

Towards dynamic database infrastructures for mouse genetics

Morris A. Swertz, Damian Smedley, Katy Wolstencroft, Rudi Alberts, Michael Zouberakis, Vassilis Aidinis, Klaus Schughart, Paul N. Schofield, Ritsert C. Jansen, members of the CASIMIR Consortium.

Abstract—A growing array of biotechnologies is being used to study the genetics of complex biomolecular traits in laboratory mice as models for human disease. Combined analysis of these datasets provides much of the power of the approach of functional genomics but this depends on the ability of databases to exchange data with each other and with analytical software. In the light of these challenges the European Commission has funded a coordination action, CASIMIR, to make recommendations on how this need might be fulfilled. We here report on two pilot projects and distill preliminary recommendations.

I. INTRODUCTION

GENETIC study of complex biomolecular traits in laboratory mice involves perturbing biological networks through genetic variation, observing the effects at one or more biomolecular level(s), finding regulatory interactions, and finally reconstructing molecular networks. Fig. 1 illustrates the challenging data management, (pre)processing and integration required:

This work was supported in part by EU-CASIMIR under Framework Programme 6 of the European Commission (LSHG-CT-2006-037811). <http://www.casimir.org.uk>.

M. A. Swertz is with the University Medical Center Groningen, Department of Genetics, P.O. Box 30001, NL-9700 RB, Groningen, The Netherlands, and with the University of Groningen, Groningen Bioinformatics Centre, P.O. Box 14, NL-9750 AA Haren, The Netherlands. (phone: +31 50 363 8091; fax: +31 50 363 7976; e-mail: m.a.swertz@rug.nl)

D. Smedley is with the European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom. (e-mail: damian@ebi.ac.uk)

K. Wolstencroft is with the University of Manchester, School of computer science, Kilburn Building, Oxford Road, Manchester M13 9PL, United Kingdom. (e-mail: katherine.wolstencroft@manchester.ac.uk)

R. Alberts is with the Helmholtz Centre for Infection Research, Dept. Experimental Mouse Genetics, Inhoffenstraße 7, D-38124 Braunschweig, Germany. (e-mail: rudi.alberts@helmholtz-hzi.de)

M. Zouberakis is with the B.S.R.C. Alexander Fleming, 34 Fleming Street, 16672, Vari, Athens, Greece. (e-mail: zouberakis@fleming.gr)

V. Aidinis is with the B.S.R.C. Alexander Fleming, 34 Fleming Street, 16672, Vari, Athens, Greece. (e-mail: aidinis@fleming.gr)

K. Schughart is with the Helmholtz Centre for Infection Research, Dept. Experimental Mouse Genetics, Inhoffenstraße 7, D-38124 Braunschweig, Germany. (e-mail: klaus.schughart@helmholtz-hzi.de)

P. N. Schofield is with the University of Cambridge, Department of Physiology, Development and Neuroscience, Downing Street, Cambridge CB2 3DY, UK. (e-mail: ps@mole.bio.cam.ac.uk)

Ritsert C. Jansen is with the University of Groningen, Groningen Bioinformatics Centre, P.O. Box 14, NL-9750 AA Haren, The Netherlands. (e-mail: r.c.jansen@rug.nl)

CASIMIR Partners: University of Cambridge, Cambridge, UK; MRC Harwell, Oxfordshire, UK; MRC, Edinburgh, UK; EBI, Hinxton, UK; EMBL, Monterotondo, Italy; BSRC Fleming, Vari, Greece; GSF National Research Center for Environment and Health, Neuherberg, Germany; Helmholtz-Zentrum fuer Infektionsforschung GmbH, Braunschweig, Germany; CNR-Consiglio Nazionale delle Ricerche-Istituto di Biologia Cellulare, Monterotondo, Italy; Geneservice Limited, Cambridge, UK

Genetical genomics experiments [1] involve large scale molecular measurements on a reference panel of hundreds of genetically different mouse strains produced by particular (in)breeding strategies. Although this type of experiments may roughly follow a common protocol, they differ in their specifics at several steps:

Each individual is typed with molecular marker technologies (markers in *1a*, SNPs in *1b*) to generate 10,000-100,000 pieces of information about their genetic make-up (genotypes). Each individual is also profiled using gene expression technologies (Qiagen-Operon microarrays in *1a*) or mass spectrometry technologies (LC-MS in *1b*) to get 100,000 pieces of information about which of the 20,000-30,000 genes are ‘switched on’ (gene expression) in a given tissue or cell population, or which genes give rise to a in protein and/or are associated with the occurrence of metabolite molecules (visible as mass peaks).

The data analysis requires data exchange with various (pre)processing algorithms for gene expression (*1a*) or mass spectrometry (*1b*) data that generate output that often exceeds input in size and complexity. Interpretation of these results requires integration of gene/locus (*1a*) or enzyme/protein (*1b*) targets with highly dispersed background information from private and public repositories on, e.g., phenotypes, genomic context and pathways. All data, annotations and protocols have to be well managed to be able to track and trace experiments and, if needed, to re-do and re-interpret analyses.

