

caBIG Workspace Developer Project Form

Developers, please complete this form in advance of the caBIG kickoff meeting and return by e-mail to adamsm@mail.nih.gov. Completed forms will be made available to all participants in advance of the meeting to enhance workspace discussions. During our conversations with you, we expressed the aspect of your program that we would like you to develop in the first year of the caBIG pilot; it is this we are asking you to address - here and in your presentation.

1. Sponsoring Cancer Center: **UCSF Cancer Center**
2. Workspace: **Integrative Cancer Research Workspace**
3. Project or Activity: **Tools for heterogeneous data analysis and data sharing of complex data sets from UCSF Cancer Center Investigators**
4. Workspace needs the project meets: **Tools for analysis of complex data sets that are accessible by non-computational experts. Community data sharing of valuable and unique multi-modal data sets (ESP, expression, CGH, etc...)**
5. Stage of project maturity (Conceptual, early beta, regular end-user use at parent center, regular use in the community): **Varies. We have tools/data at all stages of the continuum.**

See attached page for an outline that addresses the following:

6. Technical details of Tools
 - a. Software Architecture (These will likely be preliminary)
 - i. System design Component details Relevant standards UML schematics (if valid)
Size of project installed software base
 - b. Development Environment (tools, languages, bug tracking, etc.)
7. Does the project make use of existing standards? If so, what are they?
(e.g. bioinformatics standards such as MIAME for microarrays, or software standards such as XML)
8. Does other software in the community meet this need? Is this software open source? Can it be harnessed?
9. Points of possible interoperability with other caBIG systems
(This might include communication with other caBIG databases, use of caCORE APIs, caBIG-compatible APIs, etc.)
10. What resources are proposed to achieve caBIG interoperability?
 - a. Developmental requirements
 - i. Software (re)engineering
 - ii. Standards adoption
 - iii. Platform migration
 - b. Infrastructure
 - i. Facilities
 - ii. Management tools
 - iii. Personnel
11. Draft 12-month work plan, with milestones to achieve caBIG interoperability.

Our Initial 1-page Proposal:

We believe that success of the CaBIG effort depends critically on demonstration that both experimental and computational researchers make active use of the products of the effort: experimental and clinical data, tools and applications, linkable knowledge heaps (e.g. caBIO), and access to a community of developers and users. These products must be both available and usable. When the tools and data from CaBIG start to become integral parts of important papers reporting basic research advances in cancer biology and advances in complex biological data analysis, a substantial milestone will have been met.

Toward this end, the UCSF Cancer Center, using the Jain Lab as a hub, is in an excellent position to make a contribution to CaBIG as a Pilot Center. We are proposing a project built on a systems-oriented cancer biology concept that is part of multiple SPORC programs within UCSF. The deliverables to CaBIG include: 1) experimental data of multiple types [array-based expression, array-based CGH, end-sequence-profiling (ESP), proteomic, and phenotypic information], and 2) tools for quantitative analysis of such complex heterogeneous data in the context of a broad annotation space [pathway structure, genomic mapping, gene function, etc...]. The *data* will be made broadly accessible by collaboration between UCSF and NCICB to host and share the data, tightly linked with the caBIO infrastructure. The *tools* will be made broadly accessible in two ways. First, working with NCICB, we will augment our tools to interact directly with the caBIO infrastructure. Second, the tools, along with appropriate web-based tutorials for end-users and documentation for developers, will be shared freely via the CaBIG web presence.

Two examples will make this concrete. In the case of array-based CGH, UCSF Cancer Center investigators (Pinkel, Albertson, and Gray) have developed a high-density, pan-genomic method for profiling tumor genomes for abnormalities in DNA copy number. We have been working with NCICB to make such data widely available in a useful format, including extensive description of the data type, sharing of array-CGH profiles, and discussion and testing of tools (e.g. the array-CGH viewer application). We are proposing to expand this effort to include all data types being collected in our cancer cell line systems (focusing initially on the breast cancer system). The goal is not only to share the data, but to make it *useful* through linkage with caBIO and by having experimentalists make active use of the hosted data using CaBIG applications. The data we propose to include spans DNA (array-CGH, ESP), RNA (expression), protein (from proteomic measurements), and multiple phenotypic characterizations in systems of over 50 tumor-derived cell lines.

