Center for Eukaryotic Structural Genomics

Protein Structure Initiative

Implementation of PepcDB Reporting at CESG: More Trials and Fewer Tribulations

Craig A. Bingman, Xiaokang Pan, Gary Wesenberg, and George N. Phillips, Jr.

University of Wisconsin-Madison, Department of Biochemistry, 433 Babcock Drive, Madison, Wisconsin, USA 53706-1549, http://www.uwstructuralgenomics.org

Abstract

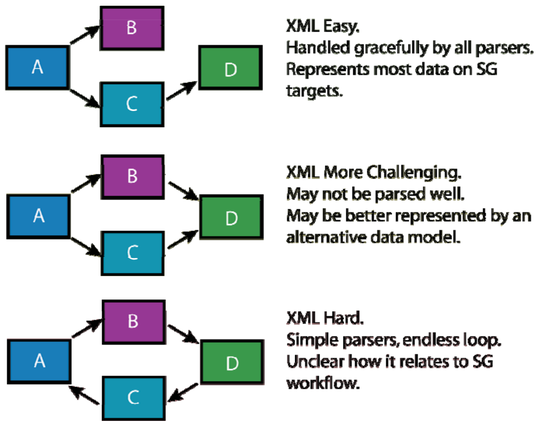
The Protein Expression and Purification DataBase (PepcDB) was created to capture data from the Protein Structure Initiative's Centers. The idea is that such data, including positive and negative results from cloning, expression, purification, and crystallization trials would advance the field and encourage complementary efforts among scientists. PepcDB extends the content of the target selection database (TargetDB) by including status history, stop conditions, text protocols, experimental details, and contact information. The specification of data fields is fairly mature and stable, consisting of an XML schema which is documented by the Protein Data Bank (http://pepcdb.pdb.org/).

The Center for Eukaryotic Structural Genomics (CESG) has been proactive in developing experiment tracking data, primarily with its Sesame laboratory information management system. Sesame gathers data, allowing a complete trace from initial selection to final publication. Standard reports from Sesame are then used to prepare a weekly PepcDB update file. Our most recent report contained experimental details for each distinct trial on a total of 7730 CESG targets.

We have found that the concept of a directed graph provides a suitable framework for both visualizing our workflow through multiple trials on a given target, and preparing XML reports for PepcDB. In particular, work on each target can be described as a finite directed acyclic graph. The sources are selection actions, and the sinks are the most advanced pipeline actions for each trial. An algorithm has been devised that accurately identifies sinks and correctly threads each experimental trial back to its source. Significantly, our current approach does not require that targets flow monotonically toward more "advanced" pipeline stages, supporting recycling of targets into multiple destination vectors at the cloning stage, lateral transfers of samples between NMR and crystallography, and "on the fly" creation of novel experimental paths. CESG has leveraged the substantial body of work devoted to rendering graphs. We currently use the open-source Graphviz software for visualizing linkages between experimental data stored in Sesame.

Preparation of PepcDB reports continues to require a substantial amount of effort, not only for the bioinformatics division, but also from all pipeline units entering data into Sesame. In particular, the retirement of old, multi-stage protocols, creation and proper implementation of the "atomized" protocols required by PepcDB required a substantial retraining effort across the entire project.

We are also preparing to provide our crystallization images to PepcDB, which could be useful for developing algorithms for both crystal detection and optimization of conditions for crystallizing new proteins.



Core Concepts

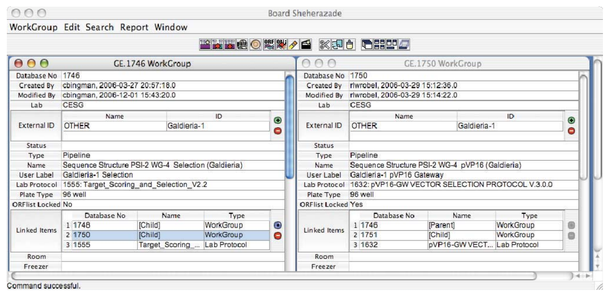
It seems important to enforce directed associations between primary data elements in a database for structural genomics. This is most simply implemented as hierarchical relationships, with one parent having one or more children. Such relationships can be implemented in many types of databases, including familiar relational databases. Additional associations can be present, but it is extremely helpful to have the primary parent-child relationships made explicit and unalterable by end users.

