nature biotechnology

The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics

Andrew R Jones[1,2,16], Michael Miller[3], Ruedi Aebersold[4,5], Rolf Apweiler[6], Catherine A Ball[7], Alvis Brazma[6], James DeGreef[8], Nigel Hardy[9], Henning Hermjakob[6], Simon J Hubbard[2], Peter Hussey[10], Mark Igra[10], Helen Jenkins[9], Randall K Julian Jr[11], Kent Laursen[11], Stephen G Oliver[2], Norman W Paton[1], Susanna-Assunta Sansone[6], Ugis Sarkans[6], Christian J Stoeckert Jr[12], Chris F Taylor[6], Patricia L Whetzel[12], Joseph A White[13], Paul Spellman[14] & Angel Pizarro[15,16]

The Functional Genomics Experiment data model (FuGE) has been developed to facilitate convergence of data standards for high-throughput, comprehensive analyses in biology. FuGE models the components of an experimental activity that are common across different technologies, including protocols, samples and data. FuGE provides a foundation for describing entire laboratory workflows and for the development of new data formats. The Microarray Gene Expression Data society and the Proteomics Standards Initiative have committed to using FuGE as the basis for defining their respective standards, and other standards groups, including the Metabolomics Standards Initiative, are evaluating FuGE in their development efforts. Adoption of FuGE by multiple standards bodies will enable uniform reporting of common parts of functional genomics workflows, simplify data-integration efforts and ease the burden on researchers seeking to fulfill multiple minimum reporting requirements. Such advances are important for transparent data management and mining in functional genomics and systems biology.

Biomedical and clinical research fields are increasingly applying a range of high-throughput experimental techniques, often called 'functional genomics', to study direct and indirect products of gene expression, molecular interactions and the cellular environment. Such approaches aim to determine the function of all genes, with 'function' broadly defined to include the relationship to phenotype, interaction partners (e.g., DNA, RNA, proteins, metabolites), localization and responses in expression to external stimuli. A functional genomics approach may use multiple techniques[1,2] in a single study to analyze multiple kinds of data[3,4]. It is widely recognized that significant benefits can result from detailed annotation and archiving of data sets resulting from these types of studies. The benefits include the ability to exchange data with collaborators or submit them to public databases, the sharing of best practice, which provides capabilities for validation of the study or reinterpretation of results, and the development of new algorithms for data analysis. However, functional genomics often involves sophisticated sample processing, complex equipment, rich data sets and intricate data analyses. As a result, describing experiments in a systematic way requires similarly rich data models that enable data to be analyzed, validated and interpreted by people other than their immediate producers.

The challenges of building data standards for functional genomics have been addressed by scientific communities well-versed in microarray and proteomics technologies. Specifically, the Microarray Gene Expression Data Society (MGED, http://www.mged.org/) was formed in 1999 and devised the Minimal Information About a Microarray Experiment (MIAME)[5] reporting requirements. MGED participants also provided a data model, the MicroArray Gene Expression object model (MAGE-OM version 1 (refs. 6,7), to capture MIAME-compliant data. In 2002, the Proteomics Standards Initiative (PSI; http://psidev.sourceforge.net/) was formed by the Human Proteome Organization (HUPO) and has since developed reporting requirements and data formats for protein interactions (PSI-MI[8]) and mass-spectrometry data (mzData, http://www.psidev.info/index.php?q=node/80#mzdata).

[1]School of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, UK. [2]Faculty of Life Sciences, University of Manchester, Simon Building, Brunswick Street, Manchester, M13 9PL, UK. [3]Rosetta Biosoftware, 401 Terry Avenue North, Seattle, Washington 98109, USA. [4]Institute of Molecular Systems Biology, HPT E 78, Wolfgang-Pauli-Str. 16, 8093 Zurich, Switzerland, Faculty of Science, University of Zurich, Switzerland and Center for Systems Physiology and Metabolic Disease at ETH Zurich. [5]The Institute for Systems Biology, 1441 North 34th Street, Seattle, Washington, 98103, USA. [6]European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK. [7]Microarray and Genome Informatics, Department of Biochemistry, Stanford School of Medicine, CCSR, Room 2255a, Stanford, California 94305, USA. [8]GenoLogics Life Sciences Software, Suite 2302-4464 Markham Street, Victoria, British Columbia, V8Z 7X8, Canada. [9]Department of Computer Science, Aberystwyth University, Ceredigion, SY23 3DB, Wales, UK. [10]LabKey Software, 312 North 49th Street, Seattle, Washington, 98103, USA. [11]Indigo BioSystems, Inc., 111 Congressional Blvd, Suite 160, Carmel, Indiana, 46032, USA. [12]Department of Genetics, University of Pennsylvania, Center for Bioinformatics, 1415 Blockley Hall, 423 Guardian Drive, Philadelphia, Pennsylvania, 19104, USA. [13]Dana-Farber Cancer Institute, 44 Binney Street, Boston, Massachusetts, 02115, USA. [14]Lawrence Berkeley National Laboratory, University of California, 1 Cyclotron Road Mail Stop 977R225A, Berkeley, CA 94720, USA. [15]Institute for Translational Medicine and Therapeutics, University of Pennsylvania, 421 Curie Blvd., Philadelphia, Pennsylvania, 19104, USA. [16]These authors contributed equally to this work. Correspondence should be addressed to A.P. (angel@mail.med.upenn.edu).

