Mutational Data Loading Routines for Human Genome Databases: the BRCA1 Case

Matthijs van der Kroon and Ignacio Lereu Ramirez
Centro de Investigación en Métodos de Producción de Software (PROS),
Universidad Politecnica de Valencia, Valencia, Spain
mkroon@pros.upv.es
iglera@fiv.upv.es

Ana M. Levin and Óscar Pastor
Centro de Investigación en Métodos de Producción de Software (PROS),
Universidad Politecnica de Valencia, Valencia, Spain
{alevin,opastor}@pros.upv.es

Sjaak Brinkkemper
Department of Information and Computing Sciences Utrecht University,
Utrecht, The Netherlands
s.brinkkemper@cs.uu.nl

Received 8 November 2010; Revised 6 December 2010; Accepted 15 December 2010

The last decades a large amount of research has been done in the genomics domain which has and is generating terabytes, if not exabytes, of information stored globally in a very fragmented way. Different databases use different ways of storing the same data, resulting in undesired redundancy and restrained information transfer. Adding to this, keeping the existing databases consistent and data integrity maintained is mainly left to human intervention which in turn is very costly, both in time and money as well as error prone. Identifying a fixed conceptual dictionary in the form of a conceptual model thus seems crucial. This paper presents an effort to integrate the mutational data from the established genomic data source HGMD into a conceptual model driven database HGDB, thereby providing useful lessons to improve the already existing conceptual model of the human genome.

Categories and Subject Descriptors: Human computing [**Conceptual Modeling**]:

General Terms: conceptual modeling, human genome, BRCA1

Additional Key Words and Phrases: BRCA1, data integration, human genome

1. INTRODUCTION

Looking from an Information System point of view, the human genome is an extremely complex system in which exists a lot of ambiguity. For example, basic concepts of what exactly defines a gene are still not explicitly described by the domain. Biology largely depends on domain experts interpreting data, in order for knowledge to appear. Combining the lack of proper data structure and the very large amounts of data generated, a clear problem emerges. How can domain experts dedicate their limited time to the right pieces of information if these are buried in noise? Computers excel at processing large amounts of data, and thus a logical step would be to apply this excellence to the present day problem in genetics, sifting the noise from potentially useful information. For this process to take place, a conceptual modeling approach is essential: it allows for an adequate representation of the domain. Present day solutions that pretend to do exactly this (i.e. ontologies) usually provide controlled vocabularies instead of fixing a conceptual gamut.

A proper conceptual model is expected to provide a clear data structure, enabling efficient and effective access to genomic data, thereby offering ways of reusing previously researched data by pharmaceutic, medical and research institutes as mentioned by [Pastor 2008]. Also, the paradigm shift implicated by considering the genome as a complex information system is expected to allow for exciting new views. To present day, most bioinformatics research is located in the solution space, by attempting to interpret the data that comes out of 'the black box'. For instance by applying powerful sequence alignment tools like BLAST [Pertsemlidis and Fondon 2001] and BLAT [Kent 2002]. Another point of view is offered by [Pastor 2008], whose efforts are directed at tracing and understanding the processes effectively leading to these data. Essentially, seen from an informatics point of view, finding the source-code, by analyzing the object-code, of what may very well be the most sophisticated software ever to be analyzed: life itself.

It has already been shown that applying a conceptual modeling approach to Information System development increases software quality, as has been discussed by [Pastor and Molina 2007] among others. In order to improve the quality of Genomic Information Systems obtaining a clear and sound structural view of the domain thus seems clear. However, this is only part of the story. For any conceptual model to be useful, it needs to be implemented and put to use. This is a vital issue in terms of data quality assurance. As it is essential to warrant that the contents of the database that corresponds to the conceptual schema are the adequate ones. In this work we emphasize the argument that only guiding the genomic data loading process with a well-defined, semantically precise conceptual schema, it will be possible to assess the quality of the subsequent data contents. This is especially the case if we consider the extremely high amount of hetereogenity, diversity and sometimes even inconsistency that can be found under what many authors call the "genomic data chaos". We could use the image of a cupboard intended to store adequately various objects. If we were to accurately pick a particular piece without searching through all of the contents, we need a system in which everything is correctly in place. Here is what we want to assure using a conceptual schema-centric approach, where only having a precise data structure-provided by the conceptual schema-, we will be able to manage contents well. A well-defined, error-free data loading strategy becomes essential for that purpose.

More concretely, In this work we explore the implementation of the Conceptual Model of the Human Genome (CSHG) in the form of a database, and the process of populating it. Indeed, for any proper exploitation to take place a loaded database is deemed necessary. The CSHG has been devised beforehand in a top-down manner, representing the domain as "should-be" according to present day state of knowledge in both the Biology and Information System domains. The problems resulting from this loading procedure are viewed to be typical for the domain, although only one source of genetic mutations will be looked at in detail: the Human Gene Mutation Database (HGMD) [Stenson et al. 2003]. For this loading procedure to take place, no one-to-one mapping can be made and thus data need to be extracted from the external source, transformed to the new structure and loaded into the new database. The domain-specific problems resulting from this process are stated in this work, and enforce our vision that for the proper understanding of this highly complex, highly ambiguous and highly evolving domain only a sound conceptual modeling strategy will do the job.

In the next section, the genomic domain will be introduced superficially and the problem context will be clarified. The basics of genetics will be mentioned, and literature resources for further reading are proposed. In Section 3 earlier work in the problem domain will be discussed. Section 4 will describe the present day conceptual schema of the human genome (CSHG) clarifying it's conceptual structure, relations and attributes. Section 5 will discuss the results of the extraction of data from the external sources, listing the encountered problems and resulting adjustments to the conceptual model. Ultimately, in Section 6 conclusions will be drawn, along with suggestions for further research.

2. THE GENOMIC DOMAIN

To understand the importance and necessity of a conceptual schema for the human genome, an introduction to the genomic domain is indispensable. [Alberts et al. 2003] provides a very complete guide to cell biology, and for a good understanding of this article some basic knowledge about genetics is recommended. This section serves in no case as an exhaustive guide to genetics, it however aims to provide a minimum of knowledge required to understand the rest of this article.

The chemical structure holding this hereditary information is called deoxyribonucleic acid, or DNA. The syntax in which the genetic code is written, consists of 4 elements; A, C, T and G denoting particular chemicals, referred to as nucleotides, or bases. Each of these nucleotides comprises of 3 components; a phosphate group, a five-carbon sugar and a nucleobase. The phosphate group and the sugar form a backbone structure while the nucleobase defines the nucleotide denotation, or meaning. 4 different nucleobases exist; Adenine, Cytosine, Thymine and Guanine, hence the nucleotide identifiers. In DNA the phosphate groups of each nucleotide bond with the sugar of the next nucleotide forming a sequence of nucleotides in that process. At the same time, nucleobases adhere to each other in a specific way: A only bonds to T and C only bonds to G. See Figure 1 for details.