After two decades of (post-)genomics research, one would hope that database infrastructures could be used ‘off-the-shelf’ to support each particular type of experiments. Collaborations in CASIMIR showed that this is not yet the case. Some genotype and phenotype databases and computational tools have been integrated in MGI [2], GeneNetwork.org [3], and dbGaP [4] but these software infrastructures are designed as public repositories and not to support particular experimental workflows. Some software components for (pre-)processing [5]-[11] and for integration of background information [12]-[14] are available but assembly into seamless software infrastructure requires time-consuming changes in hand-written software code. In practice, this leaves biologists with the challenging task to learn the interfaces of different tools, reformat data files by hand to make them fit, copy-paste data and identifiers from website to website, and merge all partial-results into a bunch of Excel or Word documents by hand. This laborious and error-prone process has to be repeated for each gene,

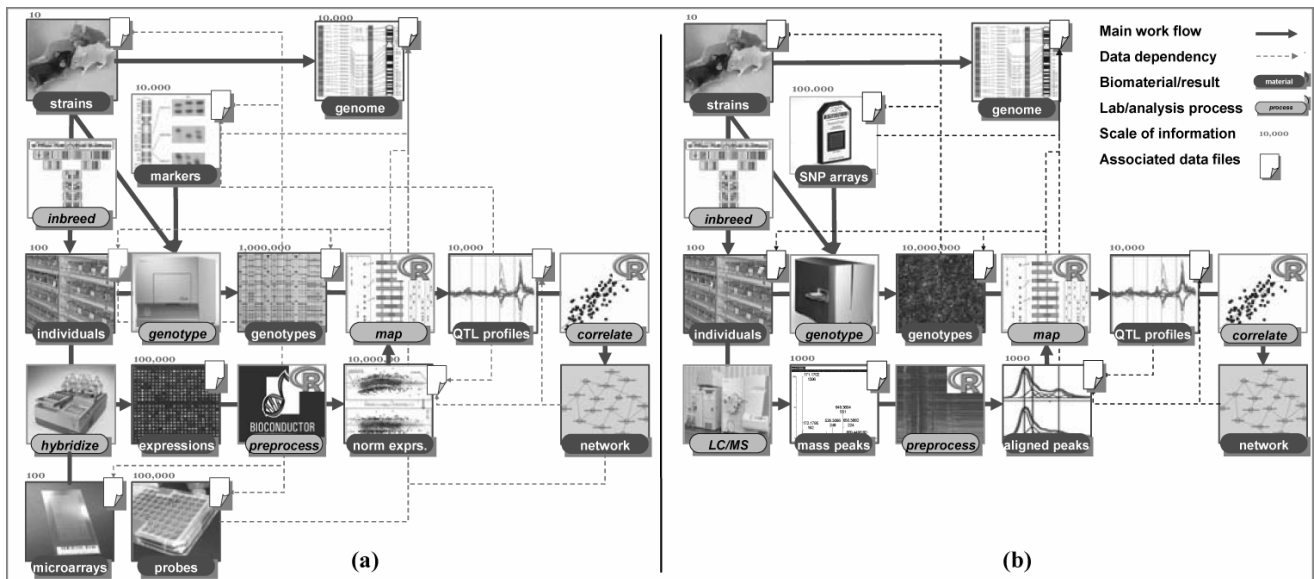


Fig. 1. Two variants of mouse genetics experiments are shown: (a) involves two-color microarrays and (b) mass spectrometry measurements. Database infrastructures supporting genetics experiments need to be dynamic to accommodate the variation that follows new experimental designs and methods.

and again when data sources are updated, and does not scale up to the hundreds or thousand of genes typically found in whole genome experiments.

Clearly, isolated development of more local infrastructures from scratch is not a sustainable option as it will exacerbate the problem with more incompatible software, more duplicated efforts and greatly reduced lifespan of the software. Instead, flexible mechanisms are desired to enable reuse, extension and foremost integration of mouse database resources. Based on two pilot studies that resulted from the first CASIMIR co-ordination meetings held in Corfu and Rome in 2007, we here present preliminary recommendations on alternative software methods, models and tools to develop the dynamic database infrastructures needed.

II. GENERATING SUITABLE EXPERIMENTAL DATABASES

The first pilot involves the generation of an easy-to-extend and -integrate database infrastructure, taking genetical genomics experiments as example. The ideal database infrastructure has a minimal data model and flat file exchange format that closely resembles biologists practice, a graphical user interface (GUI) to easily submit and retrieve data, and simple application programming interfaces (API) for bioinformatician to easily integrate analysis tools and exchange data with related databases. Most importantly, the infrastructure should be easily modified into a new database variant, e.g. when new biomolecular technologies are introduced (e.g. Orbitrap mass spectrometry), when improved or new statistical protocols for (pre)processing data are developed (e.g. RMA to replace the MAS5.0 algorithm for normalization), or when new resources with background information come available (e.g. Europhenome database).