Published online 5 October 2007; doi:10.1038/nbt1347

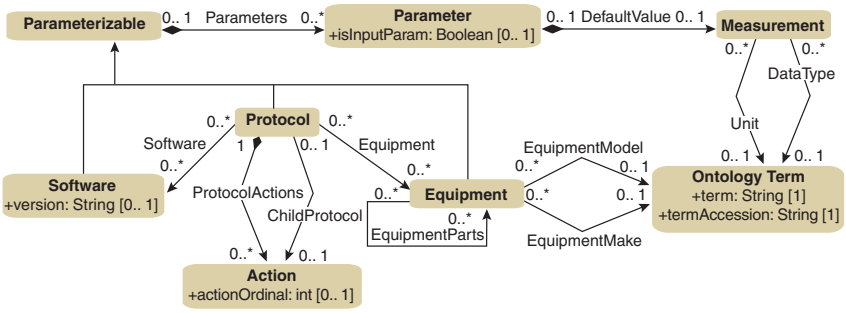


Figure 1 A UML diagram displaying a subset of the `Protocol` package. FuGE relies heavily on inheritance (such as the association between `Protocol` and `Parameterizable`), whereby classes inherit attributes and associations from the parent class. All classes have additional attributes inherited from more general parent classes (not shown) which allow a unique identifier, a name, descriptive text and various other properties to be provided.

PSI has also begun work on formats for protein-separation technologies, using the Proteomics Experiment Data Repository (PEDRo[9]) as a starting point. These standardization efforts address some of the concerns for publicly accessible formats for data and experimental annotation, but the independent nature of the standards groups caused common aspects of experimental protocols to be modeled using different terminology and levels of detail. The result is that semantically equivalent information was represented in syntactically incompatible ways across the standards, potentially complicating the publication process, the analysis and verification of studies that use multiple high-throughput technologies, and the integration of such data.

In response to the need for integration of the various technology types, independent attempts were made to merge MAGE and PEDRo into a single data model[10,11]. The conclusion from these efforts is that a comprehensive data model for all experimental types would be large and complex, hindering adoption by the technology-specific developer communities and vendors. On the other hand, these efforts also demonstrated that convergence of models in the areas shared between technologies, such as the biological source material, sample processing and the experimental variables, would yield significant benefits, both for data producers and for data consumers, if common aspects of an experimental activity could be recorded once (and not separately for each kind of technique used to study a sample). Furthermore, both data producers and consumers stand to benefit from the use of consistent styles of representation for the types of annotation that differ across techniques. The Functional Genomics Experiment model (FuGE) seeks to make the representation of data resulting from diverse experimental techniques more systematic and consistent by providing: (i) a format for representing laboratory workflows, (ii) a mechanism for supplementing existing data formats with additional metadata to describe their context within a laboratory workflow and their relationships to other data and (iii) a framework for building new data formats with a common structure for techniques that have specific requirements.

FuGE accomplishes these goals by focusing on the representation of the common aspects of experimental annotation and generally applicable information about the design of investigations. As such, FuGE provides a solid foundation for other technology-specific, life-science standards and data formats and is currently being used to develop formats for microarrays, proteomics, metabolomics and various other technologies.

In the next sections, we describe the methodology used to develop FuGE, the translation of the data model to other formats, aspects of the data model itself, and current development efforts based on provisional releases of FuGE. In this document, we designate any concept represented directly in the data model with a `fixed width font`.

RESULTS

In this section, we present some of the key concepts of the FuGE model, which consists of ten packages that have been placed in two categories: `Common` and `Bio` (**Box 1**). The FuGE specification is too large to cover in detail here; instead, we focus on how FuGE models the structure of an 'omics investigation, the experimental methods, the tracking of samples within an experimental workflow and the multidimensional data produced. Examples are presented from the

Box 1 Overview of the packages in FuGE

Common

`Audit`	Contacts, auditing and security settings for all objects.
`Description`	Additional annotations and free-text descriptions for all objects.
`Measurement`	Defines slots for providing atomic, Boolean, range and complex values with appropriate units, sourced from an ontology.
`Ontology`	A mechanism for referencing external ontologies or terms from a controlled vocabulary.
`Protocol`	A model of procedures, software, hardware and parameters. The package can define workflows by relating input and output materials and/or data to the protocols that act on them.
`Reference`	External bibliographic or database references that can be applied to many objects across the FuGE model.