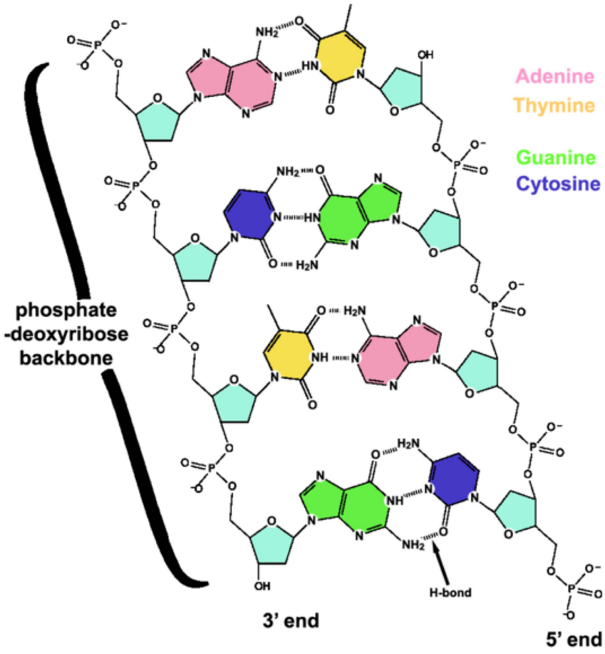

Figure 1. DNA chemical structure.

DNA as formed by these bonds consists of two deoxyribose phosphate backbones with pairs of nucleobases in between. The 3D structure of the molecule is what is known as the famous helix shape. The deoxyribose phosphate backbones are often referred to as strands and since the nucleotides adhere to each other in only one manner, the strand sequences are complementary to each other. This phenomenon is often referred to as one strand sequence being sense and the other anti-sense. The sequence on the strand that is transcribed to mRNA is called the 'sense' sequence, while the sequence on the opposite strand is called the 'anti-sense' sequence. The sense strand then is denoted with a '–' symbol, while the '+' symbol indicates a non-sense strand. Both sense and anti-sense sequences can exist on different parts of the same strand of DNA.

Since the nucleotides bond in an asymmetrical way, sugar to phosphate, DNA has a direction. In the helix, the direction of one strand is opposite to the direction of the other. The strand ends are referred to as the 5' and 3' ends, where the 5' corresponds to the end with a terminal phosphate group and the 3' to the end with a terminal sugar group. Nevertheless, genomes rarely exist as one uninterrupted DNA string, in human beings for example the hereditary information is divided over 23 pairs of DNA strings, commonly referred to as chromosomes. Contrary to what would be expected however, only a small part of the total genome sequence codes for genes. About 95% of the human genome has been designated 'junk' and percentages representing coding parts of the genome range from 1.1% [Venter 2001] to less then 5% [Lander 2001]. The non-coding parts of the genome were considered for a long time to be evolutionary artifacts, serving no present day function. However, recent research shows the non-coding parts of the genome might actually be fulfilling functions, not yet well understood [Mattick 2003; Mattick 2004] and [Ahnert et al. 2008].

Every individual of our species shares approximately 99% of DNA with all other individuals, meaning that the differences observed among individuals can be traced back to around 1% of our genetic sequence. Some of these individual differences, which we will call variations, can be retraced to relatively neutral aspects like hair and eye color, while others have more serious effects like susceptibility to disease. A clear example of the omnipresence of ambiguity in the domain is the lack of a clear distinction in concepts that distinguishes between these different manifestions of genetic variations. A large amount of recent genetic research has been focused on discovering the relationship that exists between genotype and phenotype. Genotype being the chemical ordering of the earlier mentioned bases and represented by a letter sequence "ATGGGCCT". And phenotype interpreted as the manifestation of all visible characteristics of any organism. Clearly for this research to take place succesfully, data about genotype needs to be stored, exchanged and analyzed in large amounts.

The Human Gene Mutation Database (HGMD) [Stenson et al. 2003] provides a pergene repository of mutations, or base changes that research has uncovered to be associated to disease. The full data set of HGMD can be acquired for a fee, while a less complete version is available for free in case of academic use. Given that it is curated by experts, reading scientific papers and extracting the mutational data from them, the source is considered to be highly reliable. At present day the HGMD is considered to be the primary source for obtaining genetic mutations among various genes, although alternatives exist. Many of these alternatives are non-profit based, meaning they provide open-access without limitations. Examples include the Catalogue of Somatic Mutations in Cancer (COSMIC) [Bamford et al. 2004], the HapMap [HapMap 2003], Entrez-SNP (previously known as dbSNP) [Sherry et al. 2001] and many Locus Specific Databases (LSDBs). An interesting effort to standardize these LSDBs, often databases that collect mutations on a single gene, is the Leiden Open Variation Project (LOVD) [Fokkema et al. 2005]. What differentiates the HGMD from these alternatives, is the relatively large quantity of data per gene, and that it represents a collection of many genes. Table I shows an overview of available data sources that contain data on the BRCA1 gene. The HapMap and Entrez-SNP are not included in this overview, since they provide another type of mutational data: so called Single Nucleotide Polymorphisms (SNP's). These SNP's are "usually" not (causatively) associated to disease, as are the mutations that we consider in this work, going into this very interesting issue however is outside the scope of this work.

The Online Mendelian Inheritance in Man (OMIM) [Hamosh et al. 2000; Amberger et al. 2008] project consist of a semi-structured collection of diseases and genetic mutations linked to these diseases. It is available as a single file download but due to it's low degree of structure very difficult to process automatically. The genomic domain can be characterized by three distinct properties: large data quantities, high complexity and rapid evolution. The first property poses certain challenges on resources, like processing time and storage space. Processing the large linkage disequilibrium data files provided by the HapMap [HapMap 2003] resource (around 40 Gb compressed,

Table I. Amount of genes, and amount of mutations for the BRCA1 gene per genetic mutation source.

Source	Genes	BRCA1
HGMD (academic)	2911	1018
HGMD (commercial)	3889	1275
COSMIC	18647	15
LOVD BRCA1/2	2	502

224 Gb uncompressed) for instance certainly counts as challenging, but is by no means impossible due to the regular structure. It is the high complexity and rapid evolution that pose the real problems on the long run as they call for a stable and homogeneous structure, while at the same time allowing for efficient and easy evolution of this same structure. No amount of processing power can compensate for a lack of structure, if what one is looking for simply has not been stored.