Currently, development of new databases (or adaptation of existing ones) to suit new types of experiments requires much programming effort and expertise. Our recent perspective paper [15] outlined an alternative ‘model-driven’ software engineering strategy that is adopted by several recent bioinformatics projects to generate such software more efficiently. Fig. 2 demonstrates in a simplified example how this strategy works in the pilot:

A relatively simple file is created by hand to ‘model’ what particular experiment database is needed: a minimal ‘domain specific’ programming language (DSL) is used to efficiently describe the organization of experimental data entities such as biomaterials, protocols, and measurements and how these data are to be shown on the screen. The translation of these biological features from DSL file into the many program files needed for a complete database software is automated in the MOLGENIS software generator [15]-[17]. From a DSL file, the generator automatically creates all the programmatic code that needed to be written by hand before, including (i) an SQL file with all necessary programming statements for setting up a database, (ii) several application programming interfaces (API) in that allow bioinformaticians to connect the database to their processing tools via R statistics [18], Java, ‘REST’ hyperlinks or SOAP and (iii) a graphical user interface (GUI) by which users can submit and retrieve data via a web browser, optionally using (iv) a simple tab-delimited file format for exchange of full experiment data. A new variant of database software is quickly created by just extending the textual description in

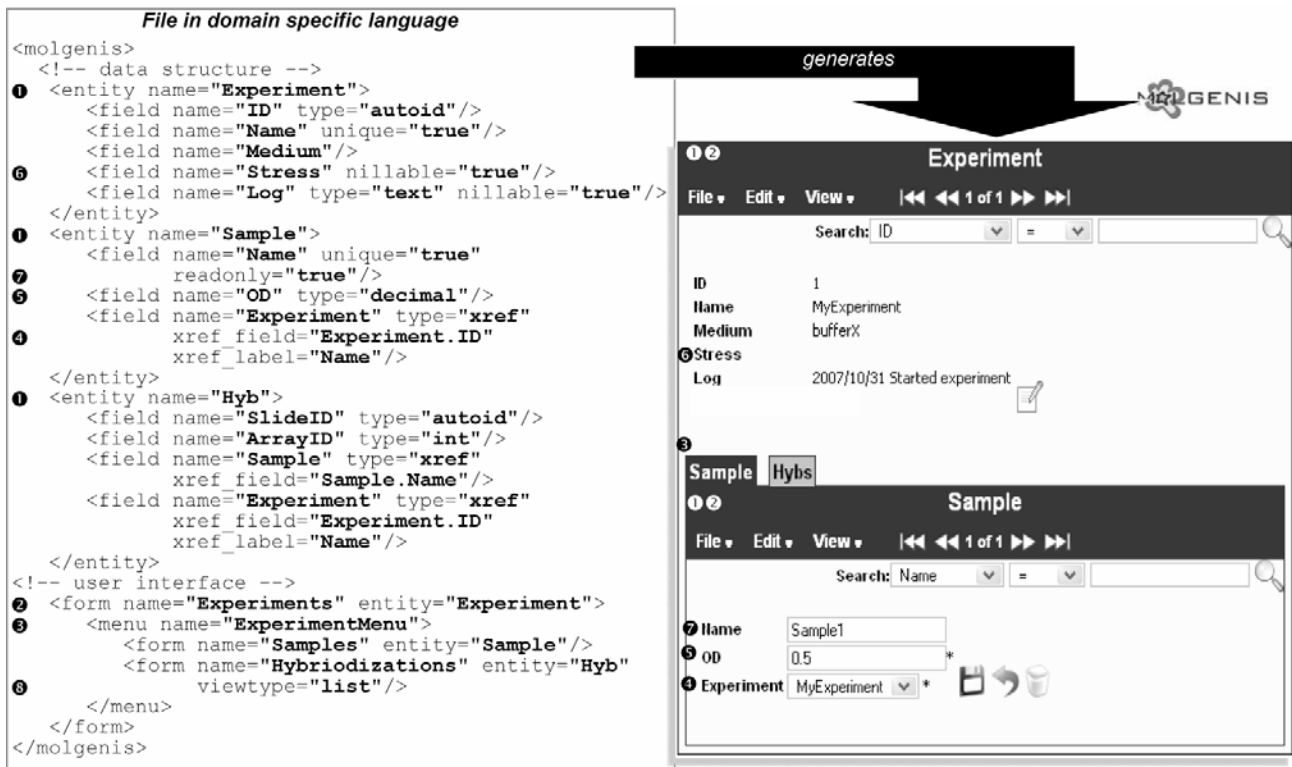


Fig. 2. Model-driven generation of biological databases using the MOLGENIS tool. Detailed software needs for an experiment are compactly modeled in a domain specific language; an simple example for microarray experiments is shown (DSL model file, left). The MOLGENIS generator reads the DSL file and, at the push of a button, automatically produces the custom software infrastructure described (right). The DSL model describes three data *entities* ❶ Experiment, Sample and Hybridization are described; the entity Sample has six *fields*, including ID, Phenotype and Chow. The DSL model also describes one user interface *form* ❷ to manage Experiments, with a sub *menu* ❸, consisting of two *child forms* for Samples and Hybridizations. These child forms are automatically linked to the parent form based on cross references, e.g. the field 'Experiment' of 'Sample' references to the 'ID' of an 'Experiment' ❹. Use of default settings keeps the DSL file short: each field is default of type 'string' (a variable character string of length 255) unless otherwise specified to e.g. 'decimal' ❺; each field has to be set to a value by the researcher unless specified to be nillable ❻; each field can be edited (updated) unless specified to be read only ❼; and each entity is viewed one-record-per-screen unless specified as list ❽ (not shown). Note: the example data in the screenshot were added post-generation.