Bio

`ConceptualMolecule`	Captures database entries of biological molecules such as DNA, RNA or amino acid sequences and provides an extension point for other molecule types, such as metabolites or lipids.
`Data`	Defines the dimensions of data and storage matrices, or references to external data formats.
`Investigation`	Defines an overview of the investigation structure by capturing the overall design and the experimental variables and by providing associations to related data.
`Material`	Models material types such as organisms, samples or solutions. Materials are characterized by ontology terms or by extension of the Material package.

Protocol, Material, Investigation and Data packages that illustrate the most important types of functionality. The Audit, Description and Reference packages are closely based on MAGE version 1, whereas the Investigation and ConceptualMolecule packages have evolved from MAGE to cover the wider context of functional genomics. The Protocol and Material packages reuse certain components from MAGE and from PEDRo but have been developed *de novo*. The Ontology and Data packages are newly created, using principles from related object-oriented proposals, as detailed in the complete specification (http://fuge.sourceforge.net/).

Basic functionality for all objects in FuGE is represented in the Common namespace. Every object can be annotated with audit information (tracking changes) and the desired security settings (users or groups that can access or modify objects). This level of control is important for larger organizations when, for example, regulatory requirements must be fulfilled. Furthermore, most objects in FuGE can be annotated with a unique identifier, a local name, a textual description and references to external database or bibliographic entries.

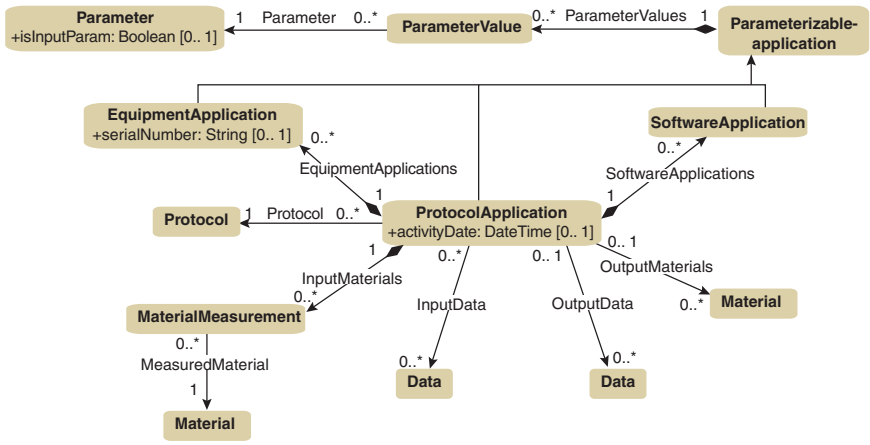


Figure 2 A UML diagram of ProtocolApplication in FuGE. ProtocolApplication, EquipmentApplication and SoftwareApplication can be used to supply runtime values (ParameterValue) for Parameters that were defined by the Protocol, Software or Equipment.

Representing biological workflows

All descriptions of an experimental workflow, from starting samples through to the final results, are encoded by the Protocol package. The package represents any method or procedure in an experiment, including standard operating procedures, the mechanism for running an instrument and the use of software for data processing.

A Protocol object can be associated with Software and Equipment, each of which can have a set of parameters with default values (**Fig. 1**). A Protocol consists of a set of Actions (or steps) that can be ordered. An Action can contain simple text describing an atomic step within a protocol, it can be associated with parameters or it can be a reference to a child Protocol. This means that a complex procedure can be represented by building a Protocol that references other Protocols in a nested structure. An example protocol is sample processing in proteomics. A single Protocol could be defined for the entire procedure, which has three Actions for the harvesting of material, protein extraction and protein solubilization. Each Action would contain a reference to a separate Protocol for each of the three steps.

A laboratory procedure is typically defined once (such as a method in a lab book or a standard operating procedure), but may be applied many times. FuGE represents this distinction by defining ProtocolApplication (**Fig. 2**). ProtocolApplication represents the running of a Protocol, allowing runtime parameter values to be supplied if they differ from the defaults. The separation of Protocol and ProtocolApplication is technically advantageous as it would be inefficient to redefine a complete protocol for every single deviation that occurs. For example, mass-spectrometry techniques use the same protocol definition for hundreds of runs, with only a small subset of the parameter values varied across them. ProtocolApplication also provides mechanisms for recording the operator and date of the procedure; both are variables that have been shown to be important when identifying and accounting for confounding factors in data analysis[12,13].