3. RELATED WORK

As mentioned, the genomic domain is frequently subject to change as progressing understanding keeps pushing the boundaries of knowledge. Fixing concepts is therefore extremely difficult, but at the same time crucial. Conceptual modeling means looking at a domain in terms of concepts, their properties, behavior and interrelations. It therefore provides a very powerful tool with two different, but related applications. First, conceptual models are used to direct the process of software creation, by functioning as the equivalent of a blueprint used in traditional construction. And second, conceptual models serve as tools to gain a deeper understanding of the domain at hand, often used to drive effective creation of these earlier mentioned blueprints.

Other solutions to the ambiguity problems associated to the genetics domain include ontologies [Ashburner et al. 2000]. To understand why ontologies alone do not fulfill the job of obtaining a full understanding of any given domain, some background information is necessary. Conceptual definitions exist on two levels: conceptually and semantically. The semantic aspect refers to instances of concepts; e.g. the BRCA2 gene, which is an instance of the abstract "*gene*" concept. A problem here for example means ambiguity about naming conventions, for instance the BRCA2 gene is also known as: "Fancd1" and "RAB163". The conceptual aspect is more abstract and handles questions like "What is a gene?". It is our strong belief that for the proper and complete understanding of any given domain, both are vital.

The Gene Ontology (GO) [Ashburner et al. 2000] is a major effort with the aim of standardizing the representation of gene and gene product attributes accross species and databases. The GO resolves many of the ambiguity problems associated to biological terms, and for instance solves the earlier mentioned naming convention problem. Being very useful as it is, GO failes to properly define the concepts that are behind the physical instances. What is missing still, is a holistic view of the human genome as a system, taking into account all aspects that influence the genotype to phenotype process. GO and many other biological ontologies, rather than being a pure ontology as defined by philosophy and information science, actually provide a controlled vocabulary of terms for describing gene product characteristics and gene product annotation data. [Smith et al. 2003] elaborates on this subject by pointing out some drawbacks on GO, among which: "*It is unclear what kinds of reasoning are permissible on the basis of GO's hierarchies*". The conceptual model of the human genome [Pastor 2008] is an initiative at devising this holistic view, resolving the conceptual fuzziness surrounding, among others, the GO's hierarchies.

An information systems approach to this specific problem space is not entirely new. [Okayama et al. 1998] describes the conceptual schema of a DNA database using an extended entity-relationship model. [Paton et al. 2000] advanced on these efforts by presenting a first attempt in conceptually modeling the *S. cerevisiae* genome by proposing a collection of conceptual data models for genomic data. Among these conceptual models are a basic schema diagram for genomic data, a protein-protein interaction model, a model for transcriptome data and a schema for allele modeling.

Whereas [Paton et al. 2000] provides a broader view by presenting conceptual models for describing both genome sequences and related functional data sets, [Pastor 2008] converged on the basic schema diagram for genomic data adapting it to the human genome and eventually produced a database, the human genome database (HGDB) corresponding to this model and following the standard rules of logical design. This database is now in the prototype phase and the first 2 genes, NF1 and BRCA1, have been partially loaded. [Pastor et al. 2010] describes the evolution HGDB went through during the process of conceptually mapping HGDB and HGMD to each other. [Pastor et al. 2010] describes the evolution of the model more in general and provides a descriptive overview of how the model came to be, and from where it evolved to what it is now.

A clear model description requires a well-understood and widely used notation. This is why the selected notation for describing the quoted conceptual models is UML Class Diagrams. Doing so, we establish a link with the Model-Driven community -whose works are normally UML-based in different degrees-, and we also open to door to identify connections between the Model-Driven software production paradigm, and the way in which genomic systems are designed and implemented in terms of software systems. The latter being of specific use in the genetics area, where the knowledge dichotomy between domain experts (biologists) and Information System designers is often hindering efficient development. The main issue when applying a conceptual modeling approach to modeling application domains is the lack of meaning of language constructs as 'object', 'class', 'attribute' and 'operation'. To conceptual modelers, these terms are pretty clear. The problems arise when communication takes place between the modelers and domain experts, who are often not fully aware what these terms mean in Information Systems development context. This vocabulary issue has been addressed earlier by [Evermann and Wand 2004], and they propose ontologically based semantics for object-oriented constructs to improve this. In case of genetics it is the other way around where the same issues arise; concepts well known to domain experts, might make no sense whatsoever to the conceptual modelers, greatly complicating any Information System development. This puzzling fact is a day to day reality in the creation of the conceptual model of the Human Genome. The problem is further enforced by the high level of ambiguity associated to Genetics in general, young as it is. The authors of this work thus propose a similar approach to resolving this issue as have chosen [Evermann and Wand 2004]: combining biological ontologies and the conceptual modeling approach to obtain the desired efficiency of communication that will ultimately lead to the full understanding of the Human Genome.

4. CSHG: RELEVANT CONCEPTS

This section describes the relevant concepts of the CSHG, for a more complete description consult [Pastor et al. 2010]. Initially, an ideal conceptual schema of the human genome was created [Pastor 2008], essentially describing how the concepts should be defined according to the latest knowledge. However, as data was matched from various external sources to this ideal model, it soon became clear a dichotomy existed between the ideal model and the way data was represented in the real world. A second model was created, logically named the real model. This real model serves as a practical tool of resolving the encountered limitations, it therefore compromises on the aspect of understanding the domain. The intention of the real model is to adapt the modeling elements included in the ideal model to the way in which we found that data are stored and managed in practical settings. There is always a conceptual mapping between concepts in the ideal model and how they appear in real models, consequently, we can focus on the ideal model without loss of genericity. The ideal model consists of three sub views; (i) the gene-mutation view, (ii) the genome view and (iii) the transcription view.

Considering that the relevant concepts form part of the gene-mutation view, we will constrain ourselves to this part of the model. This view consists of 3 main concepts, the *Gene, Allele* and *Variation* entities. [figure 2] provides an overview of these present day model relevant concepts. Each of those has various dimensions associated to it, capturing it's relevant properties. To obtain the required information, various external genomic data sources have been used. For genic information, the Hugo Gene Nomenclature Committee (HGNC) database [Eyre et al. 2006] proves to be very useful. HGNC is devoted at providing an internationally accepted list of approved gene names and symbols, thereby standardizing gene nomenclature. The allelic data, among which the actual reference sequence representing a 'standard' allele (the so called NCBI RefSeq), is obtained from the National Center for Biotechnology Information (NCBI) [Pruitt et al. 2006]. The NCBI aims to develop new information technologies to aid in the understanding of fundamental molecular and genetic processes that control health and disease. Data about genetic mutations leading to disease are acquired from the Human Gene Mutation Database (HGMD) [Stenson et al. 2003]. HGMD represents an attempt to collate known (published) gene lesions responsible for human inherited disease.