DSL with some new data entities for, e.g., a new protocol and then rerunning the generator.

This model-driven strategy promises the generation of a whole 'family' of mouse genetics database variants with each family member accommodating a particular type of experiments such as Illumina SNP arrays, Affymetrix expression arrays, and Orbitrap proteomics mass spectrum measurements. The approach has several more advantages: Researchers can in a DSL file much better oversee what can, and cannot, be standardized between experiments as compared to overseeing many differences in software code; Bioinformaticians don't need to reinvent software engineering 'wheels' because hardcore technical challenges that are common in the development of such software are encoded in the software generator; and the generated software components, and data processed with them, can be more easily reused and integrated by other laboratories because their standardized production process

Currently, the pilot is being developed into a more complete database for genetical genomics [19] including a

catalog of biotechnology specific extensions/variants at <http://gbic.biol.rug.nl/dbgg>.

III. GENERATING INTEGRATIVE WORKFLOWS

The second pilot aims to automate integration of experimental results with background information dispersed over private and public databases. *Fig. 3a* sketches the challenges when, for example, retrieving background information on a list of 'candidate' genes for (a) allelic phenotypes and strain specific genotypes, (b) the genomic context of the particular gene locations, and (c) pathways that these genes may be involved in. Automation of data retrieval from different resources such as (a) Mouse Phenome Database (MPD) [20], [21], (b) Ensembl [22], [23] and (c) Kyoto Encyclopedia of Genes and Genomes (KEGG) [24]. [25] is not straightforward because these databases cannot directly 'talk to each other': programs can talk with MPD via flat file downloads, with Ensembl via its own Perl protocol, and with KEGG via a particular flavor of web services. This is even an optimistic scenario: most current biological databases are still primarily built for human users

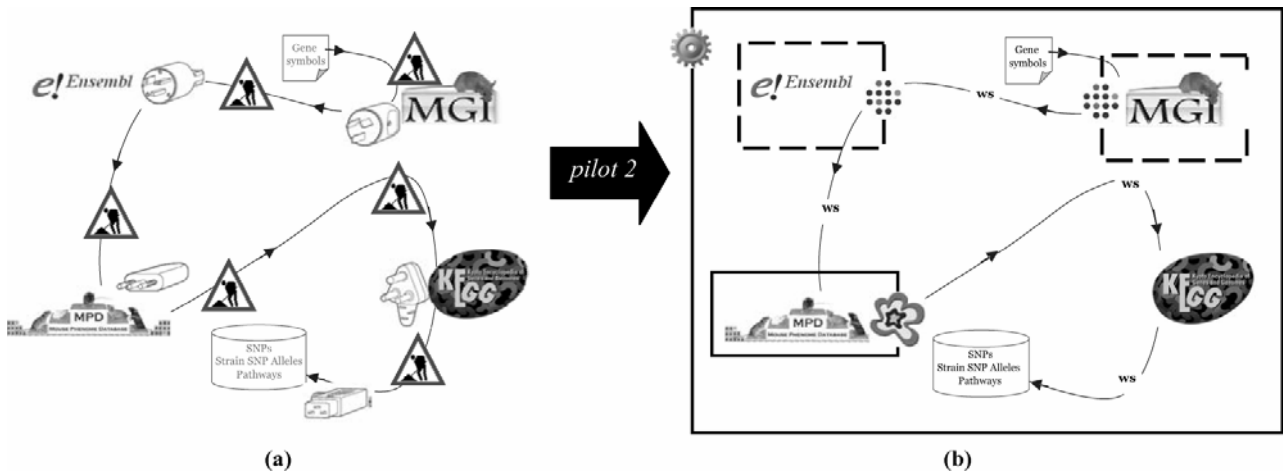


Fig. 3. Creation of integrative workflows. Until now, bioinformaticians need to put in a lot of work Δ to connect data from different and distributed biological data sources (left panel a.). This is due to differences or non-availability of programmatic access methods ☺ , ☹ , i.e. differences in languages and technical protocols. Use of technically standard wrappers, in this case using web services, makes it possible for sources to programmatically ‘talk to each other’ (right panel b.). The data can therefore be imported into a standardized query tool (in BioMART, dotted box). Alternatively, generators can be used to generate standard ‘wrappers’ for these sources (in MOLGENIS, boxes). KEGG already spoke web services and needed no wrapping. A workflow tool can be used to model the integrative workflow (in Taverna, b.). Wrapping the sources removes technical barriers so bioinformatician can focus on the important task: create computational protocols that (automatically) integrate data so it makes sense biologically. The working workflow can be downloaded from <http://www.myexperiment.org/workflows/126>.

and have no or limited support for programmatic access. Moreover, if there is support for programmatic access then the programmatic interfaces use heterogeneous technology and semantics.