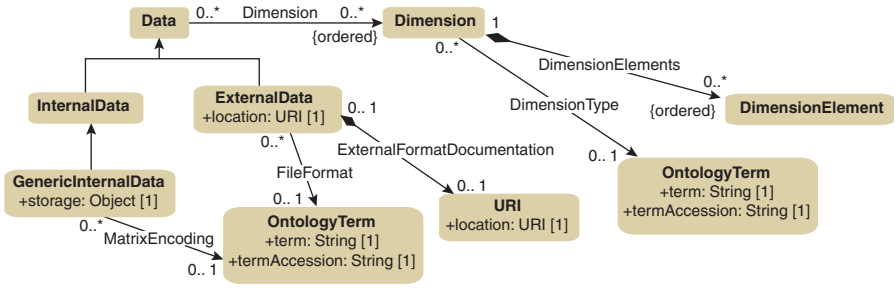
In addition to recording run-time parameter settings, a ProtocolApplication references the input and output materials and/or data that were acted upon. As such, it can be used to construct experimental workflows by tracking the identity of all samples and data files. FuGE supplies a placeholder for the description of all physical materials (e.g., samples, organisms, chemicals, solutions) represented by the Material class. A Material can be annotated with ontology terms to describe its type or the role it plays within an experimental workflow (such as sample, buffer or reagent). It is also anticipated that the Material class will be extended within technology-specific formats; possible examples include gels, antibodies, arrays, reporters and so on.

ProtocolApplication can also be used to describe a data-processing pipeline by virtue of its references to input and output Data objects, such as a series of data transformations, where the output of each step serves as input for the next. As such, a single robust mechanism can be used to demonstrate the provenance of a highly processed outcome (e.g., gene expression profiles) from the starting samples, through sample processing, raw data acquisition and data analyses.

Multidimensional data representations

A common aspect of high-throughput technologies is multidimensional data. Many technology types already have established data formats, some of which are open-source formats. These technology-specific formats tend to lack metadata structures to describe the context under which the data were produced. FuGE seeks to augment established formats with this type of metadata, thus providing a context for the data within a complete experiment. An example of this functionality is given by the Computational Proteomics Analysis System (CPAS)[14] project, as described below, which uses FuGE to integrate mass-spectrometry formats into a complete workflow description. In the Data package, referencing external data files is accomplished by the ExternalData class, which contains an attribute for referencing a file and a mechanism for referencing validation schema, descriptors or documentation on the external format. These attributes use standard URI notation to specify locations, such as Web addresses or local files (**Fig. 3**).

Alternatively, standards groups seeking to provide vendor-neutral data formats can encode data directly within FuGE using the data-matrix representation, specified by InternalData, Dimension and DimensionElement. The Dimension object describes an axis of the



Textual example
(microarray analysis measured across four doses of a drug)

Dimension1. Array features (Dependent variable dimension)
DimensionElements = Feature1; Feature2; Feature3... Feature 9500;
Dimension 2. Measurements (*Quantification* dimension)
DimensionElements = Signal (1); normalize value (2); P-value (3)
Dimension 3. Drug doses (independent variable dimension)
DimensionElements = 10 mg (1); 20 mg (2); 40 mg (3); 80 mg (4)

InternalData. Stores matrices of values (each data point accessed by a combination of three coordinates from each of the three dimensions).

Example. Feature 7, normalized value from the 40 mg assay would be at position 7:2:3 in the InternalData matrix.

Figure 3 The `Data` package enables data to be stored internally by a specification of dimensions, coordinates and matrices, or in an externally defined file format.

data matrix, which contains ordered instances of `DimensionElement` that describe the types of the coordinates in the axis. A simple example would be a tabular representation of gene expression, where one axis (`Dimension`) represents a gene list (a 9,500 feature array would have 9,500 `DimensionElement` instances for this `Dimension`), representing the dependent (responding) variable in the investigation. A second axis would represent the independent variable, such as time points within a time-course experiment, whereas a third axis represents the types of measurements derived from scanning the slide (e.g., signal, normalized value and *P* value). The `InternalData` object stores the data as a matrix of values, separated from the definition of the data dimensions. The set of coordinates of `DimensionElements` can be used to access individual values in the `InternalData` matrix. This structure for data storage and access is similar to the HDF5 specification for multidimensional scientific data (http://hdf.ncsa.uiuc.edu/HDF5/), which provides a representation that is highly efficient in terms of storage space and access speed.

Biological investigations

The `Investigation` package has been developed in consultation with cross-technology working groups[15] to capture the overall goal and design of the investigation, such as high-level description of the motivation for the experiments, and the experimental variables (**Fig. 4**). Repositories are frequently queried using this kind of metadata to retrieve data sets of interest; thus, it is important that such information is captured in a consistent manner.