The *Gene* class captures the general concept of a gene: *ID_symbol* represents an alphanumeric code for the gene according to HGNC, it also functions as the primary key; *ID_HUGO*, a numeric code assigned to the gene by HGNC; *official_name*, the full name of the gene; *summary*, a short description; *chromosome*, the *chromosome* on which the gene is located and *locus*, representing the location of the gene within the

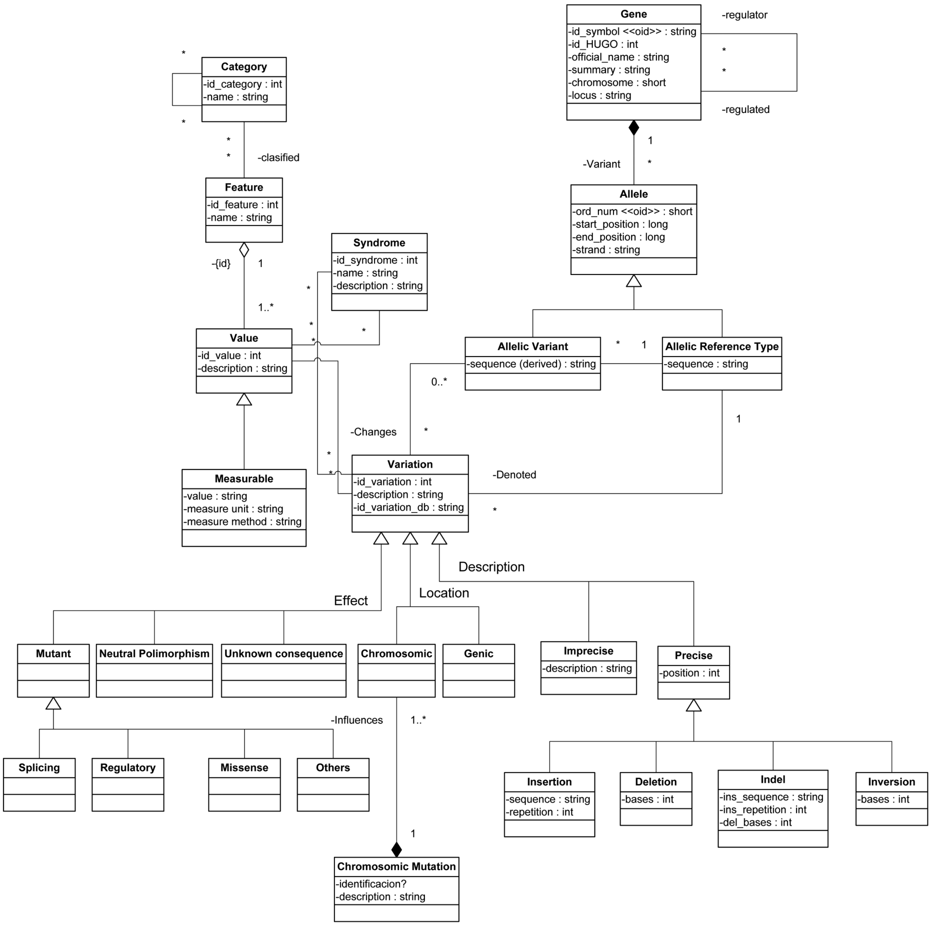

Figure 2. Gene-mutation view of the conceptual model of the human genome.

chromosome.

The *Allele* class then represents the various encountered instances of a gene. It stores information about the start and the end of the allele; the *start_position* and *end_position* attributes respectively, and which strand (+ or –) it is located: the *strand* attribute. It is important to note here, that DNA is double stranded. The *ord_num* attribute functions as an internal identifier. An allele can either be an *Allelic variant* or an *Allelic reference* type. The two entities contain the same *Sequence* attribute which contains a nucleotide string. Each gene has at least one reference allele associated to it, which is obtained from external sources, usually NCBI. The allelic variant tuples then, are derived from this allele reference in combination with information contained in the *Variation* entity. The *Variation* entity stores information about allelic changes in respect to the reference, the allelic reference type. A variation can thus be described as an observed nucleotide at a specific position of a person's DNA, that is not equal to the reference allele. It has an *id_variation* attribute for internal identifying purposes. *id_variation_db* refers to the identification used in the external source. It further holds a description, meant to store a small description about the variation.

The entities specifying the *Variation* concept through generalization can then be classified into three categories; (i) effect, (ii) location and (iii) description, each representing a specific type of polymorphism. The location classes store information about whether the variation affects one or more genes. In the case of a *genic* location, only one gene is affected, while in the case of *chromosomal*, multiple genes might be influenced. The effect entities specify the variations effect on phenotype. This can either be *mutant*, a *neutral polymorphism* or the effect might be *unknown*. The *splicing*, *regulatory*, *missense* and *others* concepts are considered to be mutations since they have a negative effect, hence they are a specialization of the mutant concept. Ultimately, the description classes include descriptive information about the variation. Depending on the degree to which the data on the variation is precise, it falls into either the *precise* or *imprecise* class. When imprecise, the entity only stores a general description. In the case of precise data, it stores the position of the variation and further specifies the nature of the variation into four classes: *insertio*n, *deletion*, *indel* and *inversion*. Each of these concepts store information about this specific type of variation and the exact attributes vary from type to type.

The *Category*, *Feature*, *Value*, *Measurable* and *Syndrome* concepts associate a *Variation* to phenotype. A variation is thus associated to a Value, which in turn is associated to a *Syndrome*. A *Syndrome* is considered to be a negative phenotypic effect, or disease. Every *Value* is composed of *Features*. These *Features* in turn, are classified by *Categories*, which have a recursive property indicated by the self-referencing relation. We do not enter into more details on these phenotype-oriented characteristics. Readers can find a more elaborate definition of all these concepts in [Pastor et al. 2010], where the whole Conceptual Schema of the Human Genome is presented.

5. RESULTS

The first part of this work centers on emphasizing the importance of applying conceptual modeling techniques before constructing any Information System, and Genomic Information Systems in particular. The second part starts here and describes the results of loading the BRCA1 gene data from HGMD into the database that results from the Conceptual Schema of the Human Genome. The encountered problems serve as examples and a case proof of why an Information System without a conceptual modeling sound backbone results in myriad difficulties and undesired behavior, thereby enforcing our earlier mentioned statement.