The second example is in the computational tools arena. We are developing a data analysis system that is capable of storing, manipulating, and analyzing multiple types of experimental data and associated annotation data (called Magellan). In the proposed project, Magellan will be augmented to make direct use of the caBIO infrastructure so that users of the system will be able to ask sophisticated questions about their data in the context of information such as genomic mapping or gene function (or computed information from other data sets). For example, upon loading array-CGH and expression data with typed identifiers (e.g. BAC IDs and Genbank Accessions) into Magellan, users will gain access to linkable annotations such as genomic mapping position. Given such information, users can immediately examine the relationship between genomic copy number and expression of genes at loci whose copy number varies. Analysis of this type is currently possible with Magellan, but the user must provide annotations of data directly. The caBIO infrastructure offers a stable and accessible source of relevant information that can be projected onto complex data sets. The tools that we propose to develop under the pilot project include Magellan and a set of tools for modeling and studying pathways in the context of quantitative measurements. The pathway-oriented tools are novel, and their aim is to go beyond pathway visualization toward pathway induction and augmentation. Both Magellan and the pathway tools are being developed specifically for study of the data proposed for inclusion in the pilot in the foregoing text. The tools have great and obvious synergy with many aspects of what is being developed at NCICB as part of CaBIG. Our goal is to exploit this synergy, make the tools broadly applicable and useful in quantitative study of cancer biology, and make the tools widely available via CaBIG.

We propose a two-year pilot project, to be directed by Dr. Jain, with collaboration from Drs. Gray, Collins, Albertson, and Pinkel from within the UCSF Cancer Center. This is to be a joint effort, with participation both at UCSF and within NCICB and possibly other pilot centers. Primary staffing of the project at UCSF will be within the Jain lab. The first order of business will be to establish communication for both the data stream and the tools stream. The immediate deliverables include mature data (expression and CGH from the breast cell line system) and mature tools (UCSF Spot, HGS, etc...). Provision of experimental data and collaboration as end-users of the hosted data (and associated applications) will occur on an ongoing basis for the entire project period, facilitated through the Jain Lab. The measurable outcome of the *data* aspect of the project is the extent to which people *outside* of UCSF are making use of the data, as documented in published peer-reviewed papers. For the *tools* aspects of the project, the short-term goals will require UCSF staff to learn the operational aspects of APIs to caBIO, which may involve direct collaboration with NCICB staff (and possibly travel). By the end of the first year, a first version of the caBIO integrated tools will be delivered, with both user and developer documentation. Continuing development for the second year will make use of additional data being generated at UCSF and will also make consumption of other data housed within the CaBIG infrastructure convenient for analysis. The measurable outcome of the *tools* aspect of the project, apart from simple delivery, is analogous to that for data: the extent to which people outside UCSF make use of the contributed tools in published papers in cancer biology.

Introduction

The UCSF Cancer Center's participation in CaBIG as a functional developer centers around three sub-projects: 1) tools for analysis of complex data sets that are accessible by non-computational experts; 2) tightly integrated tools for quantitative pathway analysis; and 3) community data sharing of valuable and unique multi-modal data sets (ESP, expression, CGH, etc...). The following will address the various considerations in the CaBIG Workspace Developer Project Form. It is organized by sub-project.

Magellan: Primary developer Chris Kingsley

Recent advances in high-throughput genomic and proteomic analysis are generating enormous amounts of quantitative biological data. In the analysis of clinical samples such as tumors, many different types of data can be collected from each sample; Genomic data such as mRNA expression and CGH data may be accompanied by patient information and clinical outcomes. Determining the relationship between genomic and clinical/phenotypic information is arguably the most important use of high throughput genomic data. Genes and genomic loci that correlate with clinical outcomes such as drug response and patient survival may be used to classify clinical samples and/or used as potential targets for therapeutic intervention in disease. A statistically rigorous analysis of genomic and phenotypic information can, therefore, streamline the process of finding diagnostic markers and therapeutic targets.

While there are distinct advantages to comprehensively quantifying biological variables such as mRNA gene expression or genomic copy number, it also presents statistical challenges. Microarray experiments can generate tens of thousands of measurements per sample, and when the ratio of the number of measurements to the number of samples becomes very large, false relationships between variables can emerge. Because so many statistical comparisons are made, strong correlations between variables may be observed even under the null hypothesis, simply by chance. This 'multiple comparisons problem' is one of the most challenging aspects of analyzing large biological data sets.