The `Investigation` class captures the name and description of the entire investigation, with which ontology terms can be used to annotate the type of design employed; suitable terms from the MGED Ontology[16] include: "dose response design" or "genetic modification design." The package also models the important sources of material (single organisms, populations, tissue, cell cultures and so on), as determined by the investigator, for the purpose of providing a summary that can be queried. The `Investigation` class can also define a hypothesis, the conclusions or other important classification information as free text or as rich ontological structures.

`InvestigationComponent` represents a single functional genomics technique, allowing the user to specify experimental replicate design, the normalization strategy and quality control procedures, which are properties given prominence in the MIAME guidelines. `InvestigationComponent` can also define experimental design in relation to the technology (e.g., 'dye swap') through the use of ontology terms.

The principal comparators in an investigation (the manipulated or independent variables), such as dosage, genetic difference or environmental factor, are modeled by `Factor`. A `Factor` can, but need not, be shared across different instances of `InvestigationComponent`; for instance, certain technologies might be used to measure certain variables but not others. The value for a `Factor` is stored in `FactorValue` in conjunction with the `Measurement` class. Although FuGE does not include a specific 'Unit' class, this essential information can be provided via `OntologyTerm` references. In addition to providing the units for `FactorValue` measurements, ontologies can provide terms for nonnumeric `FactorValues`, such as cell line or sex (**Fig. 4**, `Factor` 1). There is also a mechanism for relating particular experimental variables to data of interest, via the `DataPartition` class. This will allow queries of the type "retrieve all data relating to the 10 mg drug dose." The `Factor`, `FactorValue` model is intended for capturing a summary description of the independent variables tested; the exact details of the study design and relationships between variables are represented in the `Protocol` and `Data` packages, allowing highly complex studies to be reported.

Building extensions on FuGE

FuGE can store general details about a protocol, samples and data but does not model specific properties of techniques or instruments, which is left to experts in those domains to define. There are two methods that can be used to define extensions:

Extending the object model with more specific attributes and associations that enforce the reporting of particular information. These modular formats based on FuGE can fill this role of enforcing constraints while remaining compatible with other FuGE-based formats.

Developing external ontologies that include specific controlled vocabulary terms and rules that govern their usage.

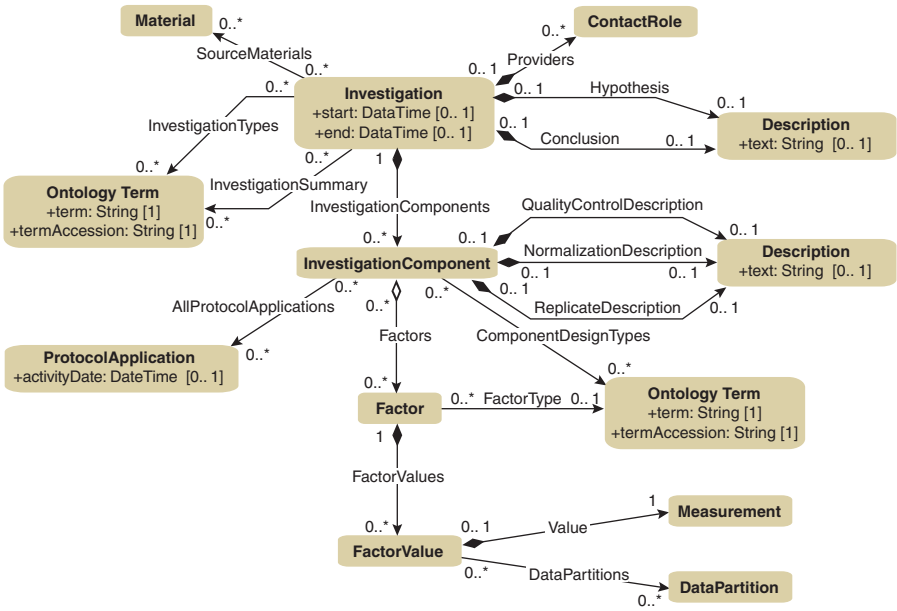
Several formats based on FUGE are being developed by PSI and MGED. To date, these formats have extended parts of the model (method 1) to capture specific details about the technology. However, ontologies are also being developed to capture parts of the model that do not have a fixed scope and may be extended incrementally over time. FuGE also relies on ontologies for enumerated lists of values, such as units. The Ontology of Biomedical Investigation (OBI), formerly called the Functional Genomics Investigation Ontology[17] (FuGO), is being developed in parallel to FuGE and will provide terminology for annotating data in a consistent manner (http://obi.sourceforge.net/).