[Pastor et al. 2010] reports a study of comparing the HGMD to the CSHG, in order to identify a conceptual mapping between the two. It is this mapping that is followed in this document, and the following section will report the encountered problems for actually loading the information from the HGMD into the HGDB for the BRCA1 gene. Roughly problems can be separated into two categories intrinsic data properties and data representation, with the final intention of providing a concrete report on what type of problems the correct load of a conceptual model-based genome data base must face and solve. This is important because the main benefits of applying conceptual modeling principles to the understanding of the human genome are located in both having the right data structure and the right data contents. Identifying and classifying these data loading problems provide general solutions that can help in solving the same problems in other biological data loading contexts. Verifiably incorrect, inconsistent or incomplete data (tuples) are examples of these encountered mishaps with the actual data, or intrinsic data properties. Difficulties associated to physically extracting the data from the external source and ambiguous description of mutation properties are typical examples of data representation problems. Naturally, the division between the two categories is not strict and thus some overlap exists, it is however useful to keep in mind that intrinsic data property problems tend to be affecting the entire genetics domain, while the data representation difficulties are restricted to HGMD.

5.1 Data loading problems

HGMD distinguishes 10 mutation types: Missense/nonsense, Splicing, Regulatory, Small Deletions, Small Insertions, Small Indels, Gross Deletions, Gross Insertions, Complex Rearrangements and Repeat Variations. Roughly all the types can be mapped to the *Variation* and *Precise* concepts of the CSHG, except for the Gross Deletions, Gross Insertions, Complex Rearrangements and Repeat Variations. The latter are described in a very unstructured manner, almost natural language, and are thus considered impossible to process automatically. The CSHG facilitates these tuples as *Imprecise*, which stores a description of the mutation. The HGMD public version does not provide any information about Regulatory mutations for the BRCA1 gene, so these are not considered in this report.

5.1.1 *Intrinsic data properties*. In some cases the HGMD mutational data simply lacks entries. For instance, the splice mutations overview provided by HGMD mentions 5 mutations in intron 22, while [Panguluri et al. 1999] states at least 2 other mutations; IVS22+67(T>C) and IVS22+8 (T>A). Three concrete examples of this problem were encountered, all three in splicing site mutations. However, this particular type of problem is very difficult to detect, since finding them involves rereading the articles HGMD provides which is hard to automate. Thus, although only three concrete occurrences of this problem have been encountered, it is likely more exist.

Splicing site mutation CS961492 describes a C>T mutation, as a possible phenotype HGMD indicates Breast cancer. However, having the read the corresponding article [Langston et al. 1996], not once breast cancer is mentioned in combination with this mutation. The article does mention the mutation as being affiliated with men suffering from prostate cancer. Thus, deducing from the rather limited information made available by HGMD on this specific mutation, it is concluded HGMD made an error during data entry.

Splicing site mutations CS063247 and CS011027 should be located near intron 4, according to the HGMD splicing site mutations overview. However according to the splice junctions overview HGMD provides, there exists no intron 4, nor an exon 4. Since indeed both of the papers [van der Hout et al. 2006] and [Shattuck-Eidens et al. 2009] state the mentioned mutations near intron 4, it would be logical to presume the problem is on HGMDs side. So at first, this specific problem instance was considered to be either a major flaw in HGMD due to inconsistent reference sequences, or a result of human error during the HGMD loading procedure. However, deeper research revealed a more subtle situation. splicing site mutations are located in HGMD by using a splice junctions overview, which provides an overview of intron- and exon borders in the gene. HGMD constructs this overview by using a NCBI reference sequence, in this case L78833. This reference provides a comment in natural language that explains the absence of an exon 4 as: "*Characterization of an aberrant BRCA1 cDNA clone in the original report [Miki et al. 1994] led to the misidentification of an inserted Alu element as exon 4. Not normally found in BRCA1 transcripts, insertion of this Alu would lead to introduction of a STOP codon. Hence, BRCA1 exons and introns are numbered 1a, 1b, 2, 3, 5, 6, etc.*".

Splicing site mutation CS012667 indicates a G>A mutation in nucleotide +3 from the start of intron 2. However neither the HGMD splice junction overview, nor the NCBI reference gene sequence indicates a G-nucleotide at this location. A very similar event happens with splicing site mutations CS001825 and CS991331. They both involve a mutation located 7 nucleotides upstream (+7) from the start of intron 22. However, the first mentions an A>G mutation, while the latter describes, for that exact same location, a T>C mutation. Since both the NCBI reference sequence and the HGMD splice junctions overview indicate an A nucleotide at the appropriate position, and the CS001825 reference article [Khoo et al. 2000] indeed mentions an A>G mutation at the specific location, one could easily conclude the CS001825 A>G mutation in the correct one. However, the CS991331 reference article [Panguluri et al. 1999] does indeed point out a T>G mutation at the location, so the truth might be slightly more subtle. Since [Panguluri et al. 1999] involves an African-American population, while [Khoo et al. 2000] entails a Chinese population, a possible reason for the irregularity might be general genetic differences between those ethnic groups. A phenomenon referred to as Single Nucleotide Polymorphisms, or SNPs [Zhao et al. 2003]. 5 Occurrences of this problem have been identified in splicing site mutations, 1 in Small Deletions and Small Insertions each, leading to a total of 7 occurrences. SNP's are currently included in the CSHG as a separate concept, facilitating the placement of these type of mutations. Finding these type of genetic variations, mixed in with another type of variations does emphasize the need for an unambiguous data representation.

5.1.2 *Data representation.* The HGMD public version, used for this research presents data through a website in HTML tables. This makes an automatic extraction procedure very difficult, but not impossible. Using a 'screen-scraping' approach including HTML parser technology, the individual tuples can be isolated and extracted. This approach however has several drawbacks, among which its inflexibility of coping with changing environments, which especially in this rapid evolving domain is undesirable. It deserves mentioning here that the professional license, available for a fee, does allow for acquisition of the HGMD database as a .sql file, making this extraction process a lot easier. Also, the public version has a delayed update cycle, while the professional license includes the entire up-to-date dataset.