One method of overcoming the problem of multiple comparisons is to use annotation information as a means of data subselection. Annotations can consist of quantitative information such as genomic mapping data for genes, textual information derived from ontologies of gene function, formal descriptions of regulatory networks, etc. Annotations that describe data can be used to constrain correlative queries by restricting a data set to those variables whose annotations meet specific criteria, thus reducing the number of comparisons based on knowledge that is orthogonal to the outcome variable under consideration. This raises an additional problem, though, in that annotation data for a large array-based experiment may exceed the size and complexity of the primary measurements.

Annotations can also be used to compare and project variables between different data types or different experimental data sets. By projecting variables of different data sets into a common annotation space, direct comparisons between data sets can be made, allowing the user to perform meta-analyses of data from different experiments. Similarly, by projecting different genomic variables into a common annotation space such as genomic position, variables derived from fundamentally different types of genomic information (such as CGH and expression) can be compared and analyzed.

Magellan: A Web-Based System for Generalized Data Analysis

The statistical considerations discussed above are non-trivial, lacking accessible solution in most biologically oriented laboratories, particularly as regards the utilization of heterogeneous annotation information. Conversely, laboratories oriented toward statistics or computer science frequently lack the depth of understanding to bring appropriate biological information to bear on constraining quantitative questions. With this in mind, we have developed a web-based system that allows biologists to perform complex analyses on heterogeneous data in an environment that does not require a background in computer programming or statistics. Since the primary aim of this system is to allow researchers to perform exploratory analyses of their data sets, we have named the system Magellan.

An important feature of this system is its generality; stored data and annotations are treated as abstract entities, such that arbitrary, user defined types of information can be stored. This abstract approach to data and annotation representation is similar to the EAV (Entity-Attribute-Value) approach for modeling heterogeneous data, which allows a flexible means of storing information without limits on the number and type of attributes per entity. Using this approach, a data set is defined as any number of p-dimensional vectors of information that belong to a sample. Likewise, annotations are abstractly defined as textual or quantitative information that describes one or more variables of a given data type, such as genomic position or biological pathway information for a particular gene in an mRNA expression data set. Thus, an arbitrary number of user defined data and annotation types can be loaded and stored in the system, allowing a very broad utility. The use of annotation information is a key aspect of this analytical system. Annotation information can serve multiple purposes: it can place results in biological context,

and it can be useful in the sub selection of data to minimize the effect of multiple comparisons on high dimensional data. For example, correlation between all genes and a clinical outcome may not yield a statistically significant result, but restricting the genes to those in a specific biochemical pathway may reduce the dimensionality of the expression data such that multiple comparisons do not predominate.

Magellan has been designed in a modular fashion, such that new analytical algorithms can be easily interfaced to it. In this way, a wide variety of analytical approaches can be used on any number of different data types. Since data and annotations are stored as general entities, their interpretation is dependent only on the particular analytical method that is used. Many different analytical packages can potentially be interfaced to the system, so while Magellan was originally developed to analyze high throughput genomic data, it can be used to analyze any type of tabular data as long as the appropriate algorithms are deployed.

The generality and modularity of Magellan set it apart from other database and analytical packages that are currently available for the analysis of genomic data. Open source database systems for genomic data have been developed, but they are usually specific for specific types of genomic data (such as mRNA expression data) and particular experimental platforms. In contrast, Magellan has specifically been designed to store virtually any type of data and annotations the user can provide. The modularity of Magellan makes it straightforward for end users with moderate programming experience to add analytical methods to the system.

Magellan Implementation

Magellan is currently used by a number of biological researchers at the UCSF Cancer Center. Over the next year, Magellan will be expanded to include a greater variety of analytical tools, a smoother user interface, and compatibility with caArray. The underlying business logic of Magellan has been developed using Java Server Pages and Apache Tomcat using the ANT build tool. The system currently runs on the Windows 2000 Server operating system, although migration to Linux is possible. All analytical entities such as data types, samples, and annotations are implemented as compiled Java classes. The Java API and all of Magellan's underlying source code will be made publicly available. Currently, all data and annotations are stored in an Oracle 8.1 relational database using a custom database schema.

The analytical methods used by Magellan are modularly deployed. Algorithms can be developed in a number of different environments (Java, C, R, etc) and are directly accessed from within the JSP pages. Since many useful statistical methods have already been developed by third parties (such as the open source bioconductor package), the ability of Magellan to interface with algorithms developed using statistical packages such as R is very advantageous.