As an example extension of FuGE, a model is under development to describe electrophoresis (GelML, http://www.psidev.info/index.php?q=wiki/Gel_electrophoresis), which is used in proteomics to separate complex mixtures of proteins in a polyacrylamide gel matrix. GelML

aims to support a proposal for the minimum reporting requirements for gel electrophoresis[18] and to serve as a format for exchanging gel electrophoresis data. For the purposes of this example, a two-dimensional gel electrophoresis protocol consists of the following steps: loading a sample onto a gel strip, performing electrophoresis in the first dimension, loading the strip onto a second gel and then performing electrophoresis in the second dimension. The example in **Figure 5** demonstrates how this complex procedure can be expressed by an extension of `Protocol` and `Action`.

The `Gel2DProtocol` class has four distinct steps, expressed by extensions of `Action`. `SampleLoadingAction` references a child protocol for capturing how the samples are loaded (`SampleLoadingProtocol`). The `Actions` for the first- and second-dimension separations have a reference to `ElectrophoresisProtocol` (which consists of a collection of parameters for voltages and timings, not shown). Finally, `InterDimensionAction` represents the stages that occur between the first- and second-dimension separations, and references the FuGE `GenericProtocol` class that captures any procedure that has no explicit model.

The advantages of using FuGE in this way are as follows. FuGE provides structure for fitting extensions into the larger context of a complete workflow, such as relating protocols to samples and data files, thus facilitating format design by allowing developers to focus on what to capture rather than on how to structure the model. In addition, extended objects gain the rich functionality of FuGE for auditing, controlling security settings and having a consistent identification system. Furthermore, by extending from specific FuGE classes, models of different techniques will share significant structural similarities, facilitating future data-integration efforts and reducing the learning time for new models. Modular formats built on FuGE will also allow developers to focus on a single representation (namely, UML development) from which the XML Schema, relational database definition and software components can be generated automatically. This automation should both simplify development and the mapping work required to maintain parallel implementations.



Textual example

Investigation. Transcriptome/Proteome analysis of WT vs Gene KO mice under various drug regimes
Providers. Prof J Smith, University of Manchester, (ContactRole = Principle Investigator)
InvestigationTypes. "dose response design"; "genetic modification design"
SourceMaterials. Mus musculus; drug ID 35f57.1

InvestigationComponent 1 (IC1). Microarray assay
Reference to ProtocolApplications for sample prep; hybridization; data analysis
Reference to Factor 1 and Factor 2

InvestigationComponent 2 (IC2). Proteome assay by LC-MS
Reference to ProtocolApplications for sample prep; separation; mass spectrometry; data anaysis
Reference to Factor 1 and Factor 2

Factor 1: FactorCategory = Genotype
FactorValue 1 = Wild-type
FactorValue 2 = Gene XYZ Knockout

Factor 2: FactoryCategory = Drug dose
FactorValue 1 = 10 mg
FactorValue 2 = 20 mg
FactorValue 3 = 40 mg

Figure 4 The `Investigation` package and a textual example instance.

DISCUSSION

In the past, functional genomics standards development focused on single technologies or solutions within a single community. In contrast, FuGE has received input from a diverse set of standards bodies and organizations with an interest in data sharing and, as such, represents a major cross-community collaboration. Several groups are currently using or evaluating FuGE as the basis for their respective data models.

The Fred Hutchinson Cancer Research Center has developed CPAS, which uses a file format for archiving based on an early release of FuGE. The archive file stores information describing the experiment, including materials, protocols and types of data involved. Files produced from assays (e.g., raw mass-spectrometry data in mzXML format[19]) and the results of data-analysis procedures (e.g., a pepXML file from a search engine result[20]) are packaged together with supporting metadata, and the collection can be submitted to CPAS or other compatible systems, such as the ProteusLIMS commercial laboratory information management system (http://www.genologics.com/). The PRIDE[21] public data repository also plans to support PSI-endorsed formats based on FuGE.

FuGE is currently being used by MGED to develop MAGE version 2, with the aim of reducing the complexity of the format as a result of feedback from MAGE version 1, and to include additional

