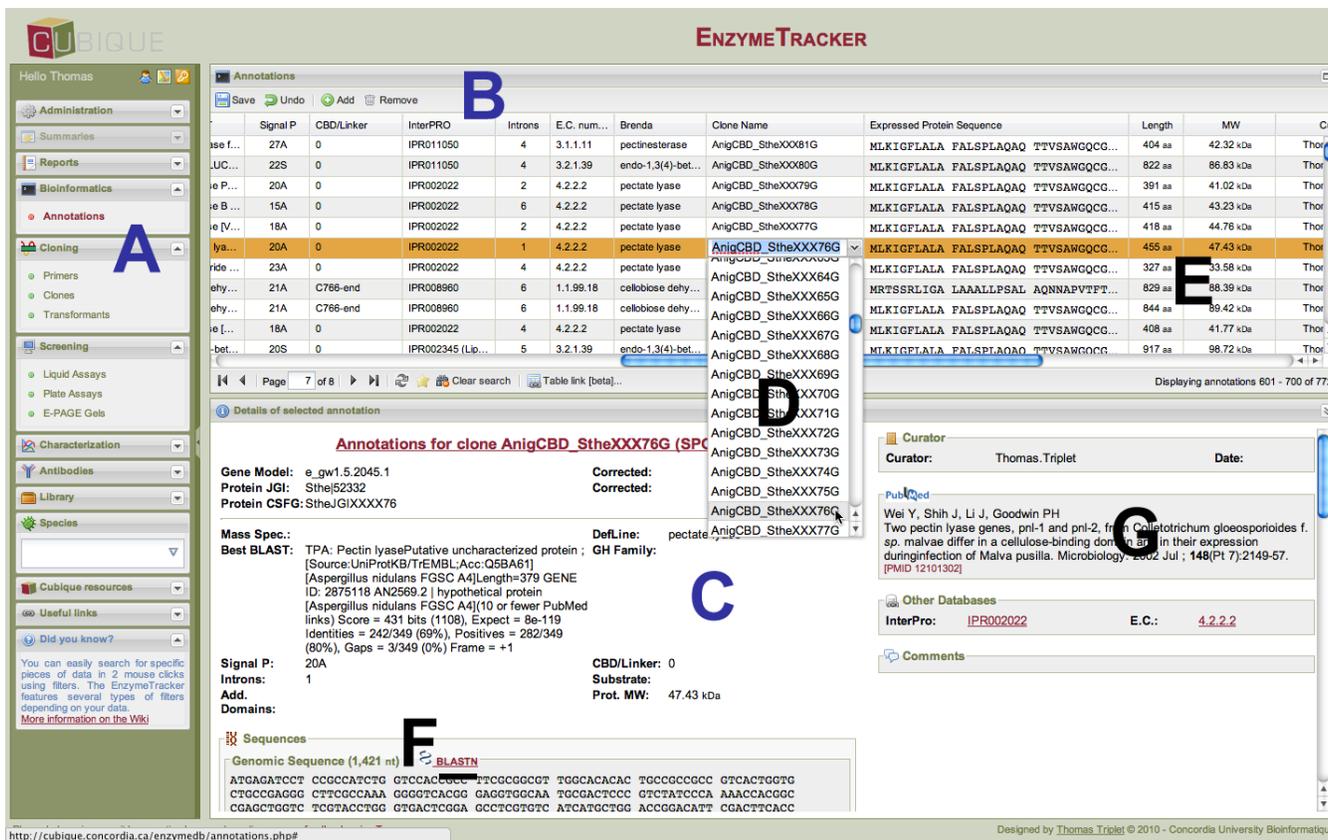


Figure 1. Screenshot of the web-based user interface of the *Annotation* page of the EnzymeTracker. The main menu (A) is on the left. The main panel is usually composed of a spreadsheet (B) and a panel at the bottom to display the entry selected in the spreadsheet using a more readable layout (C). See details in the text.

may be missing in one file, which will shift other columns and lead the parser to import the wrong data.

Instead, we implemented a drag-and-drop importation mechanism where appropriate. The user selects the data to import in the Excel file and drags and drops the selection into the browser’s window. The major benefit of this semi-automatic approach is that it makes it easier for the users to review the data before importation, hence reducing the number of errors made. It also gives more flexibility as only specific records can be selected and imported. Finally, users have the possibility to export EnzymeTracker spreadsheets to Excel documents in one click. Data may also be imported programmatically, using JavaScript and RESTful requests.

### C. Versioning and backups

Our system is supported by a relational database which efficiently handles versioning and backups. Unlike standard spreadsheets, when a record is updated or deleted in the EnzymeTracker, the current version of the record is flagged as obsolete, backed-up and logged for future reference. 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.