In the implementation of Sesame at CESG, most of these hierarchical relationships exist between individual elements (indices, wells, targets) in workgroups, and the position of the same target in a parent workgroup. Most often, database workgroups map to physical 96-well plates, but the abstraction of a workgroup of elements having independent parentage can be applied to other units of work. This model supports reformatting, merging, and bifurcation of individual targets between workgroups.

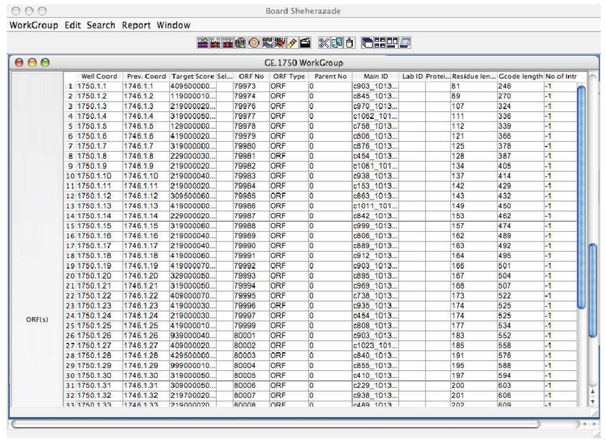
The existence of hierarchical relationships between elements of workgroups leads naturally to expressing the data as a tree or graph. While XML can describe many types of graphs (cyclic elements are possible in XML documents via the use of IDREF), we prefer to conceptualize our XML reports to PepcDB as finite, directed, acyclic graphs.

Sesame Views

Parent and Child Workgroups



Parent and Child Relationships Within a Workgroup



CESG PepcDB Reporting, Mark 2

In 2006, it became apparent that the first version of CESG's PepcDB reporting pipeline needed to be completely overhauled. The code base had been accreting since early in PSI-1. The requirements of PepcDB had changed substantially since its initial conception. Most importantly, the first version of our PepcDB reporting pipeline assumed that targets flowed linearly through the pipeline, from selection to PDB deposition, and never recycled back to previous stages. This assumption is neither necessary, nor desirable. Our production pipeline had evolved to transfer clones to multiple destination vectors from an initial PCR and sequencing event. This is a time- and cost-effective approach for our project, but placed unique burdens on PepcDB reporting software.

The second version of our PepcDB data pipeline, has been coded in Perl, to transfer the structural genomics data from CESG Sesame database to NIH PepcDB. In general, the generation of structural genomics data for an ORF target can be viewed as one or more trees, each with a "Selection" action related work group as a root, next work groups as next tree nodes. "Work Stopped" and "PDB Deposited" are special actions that in our software always denote terminal leaves. Otherwise, the most extended node (a node that has no children and is not a "Selected" action) denotes a terminal leaf. Branches tracing a leaf to the root can be regarded as a PepcDB reportable trial. Based on this assumption, the data pipeline reads in ORF, parentage and action data from a standard Sesame report and then classifies each ORF/target into one or more trials, using a recursive algorithm to read data from each leaf to the root of an ORF tree. After that, the program puts these trials with a set of the status histories and related trial sequences into XML format according to the PepcDB schema. This data pipeline also takes protocols, protein and DNA sequences, and related researcher information, which were downloaded from the CESG Sesame database, and then parses them into XML format data.

CESG PepcDB Reporting, Future

At present, all CESG PepcDB reports are handled by a data pipeline outside of Sesame, using standard formatted and XML reports generated by the database. Sesame already generates TargetDB reports, and in the near future it is expected that work on the mark 2 PepcDB data pipeline will be obviated by Sesame's generation of all CESG PepcDB reports.

One of the prerequisites for this switch was a completely stable core PepcDB schema. We observe that the most recent update to the PepcDB schema has not disturbed any fundamental structures, but it rather has additions and non-disruptive elaborations on a mature core schema. We judge that the requirements are sufficiently stable at this point to transfer the process from the realm of "real-time" programmers outside the database, and move it into our project LIMS.