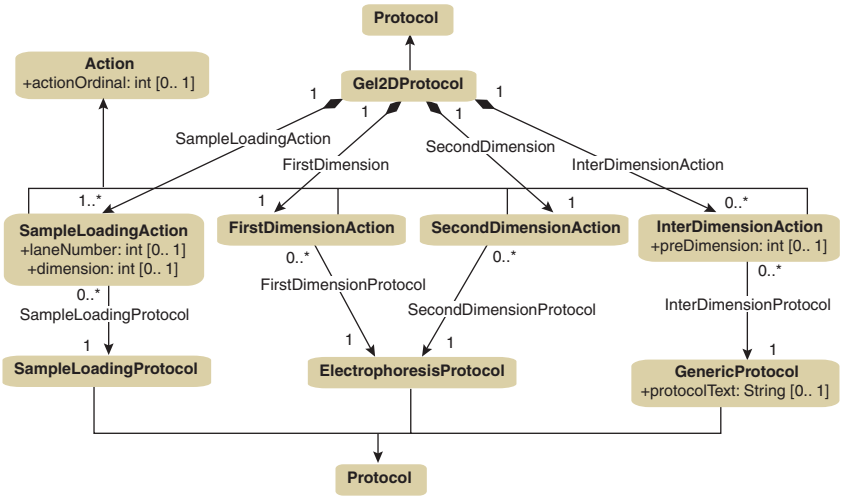


Figure 5 A UML diagram of an extension to FuGE for capturing protocols for two-dimensional gel electrophoresis from GelML. The complete UML diagrams for `SampleLoadingProtocol`, `ElectrophoresisProtocol` and `GenericProtocol` are not shown.

experimental approaches, including SNP arrays, protein arrays (developed in collaboration with PSI) and a number of data analyses. PSI is also developing formats based on FuGE for gel electrophoresis, sample processing and reporting of mass-spectral analyses[22]. These are expected to be released within the next year. To date, FuGE has not been significantly deployed to manage data resulting from clinical trials. However, FuGE can describe assays stemming from clinical samples and study-design information, and thus complements existing mechanisms for reporting clinical trials. The Metabolomics Standards Initiative (http://msi-workgroups.sourceforge.net/) has recently been formed, and the data-exchange working group is currently evaluating FuGE. The group is likely to recommend its adoption for capturing investigational design and sample processing and as a basis for future formats involving metabolite separation and analysis. FuGE is also being evaluated by groups developing formats for RNAi, flow cytometry, cellular assays and immunohistochemistry.

FuGE is not formally owned by any single standards organization. Instead, it constitutes a stable, independent artifact that will be formalized in a standardization process. The model should be extended according to a set of guidelines, a draft of which appears on the Web site, thus encouraging new formats to share a consistent structure. Formats that extend FuGE without following the guidelines may not be able to use the templates for producing XML Schema, relational database definition or software platforms. We believe this kind of open process will avoid the need for the formation of large cross-technology standards groups in which it is difficult to make rapid progress in response to technological developments.

Adoption of FuGE by the transcriptomics, proteomics and metabolomics communities will result in a common format for representation of experimental descriptors that are independent of a particular technique. Researchers can describe the overall investigation, the source of material and experimental techniques using the core FuGE model, potentially allowing for the *ad hoc* assembly of studies that cross technological boundaries. The ability to provide rich annotation using both general and domain-specific ontologies, as developed by OBI, will facilitate linkage of cross-platform and organization investigations.

As research groups move towards systems-biology approaches that cross technologies, the type of convergence of data formats that FuGE promotes will be essential to ease the burden of the capture, dissemination and publication of annotated data sets. Widespread adoption of FuGE will also aid evaluation and comparison of those published results by improving the uniformity of the annotation of data deposited in public repositories. Finally, convergence of data formats will promote the development of software applications that span technology types, facilitating the peer-review process and fostering reanalysis of data and novel methods development, such as modeling of complex biological processes.

METHODS

'Use cases' gathered from a broad spectrum of the functional genomics research communities were used to develop the FuGE data model. As such, it contains representations of the concepts common to most functional genomics experiments. The initial development stages involved analysis of MAGE, the removal of components specific to microarrays, and redesign of components to fit the wider set of use cases. Subsequently, feedback obtained during the development of external formats or projects based on the FuGE milestones was communicated to the FuGE developers and incorporated in later releases.

Several stable, provisional versions of FuGE, termed milestones, have been publicly released to allow developers to work on extensions of FuGE or implement software using FuGE. Each milestone consists of the UML model, the XML Schema produced from the model, and documentation. A formal standardization process has been followed, including a significant period for public comment on the specifications, which has resulted in an official stable release (FuGE version 1.0).

The FuGE project relies on several freely available tools for development. The UML model is developed using the MagicDraw CASE tool (http://www.magicdraw.com/) and is subsequently translated to other formats using AndroMDA (http://www.andromda.org/), an open source project that can produce various types of documents from the UML model using a set of document templates. We have specifically tailored AndroMDA templates for production of an XML Schema, a relational database definition and Java software components. Templates for supporting other software platforms, such as Perl or C++, can be developed in the future. The use of publicly available tools for model design and format generation provides a common platform for developing community-specific extensions and avoids excluding particular groups based on software costs.