Some data is provided in natural language. For instance the fact that the first two BRCA1 exons are alternative non-coding exons is only mentioned in the header of the Splice Junctions overview: '*The first 2 exons are alternative non-coding exons and The translation initiation codon is located within exon 2*', this mainly affects locating splicing site mutations (1 instance). Adding to this, in Small Deletions (2 instances) and in Small Insertions (3 instances) some mutations are located through mouse-over tags, the information communicated by these tags is highly unstructured, to a degree that we might call it natural language as well. Also, in the case of imprecise mutations (Gross Deletions, Gross Insertions, Complex Rearrangements and Repeat Variations), the greater part of the information presented by HGMD is in natural language, impeding an automated approach severely in the affected cases.

In some cases, the HGMD database uses different ways of locating mutations, within the same type of mutations. For instance, Small Insertion mutations CI030168, CI962219 and CI022582 happen in non-coding areas of the gene, just like the Small Deletions mutations CD991644 and CD994433. Since HGMD generally uses a cDNA codon referenced way of locating these types of mutations, and given that non-coding sequences simply not exist in the cDNA, HGMD locates these earlier mentioned mutations in a different way. In the case of Small Insertions, HGMD provides a Splice Junction reference, very much like the method used to locate splicing site mutations. In this case the CI030168, CI962219 and CI022582 mutations are located at IVS20+21, IVS20+48 and IVS20+64 respectively. So IVS20 indicates the intron number, where +21 indicates the offset, however since no acceptor/donor information is provided, it is unclear from which side of the intron the offset should be referenced. In the case of Small Deletion mutations CD991644 and CD994433 at first sight, no indication of how to locate them is provided. However, this information is provided through mouse-over tags in the Splice Junctions referenced form, described earlier. CD991644 is thus located by I7E8-24, aka IVS7 -15 del10. and CD994433 is located by "I12+34/polymorphism?". This problem was thus encountered 3 times in Small Insertions and 2 times in Small Deletions, making a total of 5 occurrences.

As said, HGMD refers to codons in many cases, which are sets of three nucleotides. Since HGDB will be using a nucleotide referenced position to locate mutations, a transformation of HGMD provided data is necessary. In theory, acquiring the correct nucleotide would be a matter of multiplying the codon number by three, reality is slightly more complicated as will be discussed in the next paragraph. HGMD uses this way of locating mutations in the case of Missense/nonsense (320 instances), most of the Small Deletion mutations (288 instances), most of the Small Insertion mutations (98 instances) and Small Indels (11 instances), leading to a total of 715 instances of this problem. Retrieving the corresponding nucleotide in DNA would simply be a question of multiplying the codon number by 3, if not for the existence of introns and exons. cDNA only comprises of the genes' exons, thereby excluding the introns, contained in the DNA. Due to this fact and given that HGDB will be incorporating a DNA referenced scale, a linear transformation, by multiplying the codon number by 3, simply is not possible.

HGMD splicing site mutations are located by referring them to so-called splice-junctions. These splice junctions indicate the borders between exon and introns.

HGMD thus indicates an intron border, by giving an intron number and specifying which border by providing either a donor- (ds) or acceptor- (as) site of the intron. The donor site corresponds to the side closest to the 5' end of the DNA strand, while the acceptor site corresponds to the side closest to the 3' end of the DNA strand. Then an offset is given, to indicate the amount of nucleotides between the indicated splice junction and the actual mutation. In the so-called splicing mutations overview HGMD then provides a sample sequence for each intron/exon-junction contained in the gene. This method of locating mutations is used primarily in splicing site mutations (80 instances), but in some exceptional cases HGMD also uses this notation to provide locational data for other types of mutations. For instance, In Small Deletions (2 instances) and in Small Insertions (3 instances).

In the HGMD data exists ambiguity; for instance, mutations may or may not result in a certain phenotype, this is indicated by a question mark following the supposed phenotype. However, no probability scores are stated and a mutation without a (noticeable) phenotype is considered to be a variation with neutral effect. Since variations and mutations are considered to be two different concepts in the HGDB data model, this poses problems with loading the database correctly. 94 instances of this problem have been identified: missense/nonsense mutations account for the most instances (73), splicing site mutations contains another 16, small deletion mutations 2 and small insertion mutations account for 3 instances.

In short to summarize the encountered problems in the data loading process. Nine distinct problems have been identified, each of which has been discussed separately in the above section. The following sections will provide useful insights on how to resolve future instances of these. The problems as encountered, separated by category are as follows:

Intrinsic data properties
(1) Lacking data
(2) Data entry errors
(3) Reference to non-existing introns
(4) General inconsistency

Data representation
(1) Data difficult to access
(2) Use of natural language
(3) Inconsistent way of positioning mutations
(4) Inconsistent with NCBI refSeq
(5) General ambiguity

5.2 Data loading problem solutions

Inserting the 804 precise and 90 imprecise variations provided by the non-commercial version of HGMD manually seemed like a cumbersome and more importantly, error prone operation. For this reason a series of scripts was devised to automate the procedure, while at the same time providing useful experience and knowledge about how to further process the variational data automatically. A full technical report on this process, including the source code of these scripts can be found in [van der Kroon et al. 2009]. The basic function of the software is to convert the HGMD provided mutational data into the format used by HGDB and detect inconsistencies in the data. The main tools are the cDNA- sequences provided by HGMD [Genebank, U14680] [Genebank, L78833], the NCBI DNA reference sequence [NCBI NG 005905.1] and the NCBI Coding Sequences (CDS, located in NCBI RefSeq NG 005905.1). The HGMD cDNA sequence is the aggregation of all coding sequences (exons) of the BRCA1 gene, thus excluding introns. The NCBI DNA sequence then is the complete sequence of the BRCA1 gene, including introns. The CDS information specifies what parts of the NCBI DNA sequence are coding and which are not. The scripts extract, and in some cases, calculate the variables which are to be inserted into the HGDB Variation, Precise and Imprecise tables. At the same time detecting inconsistencies in the HGMD database that can not be resolved in an automated way, indicating a manual approach is needed in those cases. Since dealing with the above mentioned problems was necessary to devise those scripts, they are considered to be the crystallized solutions to the earlier mentioned problems. The main disadvantage of this screen-scraping approach, is high vulnerability to changes in HGMD data structure. This means the scripts will require regular updating. Also, by depending on a medium like HGMD, trust is invested in the integrity of the source. However, at this point the exact reliability of HGMD has not been identified and some of the encountered problems during this project clearly suggest reasons to doubt this reliability.

5.2.1 *Intrinsic data properties.* HGMD lacks data entries for two reasons. HGMD might simply not be up-to-date with the latest information provided by scientific research on the subject, or HGMD missed entries during the manual loading of the database. A partial solution to this problem involves using the professional version of HGMD, which includes more and more up-to-date entries. However, it is also a characteristic of the immatureness of the field that no full coverage exists, simply not enough research has been performed to identify all existing mutations. Therefore the database will inherently be incomplete, and thus resolving falls outside the scope of this paper.