Integration of Magellan with caBIG

Magellan currently uses a custom database schema to store and retrieve data and annotations, which are represented by abstracted Java classes. To integrate Magellan with the caBIG initiative, it will be necessary to migrate the database component to the caArray standard of data storage. Since Magellan is itself a Java based system, integration of Magellan with caBIG should be a relatively straightforward use of the API's of the various components of caBIG.

The annotations that are curated as part of the caBIG project will be of particular use in the Magellan system, as it is built around the incorporation of annotation information in the analysis of genomic data. Many algorithms currently deployed in the Magellan system depend on the use of annotations as a means of grouping, subselecting, and visualizing genomic information. The incorporation of caBIG curated annotations will greatly enhance the power and usability of Magellan.

QPACA: Primary developer Barbara Novak

QPACA stands for Quantitative Pathway Analysis in Cancer. It is a pathway modeling and analysis system that supports exploration of quantitative biological data in the context of a pathway description such that: 1) relationships between quantitative measurements and phenotype can be usefully analyzed and visualized in the context of the pathway; 2) hypotheses that gene sets may be participating in a coordinated process or pathway can be evaluated by predictive measures; 3) predictions may be made to augment a pathway to include additional members, and 4) predictions may be made as to the interaction connectivity of gene products assumed to be part of a coordinated process.

At the center of the system is a pathway representation that enables visualization and computational analysis of pathway structure. It is designed to be flexible and extensible in order to support the widest variety of pathway structures and components possible. While other pathway model viewing tools exist (BioCarta, KEGG, etc.), this project is unique in automatically generating graphical depictions which can be annotated with experimental data and in providing pathway analysis tools.

QPACA Implementation

The pathway representation, which is at the heart of the system, is written in Java, utilizing OpenJGraph, an open-source Java library for creating, manipulating and drawing graphs. A pathway is defined as a simple directed graph consisting of sets of nodes, representing objects within the pathway, and edges, representing the interactions between these objects. Each node in the graph is composed of an assembly of one or more gene products, small molecules, processes, or other assemblies. In this way, it is possible to represent single components of the pathway as well as multimolecular complexes and families in an extensible fashion. The edges in the graph represent either activation or inhibition interactions between the nodes. A more extensive set of supported interactions is under development.

Input of pathway information is achieved through either an XML-based description or a simple text-based pathway language. Images of the pathway are generated using Graphviz, an open-source graph drawing software, which can generate many different types of image files, including SVG. The representation can be accessed via a command-line Java programs that query the structure or using a web-based quantitative data visualization module.

The pathway and data visualization module is designed to allow the end-user to visualize quantitative data within the context of pathway structure. It uses Java Server Pages and perl running on a Tomcat web server. Using SVG image files of the pathway produced by Graphviz, this module takes quantitative data from a tab-delimited text file and colors the SVG pathway image based on this data. Additionally, some simple statistics can be computed as similarly visualized (t-test, correlation). The images are clickable and can be linked to outside data sources.

This module, along with both the XML representation and the text-based pathway language, has been made available to several labs at the UCSF Comprehensive Cancer Center. For wider distribution, the interface needs to be made more accessible to the end-user. To achieve part of this goal, the data visualization portions will be integrated with Magellan. Specifically, data input will be through Magellan and user-based annotations for genes within the Magellan system will also be made available to the pathway visualization system.

Under development are two sets of pathway analysis tools that will: 1) support interrogation of specific gene sets for likelihood of participation in a coordinated process and 2) propose novel connectivity in and novel membership into existing pathway based on examination of quantitative biological data. These tools will also be made available via the web-based interface.

Standards: The pathway input is centered around XML, though a more simple text-based pathway language can be used to access the XML representation. Images of the pathway are generated in SVG format. Data input into the analysis tools is currently achieved using a tab-delimited text file.

Integration of QPACA with caBIG

One obvious point of interoperability with other caBIG systems is with caBIO. The http API can easily be harnessed to provide content linked from the pathway images. Additionally, the pathway representation can be made available via a caBIG-compatible API, allowing the development and integration of additional pathway analysis modules.

An API will also be made available of the pathway representation to allow pathway input from various databases (KEGG, BioCarta, caBIO, etc.).

