

A Platform for the Description, Distribution and Analysis of Genetic Polymorphism Data

Greg D. Tyrelle Garry C. King

School of Biotechnology and Biomolecular Sciences
University of New South Wales
Sydney 2052 NSW, Australia

{greg,garry}@kinglab.unsw.edu.au

Abstract

In this paper we suggest the requirements for an open platform designed for the description, distribution and analysis of genetic polymorphism data. This platform is discussed in terms of our implementation of a phenotypic prediction pipeline with general application to the understanding of genetic variation.

The current state of polymorphism data storage and distribution has several recognised deficiencies. These include the lack of a shared data model and low overlap between databases. To move towards overcoming these limitations we propose a universal data model for polymorphism data called biological variation markup language (BVML). We suggest an aggregation system for pooling resource description framework (RDF) descriptions of polymorphism databases to form *distributed* federated database indexes, which will facilitate the collaborative involvement of numerous laboratories. An ad hoc query interface for data mining using the extensible markup language (XML) messaging standard simple object access protocol (SOAP) is proposed. To address security issues associated with the storage and exchange of genetic variation data use of public key cryptography is considered. Software tools to manipulate and visualise BVML are also discussed.

Keywords: SNP, XML, RDF, database, web services, distributed.

1 Introduction

Genetic polymorphisms are small but frequent variations in DNA that give rise to the general phenotypic differences in any given population. These slight differences, which include insertions and deletions (indels), repetitions and point mutations, can potentially give rise to disease phenotypes. Point mutations, often referred to as single nucleotide polymorphisms (SNPs), are of particular interest as they represent the bulk of variation in the human genome. To date mainly deleterious SNPs have been characterized due to their distinct phenotypic effects. In half of all genetic disease SNPs exert their effects at the level of protein (Cooper et al., 1998). Thus deleterious SNPs are generally a result of amino acid changes in expressed proteins. Examples of non-synonymous coding SNPs (ns-cSNPs) associated with disease states include many of the more common

genetic diseases such as the Factor V variant deep-venous thrombosis relationship (Kottke-Marchant, 2002).

It is not always the case that the biological effects of SNPs are a consequence of amino acid changes in expressed proteins. DNA regulatory regions where transcription factors (TF) bind may be affected. Both exonic and intronic SNPs can cause mRNA products to be incorrectly spliced (Cartegni et al., 2002). It has also been shown that changes in RNA secondary structure can reduce levels of protein expression (Shen et al., 1999). These disruptions must cause flow-on effects in protein-protein interaction networks along with the non-synonymous changes that may affect catalysis, stability and binding sites on proteins.

1.1 Predicting the phenotypic affects of polymorphisms

Following publication of the draft human genome over 2 million SNPs have been reported (Sachidanandam et al., 2001), approximately 1 in every 1250 basepairs (Reich et al., 2002). Automated SNP discovery methods produce many putative variants of unknown biological significance. These represent a huge pool of potential disease-association candidates or risk factor determinants. To address this issue the phenotypic consequences of many putative variants may be predicted using automated methods (Sunyaev et al., 2000). To date techniques for predicting the phenotypic consequences of SNPs have been based upon non-synonymous coding SNPs. Two approaches have been reported in the literature: protein structural validation (Chasman and Adams, 2001; Sunyaev et al., 2001) and sequence homology-based methods (Ng and Henikoff, 2001).

These approaches ignore potential effects at DNA regulatory regions, mRNA structural effects, splice effects and network effects that will also have significance in some instances. We have chosen a model biological system, the hemostasis pathway, to begin to structurally model SNPs and their potential effects at all levels from DNA up to protein interaction networks. We wish to make this the foundation of a larger knowledge base for researchers and clinicians to support understanding of genetic variation and improved genotyping technologies.