Stein, in his seminal commentary [26] defined what is needed to create a ‘bioinformatics data nation’: data sources need to provide commonly accepted data formats, access methods, and a directory service that allows bioinformaticians/scripts to find them. *Fig. 3b* sketches the pilot solution [27] to enable such computational interplay build on the de facto standard integration syntax ‘web services’. Example of such technology is the SOAP protocol that is based on the simple idea of sending XML formatted text messages over computer networks, most notably the Internet protocols HTTP/HTTPS. Unfortunately, (SOAP or other) web services are not yet widely supported by data providers as their creation requires much additional implementation effort which is often too much to ask from smaller organizations. Therefore we used MOLGENIS and BioMART software tools to make existing tools MPD and Ensembl also ‘talk’ the web service language: BioMART [28], [29] is a standardized data warehouse where data providers can import their data into whilst MOLGENIS [15]-[17] generates software wrappers around a database such that data can be queried in their original structure. With web services in place, the Taverna [30], [31] software tool can be used to glue these resources together in an integrative workflow.

Note that in this pilot scenario the underlying data sources remain autonomous components which only minimally cooperate to share their specific functionality by providing a standard syntax (web services) building on standard

software tools. Next to this technical standardization, no structural or semantic standardization is assumed, instead the generic features of data are modeled and some kind of query-based logic is used for their API abstractions. This loosely-coupled approach is preferred over large scale standardization (e.g. in a data warehouse) as the domain expertise at each centre can be used to configure how and what data is presented to the researchers to address a particular research question. A drawback of this flexibility is that a lot of data conversion ‘shims’ where needed to overcome structural and semantic heterogeneity between the elements of the workflow. For example, Ensembl reports genomic location per single base pair while MPD reports per million base pairs (megabases); in the workflow a conversion is needed to allow data flow between them. Obviously, the need for such shims would be greatly reduced if data sources would standardize their data representations for ‘common’ data types. The challenge will be to standardize without sacrificing the qualities that makes a particular data source unique.

The DSL model of the workflow from *Fig. 2* is available to view and download from myExperiment (<http://www.myexperiment.org/workflows/126>) and can be run from within Taverna with the File->‘Open workflow location’ option using the same URL.

IV. PILOT IMPLEMENTATION PROCEDURE

Below we describe the technologies we used to implement the two pilot systems and provide a short overview of related resources.

A. MOLGENIS

From a model described in domain specific language (DSL),

MOLGENIS [15]-[17] can generate a database software infrastructure, including graphical front-end for human access as well as programmatic front-ends in R, Java and Web services for programmatic access. The database software can be generated *de novo* (as in Fig. 2) but can also be generated as wrappers around existing databases (as in Fig. 3). For example, to generate the “MOLGENIS for MPD” we first downloaded delimited text data files from [32]. A DSL file with the basic model of all MPD data entities was derived from the headers of the downloaded data. These descriptions were further detailed by hand to add, for example, proper cross references between SNPs and Strains. Finally, we fed this DSL file to the generator to produce the working software, see [33]. The MOLGENIS generator is open-source and available at <http://www.molgenis.org>.

B. BioMART

BioMart [28], [29] is a standardized, query optimized data warehouse. The software comes with a range of query interfaces including an 'out of the box' website that can be installed, configured and customized according to requirements as well as as a Perl API and Mart Services (BioMart's own version of web services). It is also integrated into several external software packages such as BioConductor [34] and Taverna. Several large biological datasets in the public domain have already been uploaded into BioMart, including dbSNP, Ensembl genomics, and PRIDE proteomics data which can be queried directly. New BioMARTs can be created using the MartBuilder tool to automatically transform a relational database structure into the generic BioMart schema. The BioMART data warehouse is open-source and available at <http://www.biomart.org>.

C. Taverna

Taverna [30], [31] is an environment for the design and execution of workflows that combine Web Services, BioMart queries, R-statistical analyses and/or BioMoby services, to name a few. Connecting to distributed data sources eliminates the necessity for downloading and maintaining local copies of data but combining distributed and heterogeneous services is a complex procedure. The workflow model is a record of such integration procedure describing what data sources have been linked and what 'shims' have been added for data conversion between them. New steps can be added to the protocol by connecting to more services, shown as 'processors' in Taverna's GUI. New services can be added to Taverna's processor catalog, e.g., to add the MOLGENIS MPD services we needed to right-click 'Available processors' and then click 'Add new WSDL scavenger' to add the services from the WSDL file available on [33]. The Taverna workflow workbench is open-source and available at <http://taverna.sourceforge.net>.

D. Related work

Table 1 in [15] lists several more model-driven tools to

generate biological software infrastructures to search, store, exchange and edit biological data (MOLGENIS [16], CCPN [35], caCORE [36], and Pedro [37]); share, and connect to, independently developed analysis components (BioMOBY [38], GALAXY [39] and PISE [40]); link those components together in processing workflows (Taverna [31]); and provide biologist-friendly user interfaces therefore (MOLGENIS, GALAXY and PISE). Each system has their own DSL which can either be textual, e.g., MOLGENIS has a XML-based textual language with keywords to define data entities and user interface screens (see Fig. 2), but also graphical, e.g., Taverna has a graphical language with boxes denoting processing components and the arrows denoting data flow between them (see Fig. 4).