HGMD provides erroneous data for a variety of reasons, human error on either HGMD or the source paper side might be an issue. Inconsistencies between HGMD and source paper notation style might play a role. In either case, this is a very difficult problem category to detect since detection involves rereading the papers. A possible, but unsatisfactory to some degree, solution would be to perform a deep investigation on a limited amount of mutational data provided by HGMD, thereby uncovering error frequency. This error frequency, provided it is investigated according to scientific measures, can then be extrapolated to the rest of the database, thus providing a handle from which to calculate reliability of the data-set.

Since inconsistencies can be detected, a manual check of the apparent inconsistent data is possible. Indeed, the scripts indicate only a few inconsistencies and so the shear amount of papers to be read manually can be reduced drastically, making the manual approach possible in these cases. In case the source is locating a mutation within a non existing intron, the solution might be to manually complete the splice junctions overview, combining the information from the NCBI reference sequence and coding sequences information (CDS), however contradiction exists about whether an intron 4 actually exists. The implications of this, as has been discussed earlier that reference papers are using different reference sequences, are far more serious and indicate a structural flaw in the HGMD splicing site mutations data-set since no facilities to indicate such differences exist within HGMD, neither indicate the involved reference papers exactly what reference sequence they are using. Therefore reasons exist to doubt the splicing site mutations integrity. However, detection depends on whether a given sample corresponds to the actual nucleotide occurrence at that position in the DNA sequence. Since HGMD only provides a single nucleotide sample for splicing site mutations, the odds of a nucleotide at any position in the DNA corresponding to it is 25%, reducing error detection reliability greatly. As has been mentioned, in some cases the HGMD mentioned mutation involves a nucleotide change where the to-be changed nucleotide is actually different in the reference DNA gene sequence, thereby indirectly suggesting an error on either the source paper side or the HGMD data entry side. Except for splicing site mutations, as has been discussed earlier, most of these inconsistencies can be detected with relatively high reliability. This is possible because HGMD provides a sample sequence surrounding the mutations, that can be used as a handle to see whether the mentioned nucleotide actually is the one on the calculated position in the DNA sequence. In case of an inconsistency between the reference, and the given sample sequence a manual approach is possible to investigate further.

5.2.2 *Data representation.* Solving the data extraction problems involves copy-pasting the HTML-tables with mutational data, provided by HGMD, into the programming logic of the scripts. By using the PHP explode-function the data is then cut into bite-size chunks and stored in an array, ready for further processing. Due to choosing this screen-scraping approach, information contained in HTML mouse-over tags is not captured. A simple solution to this problem would be to copy-paste the HTML source-code instead of the browsers rendering, effectively including the HTML tags and thus the mouse-over contained information. Then use a more elaborate algorithm to extract the pieces of information from the source. Also, since many biological information sources present their information in HTML tables as discussed by [Stein 2002], many solutions to this specific problem have been devised, although no silver bullet solution exists today. The information contained in these tags is highly unstructured, to a degree where it is considered natural language. Also, the informations structure of the different occurrences of the problem differs highly. So a more generic solution has been chosen to solving this problem: instead of using the information HGMD provides to locate the mutation, the PHP script simply takes the sample string provided by HGMD and matches this to the entire BRCA1 gene sequence, given that the string is unique, a location will be found and presented. Although solving this particular instance proved possible due to the fact the natural language contained non-vital information, future occurrences might be more difficult and no satisfying resolution exists, for coping with natural language inherently is a weakness in computer technology. HGMD has a tendency to disrupt it's own structure, by providing different ways of locating mutations within the same mutation type. This complicates an automated approach, however a generic solution to this problem has been devised. Since the main difficulty here is the uncertainty about whether HGMD might present locational data in yet other ways, it was decided not to use HGMDs locational data at all in these exceptional cases. Instead, the earlier mentioned sample sequence given by HGMD is extracted from the HTML table. The location of the mutation within this sample sequence is then found by detecting a character case change. The result of this detection is then considered to be the offset, very much alike the offset given by HGMD in for instance the missense/nonsense mutations. The entire sample sequence is then matched against the entire BRCA1 reference gene sequence, after which the offset is added to the found location resulting in the absolute mutation position. The major flaw in this approach however, is the fact that the given sample sequence might be happening more then once in the BRCA1 gene, thereby compromising this methods reliability to some extend. For this reason, this approach is only considered usable in case the regular method, which is considered to be more reliable due to the more rigid structure, fails.

The way HGMD stores genetic variation positions, is very distinct from the format in which HGDB stores this information. Each of the HGMD variation types, undergoes a distinct transformation operation, depending on how HGMD indicates the location within the cDNA and will be discussed separately.

Missense/nonsense

HGMD indicates a missense/nonsense location by providing a codon number, referencing cDNA plus an offset. Since HGDB requires a DNA referenced nucleotide position, transformation of this data is required. First, the software composes its own cDNA sequence by extracting and merging the coding sequences from the NCBI DNA sequence using the NCBI CDS information. It then matches the composed cDNA sequence to the HGMD cDNA to detect inconsistencies. By matching the original and substituted codon (ATG>GTG) provided by HGMD the position of the mutated nucleotide in reference to the indicated codon is detected. Subsequently, it multiplies the codon number minus 1 by 3, in order to acquire a nucleotide referenced scale. It adds to this number the exact location of the mutation within the codon hereby obtaining the exact location of the mutation, referenced on a nucleotide cDNA scale. Ultimately, to acquire the correct location within the DNA, a calculation involving the NCBI CDS and nucleotide location of the mutation in the cDNA takes place. The software calculates the length of each coding sequence and each non-coding sequence in the DNA, according to the NCBI CDS. It then identifies the amount of non-coding nucleotides between the start of the NCBI DNA sequence and the mutation position, then adds this number to the cDNA referenced nucleotide location, resulting in the DNA referenced nucleotide position of the mutation.

Splicing

Solving the problems affiliated with the manner in which HGMD locates splicing site mutations, seems rather straightforward at first: by simply using the splice junctions overview provided by HGMD, every mutation should be located. However due to the discovery of HGMDs splicing site mutations overview poor correspondence to its own

Splice Junctions overview, an alternative solution had to be devised. In this case the script uses a matching strategy in which it grabs the given nucleotide sequence from the Splice Junctions overview and matches this to the reference sequence provided by NCBI, thereby locating the location of the mutation more reliably.