We are also adding several new data fields to our database to formalize the capture of various experimentally derived target/sample parameters, such as elemental analysis data and activity data. Tags exist in the new schema for reporting these items.

Ensuring Data Integrity

Once a project starts reporting data to PepcDB, the validation tool provided may be helpful. However, by the time a project actually starts reporting data to PepcDB and realizes that there are problems, it may be very late and very expensive to correct these errors.

CESG has found the following measures to be very helpful in assuring that the underlying data is correct, and that there is at least hope of making a scientifically useful report.

1. Adequate training of database users before they interact with the database.
2. Guarding against user errors that potentially disturb parent-child relationships in the database by locking database entities by default.
3. Disallowing users from disturbing parent-child relationships. In our case, these relationships become cast in stone once an action is placed on a database item.
4. Providing users with a tool for visualizing relationships between data items.
5. Simulating ideal data for new protocols and procedures and evaluating its impact on PepcDB reports before implementing the new protocols in the pipeline.
6. Providing definitions for actions, viewable within the database.
7. Assuring that all units agree on definitions of actions.
8. Assuring that action strings are uniquely defined across all protocols.
9. Project-wide rehearsals of data entry, from "Selection" to "PDB Deposited." We have recently conducted a series of such exercises. Although there was initially some resistance, it very soon became obvious to all participants that we were unifying and consolidating our process. Data-entry bottlenecks were identified. All participants gained a greater appreciation for the activities of members upstream and downstream of their usual work area.

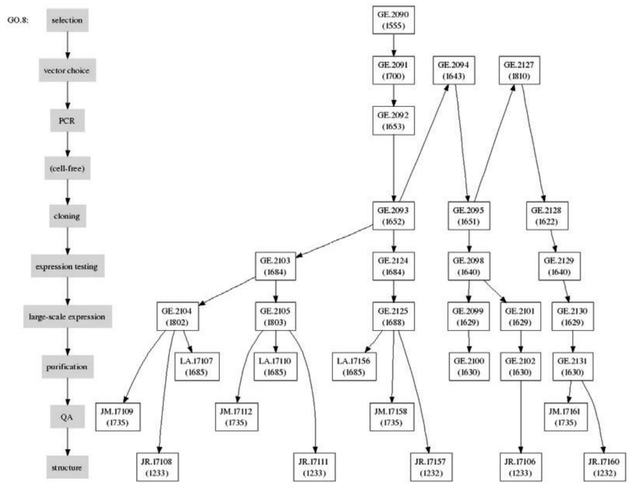
Graphviz and DOT, Global Data Views

During our analysis of project data stored in Sesame, it became apparent that a large number of data entry errors were traceable to the fact that users were not setting up proper child-parent relationships in workgroups derived from existing workgroups.

We embarked on a search for a suitable tool for visualizing relationships between linked database items. Our search led us to the graph description language Dot, and the related GraphViz package. These programs have several important attributes that make them ideal for this task.

1. DOT has a very simple syntax, making it applicable to a wide variety of problems.
2. DOT can easily describe either directed or undirected graphs.
3. There are a number of extremely powerful minimizers that interpret DOT and produce interpretable graphs. This allows us to defer all or some of the "layout problem" to preexisting code.
4. Either zero, one, or two coordinates in a graph described by DOT can be constrained. The ability to constrain one coordinate, as to a conceptual or physical stage in a pipeline process is highly desirable, as is the ability of the minimizer to spread the graph out in the other dimension without additional layout information.
5. There are programming tools for producing DOT graphs for a number of different programming languages. APIs are available for Perl, Python, Java, and a number of other commonly used languages.
6. There are CGI tools available for delivering dynamically generated, hyperlink-rich DOT graphs to web browsers.