The FuGE UML specification is restricted to class diagrams, where classes use simple inheritance (only one parent class) and define attributes and associations to other classes but no procedures (methods). The restriction on inheritance greatly simplifies the mapping to other platforms, such as the XML Schema and relational database schema, and there are relatively few instances where multiple inheritance would convey any advantage. The **Supplementary Note** contains a brief tutorial illustrating the subset of UML syntax used in FuGE.

Note: Supplementary information is available on the Nature Biotechnology website.

ACKNOWLEDGMENTS

We would like to thank MGED and PSI for their support of the FuGE project. We would also like to thank FuGO members, RSBI, the CPAS group at the Fred Hutchinson Cancer Research Center and Genologics for their input into the model, testing and supplying use cases. Work on FuGE in Manchester has been supported by a grant from the BBSRC to S.G.O., N.W.P., S.J.H. and Andy Brass. MGED is funded by the National Institutes of Health grant 1P41HG003619, which has provided financial support for development meetings. The FuGE initiative is endorsed by Oliver Fiehn, chair of the Metabolomics Standards Initiative oversight committee.

Published online at http://www.nature.com/naturebiotechnology/
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/

1. Castrillo, J.I. & Oliver, S.G. Yeast as a touchstone in post-genomic research: strategies for integrative analysis in functional genomics. *J. Biochem. Mol. Biol.* **37**, 93–106 (2004).
2. Nie, L., Wu, G. & Zhang, W. Correlation between mRNA and protein abundance in *Desulfovibrio vulgaris:* a multiple regression to identify sources of variations. *Biochem. Biophys. Res. Commun.* **339**, 603–610 (2006).
3. Pir, P. *et al.* Integrative investigation of metabolic and transcriptomic data. *BMC Bioinformatics* **7**, 203 (2006).
4. Greenbaum, D., Colangelo, C., Williams, K. & Gerstein, M. Comparing protein abundance and mRNA expression levels on a genomic scale. *Genome Biol.* **4**, 117 (2003).
5. Brazma, A. *et al.* Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat. Genet.* **29**, 365–371 (2001).
6. Spellman, P.T. *et al.* Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.* **23**, RESEARCH0046 (2002).
7. Object Management Group. Gene Expression Specification. <http://www.omg.org/docs/formal/03–02–03.pdf> (2003).
8. Hermjakob, H. *et al.* The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data. *Nat. Biotechnol.* **22**, 177–183 (2004).
9. Taylor, C.F. *et al.* A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nat. Biotechnol.* **21**, 247–254 (2003).
10. Jones, A., Hunt, E., Wastling, J.M., Pizarro, A. & Stoeckert, C.J., Jr. An object model and database for functional genomics. *Bioinformatics* **20**, 1583–1590 (2004).
11. Xirasagar, S. *et al.* CEBS Object Model for Systems Biology Data, CEBS MAGE SysBio-OM. *Bioinformatics* **20**, 2004–2015 (2004).
12. Irizarry, R.A. *et al.* Multiple-laboratory comparison of microarray platforms. *Nat. Methods* **2**, 345–350 (2005).
13. Novak, J.P., Sladek, R. & Hudson, T.J. Characterization of variability in large-scale gene expression data: implications for study design. *Genomics* **79**, 104–113 (2002).
14. Rauch, A. *et al.* Computational Proteomics Analysis System (CPAS): an extensible, open-source analytic system for evaluating and publishing proteomic data and high throughput biological experiments. *J. Proteome Res.* **5**, 112–121 (2006).
15. Sansone, S.A. *et al.* A strategy capitalizing on synergies: the reporting structure for biological investigation (RSBI) working group. *OMICS* **10**, 164–171 (2006).
16. Whetzel, P.L. *et al.* The MGED Ontology: a resource for semantics-based description of microarray experiments. *Bioinformatics* **22**, 866–873 (2006).
17. Whetzel, P.L. *et al.* Development of FuGO–an ontology for functional genomics experiments. *OMICS* **10**, 199–204 (2006).
18. Gibson, F. *et al.* MIAPE Gel Electrophoresis. Community consultation. *Nat Biotechnol.* <http://www.nature.com/nbt/consult/index.html> (2007).
19. Pedrioli, P.G. *et al.* A common open representation of mass spectrometry data and its application to proteomics research. *Nat. Biotechnol.* **22**, 1459–1466 (2004).
20. Keller, A., Eng, J., Zhang, N., Li, X.-J. & Aebersold, R. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol. Syst. Biol.* **1**, 2005.0017 (2005).
21. Jones, P. *et al.* PRIDE: a public repository of protein and peptide identifications for the proteomics community. *Nucleic Acids Res.* **34**, D659–D663 (2006).
22. Orchard, S. *et al.* Proteomics and beyond a report on the 3rd Annual Spring Workshop of the HUPO-PSI 21–23 April 2006, San Francisco, CA, USA. *Proteomics* **6**, 4439–4443 (2006).