## V. VISUALIZATION TOOLS

Most data in the EnzymeTracker can be viewed using tables. In a number of cases however, tables may be improved to give the user a more visual perspective of the data. To enhance the utility of experimental screening data, the EnzymeTracker integrates a number of annotation and visualization tools. Sections V-A and V-B describe in detail how the bottom panel of a spreadsheet (Fig. 1C) can be customized to accommodate plate assays and E-PAGE™48 gels from Invitrogen respectively.

### A. Microplate assays

Microplate assays are widely used in research and drug discovery to detect biological or chemical events of samples. Those events are typically detected by measuring the fluorescence intensity of samples from each of the ninety-six wells (labeled A1 to H12) that compose a plate. The plate assay is usually repeated twice, at two different sample dilution factors.

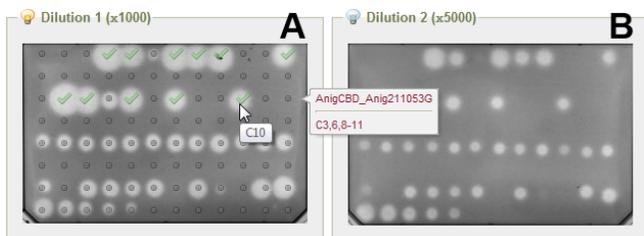


Figure 2. Graphical user interface for the annotations of plate assays. Pictures of the microplates for the two dilutions can be uploaded and automatically annotated based on the content from the tables describing clones and transformants. High-activity wells can be selected within the web interface by clicking on picture. Annotations can be laid over the picture (A) or hidden (B) as needed.

The EnzymeTracker enables users to upload the two microplate pictures for the two dilutions of each experiment (Figure 2). The tables describing clones and transformants are leveraged to automatically annotate the plate. A “virtual plate” representing the 96 wells can also be layered over the original picture (A) or hidden (B) as needed. The virtual plate is also convenient to quickly visualize and identify most active wells by simply clicking on the desired wells directly on the picture.

### B. SDS-PAGE gels

E-PAGE™48 gels are improved SDS-PAGE (Sodium Dodecyl Sulfate PolyAcrylamide Gel Electrophoresis) gels broadly used for high-throughput protein separation and analysis. Each gel comprises 48 lanes for samples and 4 marker lanes, which define the ladder of the molecular weights of the proteins on the gel. Similarly to plate assays, the picture of the gel can be uploaded and annotated within the user interface (Figure 3). Each sample lane in the gel (A) can be annotated using a form (B) that is displayed upon click. A tooltip summarizing annotations of a lane is displayed when hovered by the cursor (orange). The drop-down menus in (B) to select the clone and the transformation

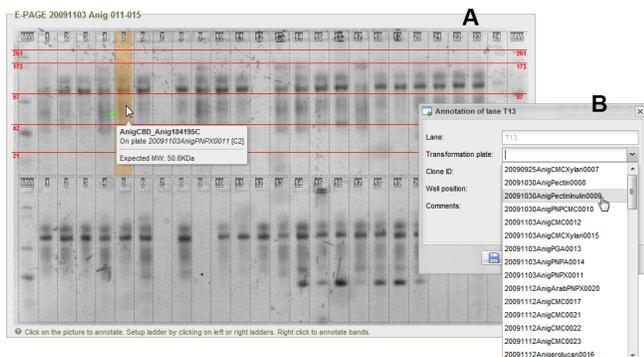


Figure 3. Graphical user interface for the annotations of E-PAGE™48 gels from Invitrogen. Pictures of the gels can be uploaded and annotated within the web interface of the EnzymeTracker.

plate loaded in each lane are dynamically built based on their respective tables. In addition, specific bands can be highlighted (green arrows) and annotated. Finally, the ladder (red) can be easily setup by clicking on one of the four outer marker lanes.

The EnzymeTracker eludes the need for external tools and leverages data from other spreadsheet to facilitate the annotations of hundreds of experimental data and to reduce data entry errors.

### C. Chart visualization

In many cases, the experiments aim at characterizing the evolution of a variable given a set of parameters. Representing the data using charts is then a suitable alternative to tables for data presentation.

The EnzymeTracker fully supports charts, in particular to represent the enzymatic activity characterization of a sample. Graphs are usually represented using curves although histograms and pie charts are also supported. The graph is dynamically updated when the underlying data is edited within the interface or imported from Excel as described in Section IV-B.

Graphs are also used in the administration console, in particular to display connection and data logs.

## VI. DATA-MINING AND REPORTING

As of January 2011, over 55,000 entries have been saved within the EnzymeTracker and a growing number are being recorded on a daily basis. Despite these large amounts of heterogeneous data, scientists routinely need to search for specific pieces of information. For example, a principal investigator may look for “all enzymatic activities detected during liquid assays performed by his assistants in the past two months on clones from *S. thermophile*”.