A Sample DOT Graph of Structural Genomics Data

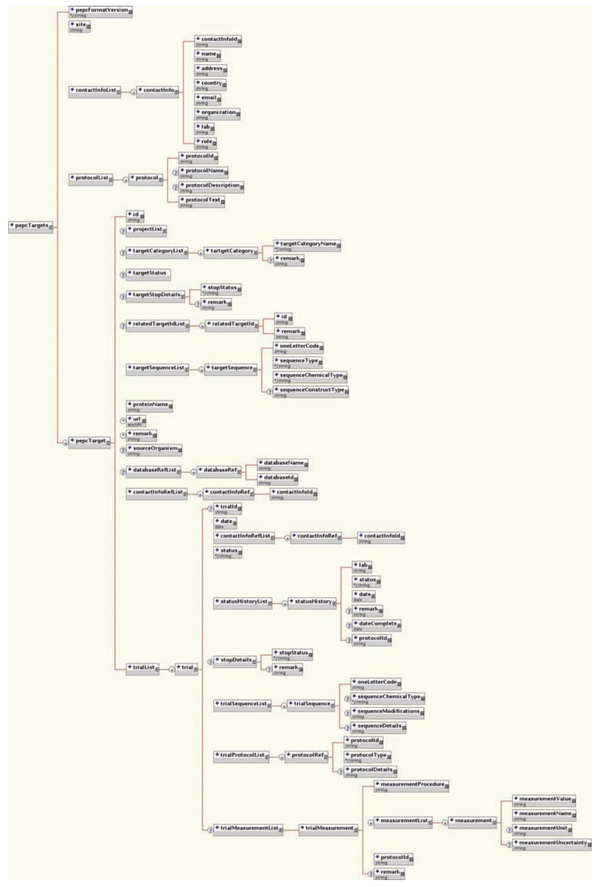


Future Plans

We plan to report all positive and negative crystallization outcomes. All data on crystallization screens since year 2 of PSI-1 has been collected at a level of detail that meets or exceeds PepcDB expectations. In terms of hard disk space, the volume of crystallomics data accumulated dwarfs all other project data aside from primary data from the structure determining branches.

1. Uniform implementation of the existing Crystal view in Sesame, for reporting all details of cryoprotection and other experimental manipulation to PepcDB, as well as reporting these details to the PDB.
2. Implementation of the newly available data tags in PepcDB v9.2.2 XSD.
3. Test export of a limited number of crystallization images, to establish a reasonable framework for transferring this data to RCSB. Our project has well over 1TB of crystallomics image data.
4. Ongoing project attempts to correct data entry errors in Sesame.
5. Limited remediation of PSI-1 data.

PepcDB 9.2.2



References

PepcDB: http://pepcdb.pdb.org

GraphViz http://www.graphviz.org

Dot User's Manual http://www.graphviz.org/Documentation/dotguide.pdf

Chen, L., Oughtred, R., Berman, H.M. and Westbrook, J. TargetDB: a target registration database for structural genomics. Bioinformatics 20(16) 2860-2862.

Berman, H.M, Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. (2000) The Protein Data Bank. Nucleic Acids Research, **28**, 235-242.

M. Carpano. Automated display of hierarchical graphs for computer aided decision analysis. *IEEE Transactions on Software Engineering,* SE-**12**(4):538-534, April 1980.

Stephen C. North. Neato User's Guide. Technical Report 59113-921014-14TM, AT&T Bell Laboratories, Murray Hill, NJ, 1992.

Zolnai, Z., Lee, P.T., Chapman, M.R, Newman, C.S., Phillips, G.N., Jr., Rayment, I., Ulrich, E.L., Volkman, B.F. and Markley, J.L. (2003) Project management system for structural and functional proteomics: Sesame. JSFG, **6**, 143-7.

Wrobel RL, Bingman CA, Jeon WB, Song J, Vinarov DA, Frederick RO, Aceti DJ, Sreenath HK, Zolnai Z, Vojtik FC, Bitto E, Fox BG, Phillips GN Jr, and Markley JL. (2006) **Structural Proteomics**, in Plant Proteomics, Ed. Finnie C., Blackwell Publishing, Oxford, 99-128.

CESG is supported by the National Institute of General Medical Sciences through the Protein Structure Initiative grant number U54 GM074901.