1.2 The need for distributed infrastructure

An impediment to automated analysis of polymorphism data has been the fact that SNPs and associated data are

not covered by any single federated database. At present a number of publicly-accessible federated databases exist, including dbSNP (www.ncbi.nlm.nih.gov/SNP), HGVBase (hgibase.cgb.ki.se), HGMD (www.hgmd.org), JSNP (snp.ims.u-tokyo.ac.jp), ENSEMBL (www.ensembl.org) and The SNP Consortium (TCS, snp.cshl.org). Researchers have ready access to these databases however their utility is only now beginning to be examined. It was found in a comparison of seven major databases that there was very little overlap between them (Marsh et al., 2002). Consequently to gain a comprehensive view researchers must currently consult several databases. Databases were also found to contain false positives that were not efficiently culled. Bioinformatics analysis is also hampered by the lack of a universal data model shared between these databases. Further issues include licensing restrictions on some databases preventing data from being downloaded and difficulty in determining update frequencies of individual databases.

Much of the available characterization data for known SNPs is found in the biological literature and on the web pages of individual laboratories that specialise in the study of a given gene or biological system. Databases of this kind are commonly referred to as locus-specific databases (LSDB). In many cases SNP and associated data, such as literature references, have been curated by hand and deposited in LSDBs in a custom data format. The data are often presented as HTML-formatted pages that are not designed to be machine-readable. An extensive survey of the structure and content of LSDBs (Claustres et al., 2002) shows that they suffer from many of the same problems as federated databases, such as redundancy and inconsistent data formats. To make full use of this data an interoperable distributed platform for analysis and storage of SNP data is needed.

Existing LSDB projects are focused on manual curation and are specifically interested in patient data. Of these projects MuStar™ (Brown and McKie, 2002) and UMD (Beroud et al., 2000) employ proprietary RDBM systems with custom internal schema. Both systems focus on user input and curation of human variation data, with no allowance for indexing, distribution or exchange of the data. For MUTbase (Riikonen and Vihinen, 1999) Bioperl (www.bioperl.org) is employed as the underlying development library and visualisation is done via browser clients. The data format is a machine-parsable flatfile format. However the licensing is restrictive and access to data is only via web pages, which does not allow for easy distribution and machine indexing of the underlying data. No universal data model is available however the seqDiff markup used by the Bio::Variation::IO objects in the Bioperl project is promising.

This paper outlines our effort to design a distributed, flexible, language- and OS-independent platform for description, distribution and analysis of polymorphism data. The components of this platform will ultimately underlie our automated phenotypic prediction pipeline for polymorphism data. Interoperable genetic polymorphism databases may form a *loosely distributed* federated and annotated database with proximal linkages representing

haplotypes. We address how this will be accomplished with our platform along with various associated issues such as data security and veracity. We believe this approach will scale with increasing data production and eventually be more flexible to changes in future bioinformatics standards. Moreover, this approach also allows for individual laboratories to maintain their own annotated store of genetic variation data, leaving curation in the hands of the experts for that particular biological system and avoiding the pitfalls associated with information monopolies.

2 Platform requirements

A web services model has been suggested recently for bioinformatics data providers as a mechanism to achieve interoperability between heterogeneous data sources (Stein, 2002). This model encompasses many of the requirements for our platform design. The guidelines suggest the use of standard formats, or standard markup formats, tools and web application programming interfaces (APIs) to access these resources. We agree in principle with these guidelines and have designed our platform to adhere to them where possible.

We focus on the following key areas for the design and specification of the platform:

- Data model and security
- Querying and indexing metadata
- Data processing and visualization

We do not wish to pre-empt standards bodies or compromise the future interoperability of our platform. Thus the platform must be based as much as possible on existing standards for modelling and exchanging data over the internet. Platform development has taken place on the Mac OS X (10.1.2) operating system.

2.1 Data model and security requirements

The Mutation Database Initiative (MDI, ariel.its.unimelb.edu.au/~cotton/mdi.htm) has put forth recommendations for the ongoing construction of LSDBs (Scriver et al., 2000). We base the following requirements for a universal genetic polymorphism data model on these recommendations. A proposal for diagnostic LSDBs presented at the 4th Australian Mutation Detection Workshop 2002 was also used as a guide.