### A. Context-dependent filtering

Each table in the EnzymeTracker 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 biochemical properties of enzymes and samples, for example protein sequence length or molecular weight, or the temperature stability of a molecule. Boolean filters are typically used to retrieve records when given a flag. For instance, this filter is convenient to list all assays where a strong activity has been reported. Calendar filters are helpful to search for records 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 static or may be dynamically generated by the server based on data from other tables. For example, it is possible to search for samples from a given organism,

the list of organisms being automatically generated by the database server.

### B. Reporting

In order to facilitate data sharing among collaborators, the EnzymeTracker provides a flexible and easy-to-use tool for designing report templates. A report template is similar to other tables within the EnzymeTracker, except that the user can dynamically select the pieces of information he/she wants to share. It is also valuable to aggregate data from various tables and display consolidated statistical data. For instance, one can easily create a report template to display the percentage of transformants which were successfully assayed.

The EnzymeTracker allows users to quickly design a report template and assemble relevant pieces of information together. A preview of the report can be automatically displayed when the configuration of the report changes or when filters are set. The report can also be refined using a number of flags, for example to decide whether to display only current values of a record or its modification log also.

More technically, when a report is designed, the corresponding SQL query is automatically generated based on the report configuration. In other words, the designer effectively enables users to design SQL queries, without writing any code. This is particularly useful for more complex queries, such as non-trivial joins: when a user selects two items from two different tables, it may happen that the two tables are not *directly* related. In that case, a number of intermediate tables must be used in order to join the two tables.

For example, consider the case when the user needs to list the plate assays performed on clones related to cellulase.

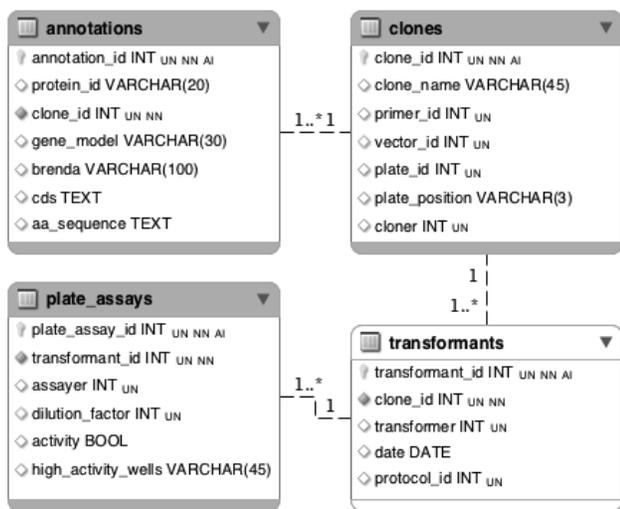


Figure 4. Simplified EER to illustrate reporting mechanisms. Foreign keys are marked with solid bullets. Shaded tables are explicitly listed in the configuration of the report. The *transformants* table (white) is not listed but is implicitly required to perform the join query.

Figure 4, which illustrates a simplified Entity-Relationship diagram of tables relevant to generate this report, shows that plate assays are performed on transformants, not on clones directly. Fortunately, transformants are related to clones, hence it is possible to define an *implicit* relation between clones and plates assays using transformants.

We defined the *cost* of a join between two tables as the length of the shortest path between the two tables in the undirected weighted graph implied by the database structure, where the nodes represent tables and edges, foreign keys. Using the above example, the cost of the join between *clones* and *plate assays* is 2.

We designed an algorithm to determine the optimal implicit joins to relate two tables, which is derived from Dijkstra’s shortest path algorithm. The optimal join is defined as the path between the two tables with the lowest cost. Edges were weighted based on the *biological* significance of the foreign keys. For instance, because of the normalization of the database, a number of intermediate joining tables are created to define the relationships between *real* biological entities — in particular in *m : n* relations — which incorrectly increases the cost of the relationship as the path between the two biologically meaningful tables is longer. The cost of edges in *m : n* relations was therefore reduced to avoid the bias induced by the normalization process during the database design.

When the configuration of the report is updated, the optimal join is computed and executed by the SQL engine. The results of the query are finally used to build the configuration of the ExtJS spreadsheet used to display the report. Advanced users can also create a report by typing the SQL query directly.

Once a template is created, it can be shared and displayed like other tables. In particular, the report can be further refined using filters as described in section VI-A. In addition, reports are automatically updated as more data is added to the EnzymeTracker: there is therefore no need to re-design a report to display up-to-date data. Finally, reports can be easily shared with collaborators or saved as standard Excel files for further analysis.