Tools and Data Sharing: Primary developer Taku Tokuyasu

This aspect of our participation addresses two areas: 1) sharing of Spot/Sproc microarray quantitation tools; and 2) array cgh, expression microarray, ESP, and other data sharing. It addresses CaBIG needs including: imaging tools and databases, microarray and gene expression tools, and access to data. Some of these tools are quite mature (e.g. Spot/Sproc are in regular end-user use at parent center with regular use at a number of external sites).

Implementation

UCSF-Spot ("Spot") and Sproc are standalone tools for microarray image quantitation. Spot expects standard 16 bit grayscale tiff images and is designed for two-channel microarrays. It reports image quantitation data in each channel separately, along with log₂ ratio data and a number of quality measures per spot. Sproc is a post-processor for Spot output, averaging over replicate spots that survive user-defined quality criteria. The data is normalized by the global median of the log₂ ratios. The source base comprises roughly 3000 lines of C, with GUI interfaces built from Windows MFC widgets. The code relies on ImageMagick libraries for basic image file handling support. Other non-image inputs and outputs are tab-delimited text files.

Spot and Sproc executables are freely available for download at the following website: <http://jainlab.ucsf.edu/Downloads.html>. The core parts of the source base will be made available as open source

shortly. The GUI portions will be re-implemented in a language such as perl/Tk in order to make them more freely available and cross-platform.

Spot strictly requires only basic geometrical information about an array print, concerning the number of subarrays within an array, and the number of spots within each subarray. Additional information such as the approximate inter-spot spacing can improve its performance. It is not currently designed to handle multiple arrays within a single image, although this can be addressed once the needs of a wider user population are better understood. Sproc utilizes array print information to perform averaging over replicate spots on the array. It uses an additional file to supply clone or gene annotations, which are largely just passed through to the output

Spot and Sproc have been developed and are currently used largely in the context of array cgh technology. They can be used without modification in the analysis of other microarray types, e.g. two-channel cDNA microarrays, although extensions such as the ability to handle multiple arrays on a single slide are envisioned to make them more convenient to use in commonly encountered cases.

In addition to the distribution and further development of microarray analysis tools, we are working on the development of an infrastructure for the sharing of microarray data. The need for such an infrastructure is particularly acute for array cgh data. We have been assisting a group at NCICB in their development of WebCGH (<http://tomcatdev.rti.org/webCGH/index.jsp>), a tool for the visualization and analysis of array cgh data. We expect to beta test WebCGH shortly, perhaps through the installation of a local instance. We also plan to provide facilities for the sharing of expression array data through the local instantiation of established resources such as the Stanford Microarray Database (SMD). The raw image files are probably best stored on networked file servers, with links to these stored remotely in databases.

Standards: The non-image data files for both Spot and Sproc are tab-delimited text files. While many of our users have come to rely on this format (for e.g. ad hoc visualization and analysis within Excel), we will also provide MAGE-ML compliant output, especially as the requirements for integration with caBIO become clearer.

It appears that there is less experience in general with data standards for array CGH experiments than with their expression array brethren. We expect that our ongoing collaboration with the NCICB WebCGH project will aid in fleshing out this aspect of microarray data sharing.

Community Needs: Proprietary software packages similar to UCSF-Spot have been developed elsewhere in academia and industry, such as Spot (developed at U.C. Berkeley and CSIRO Australia) and GenePix (from Axon Instruments). We are not aware of other open-source software that provides the same functionality as UCSF-Spot. Sproc provides relatively simple statistical functionality that can be easily reproduced in an environment such as R. Sproc in addition provides easy integration with Spot and a flexible mechanism for annotation through its input files. It is also highly suitable for efficient batch mode operation.

The usefulness of the annotations in the output of Sproc are largely determined by the information provided to it as input. Sproc users (and microarray print suppliers) can in particular provide annotations that enhance integration with external resources such as CGAP (the Mitelman Database), MMHCC, etc. Such enhancements could also be provided in downstream applications such as Magellan.

Integration with caBIG

To the extent that this last effort consists largely of developed software, there is little integration to be done. We will release the tools, with appropriate documentation, as open-source via the CaBIG web presence.

We will also be acting to a degree as a data conduit to CaBIG for our experimental colleagues, who are eager to share their large data sets with the broader community. This may take place in a number of ways, but one possibility is to engineer appropriate interfaces between our local databases and CaBIG. The other possibility is to directly adopt CaBIG-developed data systems for local hosting at UCSF.