A universal data model for genetic polymorphisms must contain metadata about the reference sequence and its variants. The model needs to be generic enough to describe all types of genetic polymorphisms associated with a given sequence or genotype, such as indels, repetitions and point mutations. A core set of elements must describe the effects of polymorphisms at the levels of DNA, mRNA, protein and network. Unique identifiers for referencing both the gene and the polymorphisms it contains are required. Cross-references to other relevant data sources must also be included along with the formal genotype of the variant.

Descriptions of laboratory reports for both validation of the variant and computational phenotypic prediction are

necessary. The model must allow for extended information to be included without cluttering the core description i.e. the data model must be modular. Given the potentially sensitive nature of polymorphism data the model must allow for security mechanisms. Lastly the format of the data must be easily parsable, ideally human readable and in a format that allows for easy transformation to different views of the data.

2.2 Indexing and querying of polymorphism metadata

Initially a simple mechanism must be in place for gathering metadata about genetic polymorphisms. Metadata descriptions for the polymorphism may be distilled from the individual polymorphism data models while remaining independent of the polymorphism description format. Metadata descriptions must be modular to cope with future needs and the indexed data must be made available at central sites that can be queried via a web API. This simple model will facilitate adoption of standards amongst polymorphism databases.

In future each polymorphism database should be queried via its own web API, which will produce a bi-directional communication model. Communication between central polymorphism database index sites and the individual databases will produce a loosely distributed federated database framework.

2.3 Data processing and visualisation requirements

Adoption of any platform standard requires development tool kits to be made available to database maintainers. Efforts must be made to produce code libraries in a variety of languages for parsing a universal polymorphism file format, converting existing polymorphism databases to new file formats and for minimal processing and checking of data. Documentation for developers and database administrators to aid in converting existing polymorphism databases will also be essential.

Simple mechanisms for visualisation of polymorphism data that also allow for more extended data views in the future are required.

2.4 Open source

Much debate has taken place with respect to the Open Source movement and its impact on academic bioinformatics. One crucial aspect of any scientific analysis is the publication of methods and data. To this end the fit between the Open Source philosophy and the scientific method is appropriate. The open source model is not incompatible with the sharing of proprietary or sensitive scientific data and methods. We have chosen to build components of our platform using open source tools and libraries thus making our system available to the wider research community for evaluation and use.

3 Biological Variation Markup Language

The construction and deployment of biological databases has grown in a heterogeneous fashion, with the proliferation of data models and file formats from different providers proving an impediment to rapid aggregation, analysis and sharing of basic biological data. No universal data model for the exchange of polymorphism data is presently in use for not only basic descriptive data but also for additional data derived from bioinformatics analyses. The need for a universal data model for describing genetic variation is essential for the future interoperability of genetic polymorphism databases. We have chosen to implement a biological variation markup language (BVML) in extensible markup language (XML) for modelling polymorphism data. We take a gene-centric approach in our data model for use with LSDBs and derived bioinformatics data.

The XML specification published by the W3C (www.w3c.org/TR/2000/REC-xml20001006) has proven to be effective for specifying semi-structured data. It has inherent modularity such as the combining of two document models using name spaces. It is ideally suited to the Internet, is to a certain degree human-readable and a variety of tool libraries are available for parsing XML formatted data.

XML is a means of formalising the syntax for a data model. It does not specify how XML data should be queried or stored. Given the structured nature of XML, technologies for transforming XML into relational data models are well known. Individual implementers may choose from a variety of relational databases, build biological content management systems, or simply store XML documents on the file system. New XML databases and XQL are emerging for the storage of XML formatted data (Wong et al., 2001).