## VII. IMPLEMENTATION AND AVAILABILITY

The EnzymeTracker aims at providing an interactive web-based user interface. To achieve this goal, the EnzymeTracker of composed of a set of highly dynamic web pages implemented using AJAX (Asynchronous JavaScript and XML) web technologies [11], 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. The web user interface of the EnzymeTracker was implemented using ExtJS, the general Asynchronous JavaScript and XML (AJAX) framework from Sencha. It is backed-up by the freely available

MySQL relational database management system. The server-side code was implemented using PHP 5.

The EnzymeTracker and its documentation are available at <http://cubique.concordia.ca/enzymedb/index.html> under the GNU General Public License version 3.

#### VIII. USAGE AND FUTURE DIRECTIONS

The EnzymeTracker 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. The EnzymeTracker has been successfully used for 15 months on a daily basis by over 50 scientists to monitor protocols and experiments conducted to identify, annotate and fully characterize thousands of samples from multiple fungal species.

The initial implementation of the EnzymeTracker has focused on facilitating sample tracking and experimental data annotation and visualization. The future development of the EnzymeTracker will focus on the implementation of widgets based on the online spreadsheets, which will facilitate data sharing as widgets can be embedded in virtually any web page. We will also enhance reporting by allowing chart generation in addition to tabular data. Finally, the EnzymeTracker will be expanded to enable bar-coding of samples using QR codes (two-dimensional matrix codes), which will facilitate the identification, physical tracking and long-term storage of samples.

#### ACKNOWLEDGEMENT

We thank Annie Bellemare, Noutcheka St-Felix, Marek Krajewski and their teams for providing data and their valuable feedback.

#### FUNDING

This work was supported by the Cellulosic Biofuel Network, funded by Agriculture and Agri-Food Canada.

#### REFERENCES

- [1] K. J. Gordon, "Spreadsheet or database: Which makes more sense?" *Journal of Computing in Higher Education*, vol. 10, no. 2, pp. 111–116, Mar. 1999.
- [2] J. Pemberton and A. Robson, "Spreadsheets in business," *Industrial Management & Data Systems*, vol. 100, no. 8, pp. 379–388, 2000.
- [3] E. V. Denardo, "The Science of Decision Making: A Problem-Based Approach Using Excel," *OR/MS Today*, vol. 28, no. 4, 2001.
- [4] B. B. Gansel, "About the Limitations of Spreadsheet Applications in Business Venturing," in *Operations Research Proceedings*, ser. Operations Research Proceedings, J. Kalcsics and S. Nickel, Eds., vol. 2007. Berlin, Heidelberg: Springer, 2008, pp. 219–223.
- [5] E. Rogers, *Diffusion of Innovations, 5th Edition*. New York, NY, USA: Free Press, 2003.
- [6] G. Stocker, M. Fischer, D. Rieder, G. Bindea, S. Kainz, M. Oberstolz, J. G. McNally, and Z. Trajanoski, "iLAP: a workflow-driven software for experimental protocol development, data acquisition and analysis." *BMC bioinformatics*, vol. 10, p. 390, Jan. 2009.
- [7] W. Rasband, "ImageJ," Nov. 2010. [Online]. Available: <http://imagej.nih.gov/ij/>
- [8] D. Daley, M. Lemire, L. Akhbir, M. Chan-Yeung, J. Q. He, T. McDonald, A. Sandford, D. Stefanowicz, B. Tripp, D. Zamar, Y. Bosse, V. Ferretti, A. Montpetit, M.-C. Tessier, A. Becker, A. L. Kozyrskyj, J. Beilby, P. A. McCaskie, B. Musk, N. Warrington, A. James, C. Laprise, L. J. Palmer, P. D. Paré, and T. J. Hudson, "Analyses of associations with asthma in four asthma population samples from Canada and Australia." *Human genetics*, vol. 125, no. 4, pp. 445–59, May 2009.
- [9] E. Codd, S. Codd, and C. Salley, *Providing OLAP to User-Analysts: An IT Mandate*. San Jose, CA, USA: Codd & Date, Inc, 1993.
- [10] J. J. Garrett, *The Elements of User Experience: User-Centered Design for the Web*. Berkeley, CA: Peachpit Press, 2002.
- [11] —, "Ajax: A New Approach to Web Applications," Aug. 2005. [Online]. Available: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>
- [12] The Gene Ontology Consortium, "The Gene Ontology (GO) project in 2006," *Nucleic Acids Research*, vol. 34, pp. D322–D326, 2006.
- [13] R. L. Tatusov, N. D. Fedorova, J. D. Jackson, A. R. Jacobs, B. Kiryutin, E. V. Koonin, D. M. Krylov, R. Mazumder, S. L. Mekhedov, A. N. Nikolskaya, B. S. Rao, S. Smirnov, A. V. Sverdlov, S. Vasudevan, Y. I. Wolf, J. J. Yin, and D. A. Natale, "The COG database: an updated version includes eukaryotes," *BMC Bioinformatics*, vol. 4, p. 41, 2003.