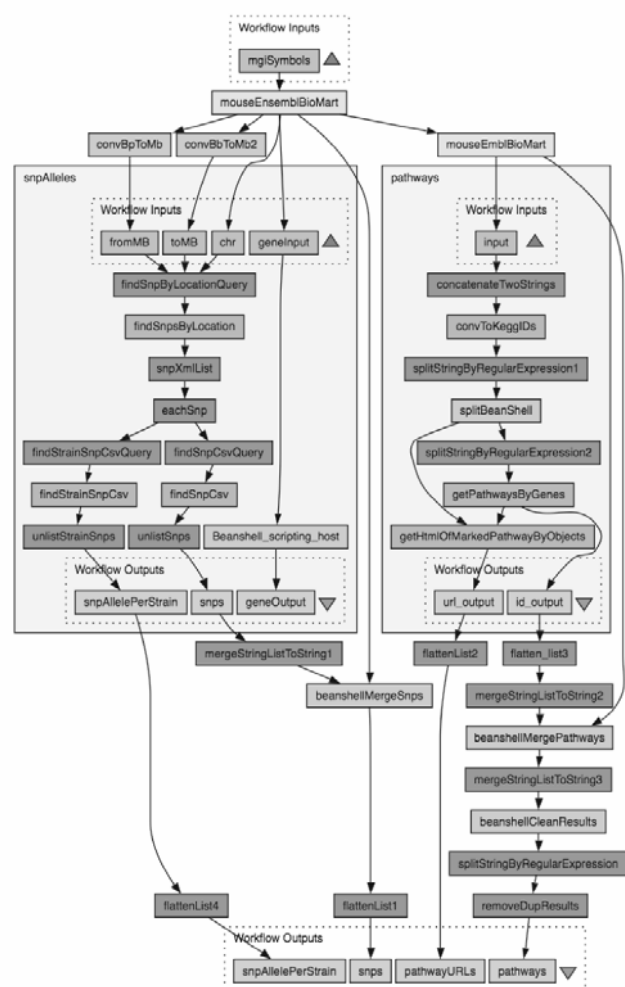


Fig. 4. Model of the integrative workflow pilot described using Taverna's domain specific language (DSL).

The key to the success of a model-driven tool/domain specific language is the higher level of abstraction as compared to a 'general' programming language. This is made possible by limiting the scope of the 'family' of software that can be produced. If the members of the software family vary too widely then the DSL become very

complicated and the generator very laborious to build [41]. For example, software to ‘manage microarray experiment data’ fits inside the MOLGENIS family while software to ‘calculate gene networks from the collected microarray data’ does not. This may sound strange to a life scientist given the obvious biological commonalities amongst raw and processed microarray data but a calculation tool has different informatic needs (e.g. running/stopping algorithms) than a database tool (e.g. storing/searching data). Such calculation tools can for example be modeled by manually adding a plug-in written in the R statistical language, which although at a much lower level of abstraction, can also be considered a DSL to efficiently model statistical protocols (as compared to describing such protocols in a general programming language).

V. RECOMMENDATIONS

What has already become clear in the CASIMIR pilots conducted so far is that whatever standards are adopted they will inevitably remain dynamic and continue to develop, particularly as new data types are collected. Crucially they should allow the open-ended development of new analytical and data mining software and integration of efforts to agree such standards and develop new software is essential. This paper explored bioinformatics models to support such development to timely produce software infrastructures that ‘mouse geneticists really want to have’. How can the mouse community optimally benefit?

First, we recommend the development of a catalog of mouse specific databases and tools (e.g. for running analysis tools and data integration workflows) including user interfaces so mouse researchers can use them. They should also include the underlying DSL models, or modules thereof (MOLGENIS data models, Taverna integration workflows, BioMART queries, R analysis scripts) to help mouse genetics software developers to optimally benefit from each other’s work notwithstanding variation in research aims. An interesting example on how that can work is shown in the myExperiment.org project: a social networking portal where researchers can upload and download Taverna models of analysis workflows over the internet.

Second, we recommend standardization of common parts of the infrastructures models (in DSL) to reduce the need for ‘shims’ for making databases and tools talk to each other. However, mouse genetics is developing rapidly and methodologies to generate and analyze data are still being established which makes it hard to know what standards should look like. For this purpose, extensible data models have been proposed such as FuGE [42], the extensible data model for high-throughput investigations. These models exploit the fact that while the details of experiments may vary wildly, they share commonalities in terms of having protocols, applications of these protocols, samples, data which can be addressed in a standard way. The CASIMIR consortium, in collaboration with the GEN2PHEN

consortium for human genetics, is now developing such extensible ‘standard’ data model for molecular phenotypes and genotypes [19].

Finally, we recommend that domain specific toolboxes (like MOLGENIS, Taverna, BioMART, R) should become more seamlessly integrated from a biologist perspective. For example, one can now already seamlessly access BioMART and MOLGENIS from within Taverna workflows but use of Taverna itself requires significant background knowledge which is beyond non-technical users. Integration of databases and workflows such that they can be run at the push of a button from within, for example, the MOLGENIS user interface promises a future with many benefits from the generation of ‘dynamic software infrastructures for mouse genetics’.