The bioinformatics community has begun producing domain specific XML markup specifications. To date these have focused on sequence annotation, for example AGAVE (www.animorphics.net/lifesci.html), BSML (www.bsml.org), and GAME (www.bioxml.org/Projects/game/). Annotations, including genetic variation, are usually described as sequence features and are specific to the sequence record. To some extent the description of biological variation at the sequence level is sequence annotation. However to satisfy the descriptive requirements of SNPs and their potential effects at all cellular levels requires a richer markup syntax.

The MDI recommendations mention the description of polymorphism data using XML (Scriver et al., 2000). We assume this to be the seqVar specification developed at the European Bioinformatics Institute (EBI), available online (www.ebi.ac.uk/mutations/central/xml.html). However the draft specification is targeted to description of variants submitted to LSDBs and a more general data model that is not limited to a specific circumstance is needed.

3.1 Core content model

BVML consists of various main container elements, which describe metadata about the gene, the variants it contains, haplotype groups and reports that describe either the validation or the analysis of the variant. If the document describes a genotype (a group of patient alleles), patient data maybe included as metadata in the document.

```
<?xml version="1.0"?>
<bvml>
  <bvmlinfo>
  </bvmlinfo>
  <variants>
  </variants>
  <haplotypes>
  </haplotypes>
  <reports>
  </reports>
</bvml>
```

Figure 1 – BVML document outline

BVML consists of three key top-level container elements, Bvml-Info, Variants and Reports. The *bvmlinfo* element holds metadata about the sequence and variants the document describes. Descriptions of the effects of polymorphisms at the levels of DNA, mRNA, protein and network are contained within the *variant* element. Reports describing validation of polymorphisms or bioinformatics analysis are contained within the *report* element. Haplotypes groups are represented by *haplotype* elements.

Metadata describing the polymorphism data contained in a BVML document is included in the *bvmlinfo* element. This section of the document should ideally be automatically generated by software used to maintain the database. Important metadata such as the details of the database curator, official human genome organization (HUGO, www.gene.ucl.ac.uk/nomenclature/) gene name, links to reference sequences, intron-exon boundaries for the gene, links to reference sequences and homologous sequences in other databases are included.

The variant descriptive elements defined for BVML are adapted from the MDI mutation database guidelines (Scriver et al., 2001). For each allele there is a variant element that encloses data that describes the variant at the level of DNA, RNA, protein and network. Variants may be described in terms of a reference sequence or alternatively if upstream and downstream sequence is provided, variant locations maybe positioned relative to any sequence using alignment software.

BVML also provides a generic report element for describing methods and associated genetic data used in the validation of polymorphisms. The report element may also be used to describe bioinformatics analysis methods.

Report elements are linked in the BVML document to the variant that they describe, multiple reports maybe linked to the same variant. Figure 1 shows a proposed outline of a BVML document.

Support for universal naming schemes is emerging from life science standard bodies: the I3C (www.i3c.org) has proposed the life sciences identifier protocol (LSID, Apgar et al., 2002), which at present is in draft stage. LSID however provides a unique ID for all types of biological data. The naming scheme is based on a unique laboratory domain identifier with sub-identifiers allowed. We have included the LSID in our draft specification of BVML to identify the document and the variants contained within the document.

Haplotypes, combinations of closely linked alleles, are of interest to geneticists for disease association studies. For the purposes of description in BVML haplotypes are defined as a combination of alleles of closely linked loci contained in a haplotype element. A given haplotype is referenced in BVML using haplotype ID and multiple haplotypes may be present. We propose that the haplotype ID is in the form of a LIDS identifier.

3.2 Modularity and extensibility

The XML namespaces specification (<http://www.w3c.org/TR/1999/REC-xml-names-19990114/>) provides a built-in mechanism for partitioning documents or combining different document models. We propose the use of XML namespaces for creating extensible modules for BVML without cluttering the core element specification. A basic description of the polymorphism within a gene may be supplemented with derived or experimental data using custom modules. These modules may include structural data, physiochemical properties and expression data. Figure 2 shows a proposed outline of a BVML document with namespaces.