REFERENCES

- [1] R. C. Jansen and J. P. Nap, "Genetical genomics: the added value from segregation," *Trends in Genetics*, vol. 17, pp. 388-391, 2001.
- [2] J. T. Eppig, C. J. Bult, J. A. Kadin, J. E. Richardson, J. A. Blake, A. Anagnostopoulos, R. M. Baldarelli, M. Baya, J. S. Beal, S. M. Bello, W. J. Boddy, D. W. Bradt, D. L. Burkart, N. E. Butler, J. Campbell, M. A. Cassell, L. E. Corbani, S. L. Cousins, D. J. Dahmen, H. Dene, A. D. Diehl, H. J. Drabkin, K. S. Frazer, P. Frost, L. H. Glass, C. W. Goldsmith, P. L. Grant, M. Lennon-Pierce, J. Lewis, I. Lu, L. J. Maltais, M. McAndrews-Hill, L. McClellan, D. B. Miers, L. A. Miller, L. Ni, J. E. Ormsby, D. Qi, T. B. Reddy, D. J. Reed, B. Richards-Smith, D. R. Shaw, R. Sinclair, C. L. Smith, P. Szauter, M. B. Walker, D. O. Walton, L. L. Washburn, I. T. Witham, and Y. Zhu, "The Mouse Genome Database (MGD): from genes to mice—a community resource for mouse biology," *Nucleic Acids Research*, vol. 33, pp. D471-5, Jan 1 2005.
- [3] E. J. Chesler, L. Lu, S. M. Shou, Y. H. Qu, J. Gu, J. T. Wang, H. C. Hsu, J. D. Mountz, N. E. Baldwin, M. A. Langston, D. W. Threadgill, K. F. Manly, and R. W. Williams, "Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function," *Nature Genetics*, vol. 37, pp. 233-242, 2005.
- [4] M. D. Mailman, M. Feolo, Y. Jin, M. Kimura, K. Tryka, R. Bagoutdinov, L. Hao, A. Kiang, J. Paschall, L. Phan, N. Popova, S. Pretel, L. Ziyabari, M. Lee, Y. Shao, Z. Y. Wang, K. Sirotkin, M. Ward, M. Kholodov, K. Zbicz, J. Beck, M. Kimelman, S. Shevelev, D. Preuss, E. Yaschenko, A. Graeff, J. Ostell, and S. T. Sherry, "The NCBI dbGaP database of genotypes and phenotypes," *Nature Genetics*, vol. 39, pp. 1181-6, Oct 2007.
- [5] V. J. Carey, M. Morgan, S. Falcon, R. Lazarus, and R. Gentleman, "GGtools: analysis of genetics of gene expression in bioconductor," *Bioinformatics*, vol. 23, pp. 522-3, Feb 15 2007.
- [6] R. Alberts, G. Vera, and R. C. Jansen, "affyGG: computational protocols for genetical genomics with Affymetrix arrays," *Bioinformatics*, vol. 24, pp. 433-4, Feb 1 2008.
- [7] J. Fu, M. A. Swertz, J. J. Keurentjes, and R. C. Jansen, "MetaNetwork: a computational protocol for the genetic study of metabolic networks," *Nature Protocols*, vol. 2, pp. 685-94, 2007.
- [8] S. V. Bhave, C. Hornbaker, T. L. Phang, L. Saba, R. Lapadat, K. Kechris, J. Gaydos, D. McGoldrick, A. Dolbey, S. Leach, B. Soriano, A. Ellington, E. Ellington, K. Jones, J. Mangion, J. K. Belknap, R. W. Williams, L. E. Hunter, P. L. Hoffman, and B. Tabakoff, "The PhenoGen informatics website: tools for analyses of complex traits," *BMC Genetics*, vol. 8, p. 59, 2007.
- [9] H. Zeng, L. Luo, W. Zhang, J. Zhou, Z. Li, H. Liu, T. Zhu, X. Feng, and Y. Zhong, "PlantQTL-GE: a database system for identifying candidate genes in rice and Arabidopsis by gene expression and QTL information," *Nucleic Acids Research*, vol. 35, pp. D879-82, Jan 2007.