We are particularly interested in the effects of polymorphisms on bimolecular structures, thus a “structure” namespace was created for the addition of information regarding the potential structural effects on DNA, RNA and protein predicted by the phenotypic analysis pipeline. Figure 2 shows the inclusion of the structure module in a proposed BVML document outline. The structure module includes elements that describe disruptions caused by SNPs in regulatory regions of the gene, RNA secondary structure effects and protein structural effects such as buried hydrophobic charges. A full draft specification for BVML is available online (bioinformatics.kinglab.unsw.edu.au).

3.3 Verifiability and security

The security and veracity of biological data in a BVML document must be assured. For example genetic variations may consist of sensitive patient data or information that is covered by intellectual property laws (Maurer, 2000). Mechanisms to maintain data security while not limiting the utility of the document are therefore needed. We have investigated the use of public key cryptography for encrypting sensitive data and data

```

<?xml version="1.0"?>
<Bvml
xmlns:struct="bvml.kinglab.unsw.edu.au/Struct">
  <bvmlinfo>
  </bvmlinfo>
  <variants>
    <variant id="01">
      <description>
        <dna />
        <rna>
          <struct:rna>
            </struct:rna>
          </rna>
        </description>
      </variant>
    </variants>
    <reports>
      <report var_id="01"
              id="01">
      </report>
      <report var_id="01"
              id="02">
      </report>
    </reports>
  </bvml>

```

Figure 2 – BVML document with XML namespaces

BVML may include custom modules via XML namespaces. The example shown includes the “struct” namespace for including extended information about the structure of the biomolecules that the polymorphism effects. The example shown is the struct:rna extension element. The example also illustrates how multiple report elements can be linked to a single variant.

signing to provide a verifiable audit trail. We have chosen to incorporate cryptographic information into BVML using the draft XML-Encryption-requirements (www.w3.org/TR/xml-encryption-req) from the W3C. The XML-Encryption and signing recommendation from the W3C allows for any part of an XML document to be encrypted depending on a data providers requirements.

A major concern when performing bioinformatics analysis is the veracity of data. We propose that each LSDB uses a lab specific public key pair for encrypting and signing data. XML-Signature incorporates document checksums to allow for document integrity to be assessed. Document integrity is essential when BVML documents are transmitted via unsecured channels. The usage of XML-Signature may require canonical XML documents

to be used as parsers often alter documents during parsing.

If further auditing is required for hand-curated databases then each researcher involved in the maintenance can have an individual key pair generated to sign any changes. The LSID draft specification also contains guidelines for data signing and security, however this is still in the early draft stages. It is however possible to implement either in BVML.

4 Distributed metadata indexing and querying

The proposed MDI recommendations call for global indexes of all LSDBs (Scriver et al., 2000). Our platform contains a specification for the construction of such indexes using BVML metadata. Metadata is “data about data” and is useful for giving context to data contained in indexes and potentially for locating data. Thus metadata makes documents *machine understandable*.

4.1 Resource description framework (RDF)

BVML may include resource description framework (RDF) descriptions of data contained in the document. RDF descriptions may also be used to describe the BVML document itself. Specifically RDF is a specification from the W3C for metadata descriptions proposed as part of the semantic web initiative and is able to be expressed in XML syntax (www.w3.org/TR/1999/REC-rdf-syntax-19990222). These RDF metadata can be collected by a web aggregator and displayed on one or more central mutation web sites. This is in effect applying simple agent architectures to the problem of aggregating heterogenous data sources (Bryson et al., 2000).

Automated indexing of sites using RDF has important advantages over manually maintained lists. For example building a network model of variation for a biological system using distributed polymorphism databases that produce RDF descriptions of their data becomes straightforward. Producing RDF descriptions for polymorphism data is easily facilitated by our platform specification, however RDF can be independent of the BVML metadata specification. If other polymorphism databases choose to support the RDF metadata model then distributed queries can be made of this knowledge base. The basic architecture of this component is shown in Figure 3. Support scripts can be written for existing databases or if needed the description can be hand-written with a text editor.

4.2 SOAP query interface

The application of RDF for describing polymorphism metadata does not address querying that data. A web services model (Stein, 2002) suggests an ad hoc querying mechanism for data stored on web pages via a simple web API or remote procedure call interface (RPC). The simple object access protocol (SOAP, www.w3.org/TR/SOAP) provides an XML messaging and RPC specification. Several existing bioinformatics initiatives are deploying SOAP services. Examples include a BLAST web service (mendel.mc.duke.edu:8090/services/blast) and the

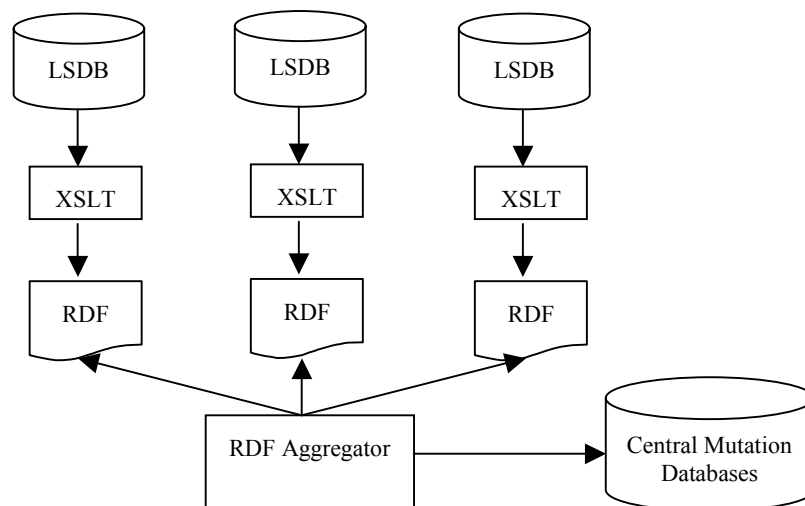


Figure 3 – Distributed RDF architecture

Each LSDB in this architecture produces a resource description framework description (RDF) that describes the data it contains and its associated variants. These are distilled from BVML documents via XSLT. A web aggregator then collects this metadata and stores it in a central mutation database. The central mutation database will employ a simple web API for ad hoc queries of the aggregated data. The web API will be implemented as a simple object access protocol (SOAP) remote procedure call.

Biomoby bioinformatics web services registry (www.biomoby.org).

As a first stage goal in the construction of central polymorphism database index sites, a simple RPC method for querying the aggregated metadata via SOAP messages will be implemented. A SOAP query interface to individual databases could be extended in the future to include search by ontologies or more complex searching and querying via other SOAP interfaces. The SOAP specification allows for envelope rewriting and thus proxying of messages. Central polymorphism database index sites could therefore proxy queries to individual databases. Using standards such as XML and SOAP mean that moving a system such as this to other indexing services should not be difficult.

5 Data processing and visualisation

After data has been exchanged some level of processing is required. Tool kits for parsing the data are necessary so that other software components may then build on top of these tool kits. One of the keys to having any platform accepted by the community is the ease of use for developers. We have chosen the Python scripting language (www.python.org) and specifically the Biopython bioinformatics libraries (www.biopython.org) as a basis for a comprehensive tool kit for initial processing of genetic polymorphism data.

5.1 Python genetic variation software development kit (PGV-SDK)

Sequences are normally represented as strings in the various sequence objects of the Bio* project libraries (www.open-bio.org). String manipulation is inefficient for adding and representing SNPs and other types of genetic variation in DNA. Due to the addition of base additions or deletions of DNA, coordinate systems will change. To overcome these limitations the current MutableSeq object in the BioPython library has been extended to manipulate DNA as a new Python data structure. This concept was originally implemented as Chain.pm as part of the LiveSeq project now included in the Bioperl distribution. We combine this with a Gene interface to GenBank records and a Mutation object to hold mutation information to be applied to a MutableSeq. These are finally combined in a variation object that does the minimal calculations possible to determine if the mutation creates changes at the DNA, RNA, or protein level.