- [10] Z. L. Hu, E. R. Fritz, and J. M. Reecy, "AnimalQTLdb: a livestock QTL database tool set for positional QTL information mining and beyond," *Nucleic Acids Research*, vol. 35, pp. D604-9, Jan 2007.
- [11] M. Mueller, A. Goel, M. Thimma, N. J. Dickens, T. J. Aitman, and J. Mangion, "eQTL Explorer: integrated mining of combined genetic linkage and expression experiments," *Bioinformatics*, vol. 22, pp. 509-11, Feb 15 2006.
- [12] KEGG API. (last accessed 16 April 2008). <http://www.genome.jp/kegg/soap/>
- [13] NCBI E-utils. (last accessed 16 April 2008). http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
- [14] Ensembl API. (last accessed 16 April 2008). <http://www.ensembl.org/info/using/api/index.html>
- [15] M. A. Swertz and R. C. Jansen, "Beyond standardization: dynamic software infrastructures for systems biology," *Nature Reviews Genetics*, vol. 8, pp. 235-43, Mar 2007.
- [16] M. A. Swertz, E. O. de Brock, S. A. F. T. van Hijum, A. de Jong, G. Buist, R. J. S. Baerends, J. Kok, O. P. Kuipers, and R. C. Jansen, "Molecular Genetics Information System (MOLGENIS): alternatives in developing local experimental genomics databases," *Bioinformatics*, vol. 20, pp. 2075-2083, 2004.
- [17] MOLGENIS - Molecular Genetics Information System generator. (last accessed 16 April 2008). <http://molgenis.sourceforge.net>
- [18] R. Ihaka and R. C. Gentleman, "R: A language for data analysis and graphics," *Journal of computational and graphical statistics*, pp. 399-414, 1996.
- [19] M. A. Swertz, B. M. Tesson, R. A. Scheltema, G. Vera, R. Alberts, P. Schofield, K. Schughart, J. M. Hancock, D. Smedley, K. Wolstencroft, H. Parkinson, E. O. Brock, A. R. Jones, C. Goble, m. o. t. C. consortium, and m. o. t. G. P. consortium, "A minimal and extensible data model for genetical genomics," Submitted.
- [20] M. A. Bogue, S. C. Grubb, T. P. Maddatu, and C. J. Bult, "Mouse Phenome Database (MPD)," *Nucleic Acids Res*, vol. 35, pp. D643-9, Jan 2007.
- [21] Mouse Phenome Database (MPD). (last accessed 16 April 2008). <http://www.jax.org/phenome>
- [22] P. Flicek, B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, T. Eyre, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, R. Holland, K. L. Howe, K. Howe, N. Johnson, A. Jenkinson, A. Kahari, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, K. Megy, P. Meidl, B. Overduin, A. Parker, B. Pritchard, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. J. Vilella, J. Vogel, S. White, M. Wood, E. Birney, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, J. Herrero, T. J. Hubbard, A. Kasprzyk, G. Proctor, J. Smith, A. Ureta-Vidal, and S. Searle, "Ensembl 2008," *Nucleic Acids Res*, vol. 36, pp. D707-14, Jan 2008.
- [23] Ensembl. (last accessed 16 April 2008). <http://www.ensembl.org>
- [24] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi, "KEGG for linking genomes to life and the environment," *Nucleic Acids Res*, vol. 36, pp. D480-4, Jan 2008.
- [25] "Kyoto Encyclopedia of Genes and Genomes (KEGG)."
- [26] L. Stein, "Creating a bioinformatics nation," *Nature*, vol. 417, pp. 119-120, 2002.
- [27] D. Smedley, M. A. Swertz, K. Wolstencroft, G. Proctor, E. Birney, M. Zouberakis, P. Schofield, and a. o. m. o. t. C. consortium, "Solutions for database interoperability: a report from the CASIMIR consortium," submitted.
- [28] BioMart. (last accessed 16 April 2008). <http://www.biomart.org>
- [29] A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birney, "EnsMart: a generic system for fast and flexible access to biological data," *Genome Res*, vol. 14, pp. 160-9, Jan 2004.
- [30] Taverna. (last accessed 16 April 2008). taverna.sourceforge.net
- [31] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, "Taverna: a tool for building and running workflows of services," *Nucleic Acids Res*, vol. 34, pp. W729-32, Jul 1 2006.
- [32] Mouse Phenome Database download center. (last accessed cited 16 April 2008). <http://phenome.jax.org/pub-cgi/phenome/mpdcgi?rtn=docs/downloadcenter>
- [33] MOLGENIS integration demo for Mouse Phenome Database. (last accessed cited 16 April 2008). <http://www.casimir.org.uk/molgenis/mpd>
- [34] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. C. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. H. Zhang, "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, 2004.
- [35] R. H. Fogh, W. Boucher, W. F. Vranken, A. Pajon, T. J. Stevens, T. N. Bhat, J. Westbrook, J. M. C. Ionides, and E. D. Laue, "A framework for scientific data modeling and automated software development," *Bioinformatics*, vol. 21, pp. 1678-1684, 2005.
- [36] P. A. Covitz, F. Hartel, C. Schaefer, S. De Coronado, G. Fragoso, H. Sahni, S. Gustafson, and K. H. Buetow, "caCORE: A common infrastructure for cancer informatics," *Bioinformatics*, vol. 19, pp. 2404-2412, 2003.
- [37] D. Jameson, K. Garwood, C. Garwood, T. Booth, P. Alper, S. G. Oliver and N. W. Paton, "Data capture in bioinformatics : requirements and experiences with Pedro", *BMC Bioinformatics*, vol. 9, pp 183, 2008.
- [38] M. D. Wilkinson and M. Links, "BioMOBY: an open source biological web services proposal," *Briefings in Bioinformatics*, vol. 3, pp. 331-341, 2002.
- [39] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor, W. Miller, W. J. Kent, and A. Nekrutenko, "Galaxy: a platform for interactive large-scale genome analysis," *Genome Research*, vol. 15, pp. 1451-5, Oct 2005.
- [40] C. Letondal, "A Web interface generator for molecular biology programs in Unix," *Bioinformatics*, vol. 17, pp. 73-82, 2001.
- [41] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*: Addison-Wesley, 2001.
- [42] A. R. Jones, M. Miller, R. Aebbersold, R. Apweiler, C. A. Ball, A. Brazma, J. Degreef, N. Hardy, H. Hermjakob, S. J. Hubbard, P. Hussey, M. Igra, H. Jenkins, R. K. Julian, Jr., K. Laursen, S. G. Oliver, N. W. Paton, S. A. Sansone, U. Sarkans, C. J. Stoeckert, Jr., C. F. Taylor, P. L. Whetzel, J. A. White, P. Spellman, and A. Pizarro, "The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics," *Nature Biotechnology*, vol. 25, pp. 1127-1133, Oct 2007.