Additionally we intend to provide parsers for BVML formatted databases and suitable container objects to hold the data. Parsers for many other LSDBs and federated databases will also be included. Additional documentation for developers and database administrators looking to convert databases to BVML or describing their existing databases as RDF metadata descriptions will be provided.

5.2 Data model to container objects

The results of parsing biological documents whether they be flat-file, XML, database query results or other, often result in the return of class container objects. These objects usually consist only of attributes, and can thus be easily serialised as XML (when methods do not need to be persistent). XML technologies such as XSLT, which is a templating language specification for XML documents (www.w3c.org/TR/xslt), can be used to convert BVML documents directly into serialised XML objects. We have implemented various XSLT templates to do this. These style-sheets support the Python XML serialisation standard for lightweight object persistence.

5.3 Data visualisation

Once data has been either processed or exchanged a mechanism for transforming that data into rich visualisations is necessary. The researchers that will mainly be concerned with the end product of data analysis and storage are molecular biology researchers, physicians and clinicians who will be expected to have minimal experience with the techniques and skills of computer science. Thus, the ability to provide rich visualisations of data via different clients is also a requirement.

Using XML as the syntax for BVML allows for standard XSL style-sheets to be written for the transformation of BVML polymorphism data to different visualisation formats. In our initial implementation of BVML we have written simple style-sheets to extract metadata and individual variant descriptions for formatting as simple HTML web pages.

Finally the presentation and controlled flow of data to clients in our implementation is handled by existing open source web servers and application servers such as Apache (www.apache.org) and Webware, a Python web application server (webware.sourceforge.net).

6 Conclusions

Whilst recommendations for LSDBs have been proposed by the MDI they give little consideration to platform details for developing polymorphism databases. We have proposed a simple platform for allowing existing databases to provide a universal data model (BVML) for exchanging data. Central indexes of polymorphism databases can be automatically maintained via machine indexing RDF descriptions of polymorphism data. Interoperable polymorphism databases can therefore act as a larger federated database using this open platform.

An issue that has not been addressed in our platform requirements is ontology integration.

7 References

APGAR, J., BACON, S., GILMAN, B., LIEFELD, T. and WERNER, P. (2002) Life science identifier (LSID): Draft specification for review and comment, Available online:

http://www.i3c.org/workgroups/technical_architecture/index.html

- BEROUD, C., COLLOD-BEROUD, G., BOILEAU, C., SOUSSI, T. and JUNIEN, C. (2000) UMD (Universal Mutation Database): A generic software to build and analyze locus-specific databases. *Hum. Mutat.* **15**(1): 86-94.
- BROWN, A. F. and MCKIE, M. A. (2000): MuStar™ and other software for locus-specific mutation databases. *Hum. Mutat.* **15**(1): 76-85.
- BRYSON, K., LUCK, M., JOY, M. and JONES, D.T (2000): Applying agents to bioinformatics in GeneWeaver. In *Proceedings of the Fourth International Workshop on Collaborative Information Agents*. pp. 60-71.
- CARTEGNI, L., CHEW, S. L. and KRANIER, A. R. (2002): Listening to silence and understanding nonsense: Exonic mutations that effect splicing. *Nature Rev. Genet.* **3**(4): 285-298.
- CHASMAN, D. and ADAMS M. R. (2001): Predicting the functional consequences of non-synonymous single nucleotide polymorphisms: Structure-based assessment of amino acid variation. *J. Mol. Biol.* **307**(2): 683-706.
- CLAUSTRES, M., HORAITIS, O., VANEVSKI, M. and COTTON, R. G. H. (2002): Time for a unified system of mutation description and reporting: A review of locus-specific mutation databases. *Genome Res.* **12**(5): 680-688.
- COOPER, D. N., BALL, E. V. and KRAWCZAK, M. (1998): The human gene mutation database. *Nucleic Acids Res.* **26**(1): 285-287.
- KOTTKE-MARCHANT, K. (2002): Genetic polymorphisms associated with venous and arterial thrombosis. *Arch. Pathol. Lab. Med.* **126**(3): 295-304.
- MARSH, S., KWOK, P. and MCLEOD, H. L. (2002): SNP databases and pharmacogenetics: great start, but a long way to go. *Hum. Mutat.* **20**(3): 174-179.
- MAURER, S. M. (2000): Coping with change: Intellectual property rights, new legislation, and the human mutation database initiative. *Hum. Mutat.* **15**(1): 22-29.
- NG, P. C. and HENIKOFF, S. (2001): Predicting deleterious amino acid substitutions. *Genome Res.* **11**(5): 863-874.
- REICH, D. E., SHAFFNER, S. F., DALY, M. J., MCVEAN, G., MULLIKIN, J. C., HIGGINS, J. M., RICHTER, D. J., LANDER, E. S. and ALTSHULER, D. (2002): Human genome sequence variation and the influence of gene history, mutation and recombination. *Nature Genet.* **32**(1): 135-142.
- RIIKONEN, P. and VIHINEN, M. (1999) MUTbase: maintenance and analysis of distributed mutation databases. *Bioinformatics* **15**(10): 852-859.

- SACHIDANANDAM, R., WEISSMAN, D., SCHMIDT, S. C., KAKOL, J. M., STEIN, L. D., MARTH, G., SHERRY, S., MULLIKIN, J. C., MORTIMORE, B. J., WILLEY, D. L., HUNT, S. E., COLE, C. G., COGGILL, P. C., RICE, C. M., NING, Z., ROGER, S. J., BENTLEY, D. R., KWOK, P. Y., MARDIS, E. R., YEH, R. T., SCHULTZ, B., COOK, L., DAVENPORT, R., DANTE, M., FULTON, L., HILLIER, L., WATERSTON, R. H., MCPHERSON, J. D., GILMAN, B., SCHAFFNER, S., VAN ETEN, W. J., REICH, D., HIGGINS, J., DALY, M. J., BLUMENSTIEL, B., BALDWIN, J., STANGETHOMANN, N., ZODY, M. C., LINTON, L., LANDER, E. S., ALTSHULER, D. and THE INTERNATIONAL SNP MAP WORKING GROUP. (2001): A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature* **409** (6822): 928-933
- SCRIVER, C. R., NOWACKI, P. M., LEHVASLAIHO, H. and THE WORKING GROUP (2000) Guidelines and recommendations for content, structure, and deployment of mutation databases: II. Journey in Progress. *Hum. Mutat.* **15**(1): 13-15.
- SHEN, L. X., BASILION, J. P. and STANTON JR, V. P. (1999): Single-nucleotide polymorphisms can cause different structural folds of mRNA. *Proc. Natl. Acad. Sci. U.S.A.* **96**(14): 7871-7876.
- STEIN, L (2002) Creating a bioinformatics nation. *Nature* **417**(6885): 119-120.
- SUNYAEV, S., RAMENSKY, V. and BORK, P. (2000): Towards a structural basis of human non-synonymous single nucleotide polymorphisms. *Trends Genet.* **16**(5): 198-200.
- SUNYAEV, S., RAMENSKY, V., KOCH, I., LATHE III, W., KONDRASHOV, A. and BORK, P. (2001): Prediction of deleterious human alleles. *Hum. Mol. Genet.* **10**(6): 591-597.
- WONG, R.K., LAM, F., GRAHAM, S. and SHUI, W. (2000): An XML repository for molecular sequence data. In *IEEE International Symposium on Bio-Informatics and Biomedical Engineering*. pp. 35-42.