Regulatory
The HGMD non-commercial database does not indicate any regulatory problems in the BRCA1 gene.

Small Deletions/Small Insertions/Small Indels
These three types of variations are located by HGMD in exactly the same manner, therefore they are discussed together. HGMD uses a codon referenced cDNA scale to locate these variations, just like with missense/nonsense mutations. However, in this case, the mutation can involve up to 20 bps and therefore often happens outside the referenced codon, either to the 3' or the 5' side. In this case, the transformation software calculates the given codon location in very much the same way as with the missense/nonsense mutations, by matching the NCBI CDS data with the codon number, resulting in a nucleotide position on a DNA scale for the first codon base. The software then calculates the offset in nucleotides between the referenced codon, and the actual start of the mutation, adding (or subtracting, depending on where the mutation starts in relation to the reference codon, downstream or upstream the DNA strand) this quantity to the nucleotide position of the codon in the DNA, resulting in the DNA referenced nucleotide start position of the mutation.

Gross Deletions/Insertions/Complex rearrangements/Repeat Variations
Since these type of mutations are considered to be imprecise, no data about them needs processing. In this case, the script simply appends an identifier and applies the values to the corresponding cells in the HGDB Variation and Imprecise tables.

Absolutely resolving the data ambiguity would include rereading all the papers used to populate HGMD. However, this approach is impossible to automate due to the highly unstructured nature of scientific papers. Hence another solution is suggested: modifying the HGDB to account for the uncertainty by adding a certainty attribute to the variational table. HGMD presents on their background pages an indication of how to interpret this uncertainty by providing the inclusion criteria for Disease-Associated/Functional Polymorphism's.

6. CONCLUSIONS

In this document we have shown the primary reason of existence of conceptual modeling techniques. The HGMD is considered an extremely useful source of data about genetic mutations in the field. For being curated, it is also considered to be highly reliable. This document shows, that although we do not doubt the quality of the data an-sich, a lot remains to be wished for. Some of the problems can be retraced to the relative youth of the genetic domain. The reference to the non-existing intron 4 exemplifies this nicely. The data are difficult to extract, but we do not consider this a real problem. It has been the HGMD decision to choose a more commercial route, and when respected one can obtain a more accessible means of retrieving the desired data. This being said, the apparent lack of a thorough conceptual modeling approach seems to bear it's traces on the service.

Every tuple in the HGMD is supposed to represent a genetic variation, known to be associated to disease. This quite rigorous definition becomes endangered in cases where indicated variations 'might' be associated to disease, as indicated in the HGMD by the question mark. Indeed, a variation that is not associated to disease should not be considered a mutation and thus not enter the dataset as is. The CSHG handles these cases by providing the *neutral polymorphism* dimension, for the *Variation* concept.

Another point of improvement is the lack of a proper way of facilitating the various reference sequence in common use by research papers. For illustration, a certain mutation might be located in position 131 in reference sequence X, but correspond to position 125 in reference sequence Y. The HGMD provides it's own cDNA sequence, from which it locates the majority of it's mutations. However this cDNA sequence is 'based' on an NCBI sequence, and can thus differ from it. For an optimal use of the data provided by HGMD, this means an expert in many cases still needs to evaluate and interpret the data. This is expensive in both time and money. Aligning the HGMD set of mutations to the NCBI reference sequence, that is considered to be the 'golden standard', thus seems a logical step and has been one of the merits of this work.

Concretely, we suggest two major changes to the HGMD: (i) facilitate a more elaborate way of handling associated phenotype, perhaps link directly to the Online Mendelian Inheritance in Man (OMIM) database [Amberger et al. 2008]. And (ii) add a new column, in which the reference sequence indicated by the source paper is also stored. This will allow for a much easier, and more efficient use of the HGMD data set. Considering data is acquired manually from the papers, adding this element of extracted data seems to be relatively low cost.

The CSHG aims to avoid the earlier mentioned problems, by applying the conceptual modeling approach. It is our strong belief that the only way of accurately representing any data, and perhaps genetic data in particular, can only be done by means of careful analysis of the domain and its peculiarities. Furthermore, following the selected conceptual modeling approach, the conceptual schema itself is open to incorporating new concepts and new discoveries in a domain whose constant evolution is without a doubt. A conceptual schema is ready to adapt to any new knowledge in a way that enables having a continously updated whole, correct picture of the problem. The logical step of representing the acquired model appears to be in the form of a conceptual schema. When we look at the HGMD we can not help but notice that although very useful, a lot is still to be wished for from an information systems point of view. It is exactly this what characterizes the present day problems in the genetics domain; so many data are generated but no coherent, holistic view, of these data exists. The data are scattered around the globe in various databases, many much like the HGMD, and hidden among them we possess over solutions to so many problems. It is in this haystack, that the needle will allow for exciting new possibilities and huge improvements in healthcare. We are now facing the choice, to either brute force our way through the haystack, sifting and sifting until we encounter the big prize. Or we structure the haystack, bring order in chaos and obtain an understanding of what exactly composes the mechanical processes that drive life. It is our belief that the only way to do this, is by applying the use of a conceptual modeling approach.

REFERENCES

AHNERT, S., FINK, T., AND ZINOVYEV, A. 2008. How much non-coding DNA do eukaryotes require? *Journal of Theoretical Biology 252*, 4, 587–592.

ALBERTS, B., BRAY, D., HOPKIN, K., JOHNSON, A., LEWIS, J., RAFF, M., ROBERTS, K., AND WALTER, P. 2003. *Essential Cell Biology*, 2nd ed. Garland Science USA.

AMBERGER, J., BOCCHINI, C., SCOTT, A., AND HAMOSH, A. 2008. McKusick's Online Mendelian Inheritance in Man (OMIM (R)). *Nucleic Acids Research.*

ASHBURNER, M., BALL, C., AND BLAKE, J. 2000. Gene ontology: tool for the unification of biology. *Nature genetics 25*, 1, 25–30.

BAMFORD, S., DAWSON, E., FORBES, S., CLEMENTS, J., PETTETT, R., DOGAN, A., FLANAGAN, A., TEAGUE, J., FUTREAL, P., STRATTON, M., ET AL. 2004. The COSMIC (Catalogue of Somatic Mutations in Cancer) database and website. *British journal of cancer 91*, 2, 355–358.

EVERMANN, J. AND WAND, Y. 2004. Ontology bases object-oriented domain modeling: Fundamental concepts. *Requirements Engineering 10*, 2, 146–160.

EYRE, T., DUCLUZEAU, F., SNEDDON, T., POVEY, S., BRUFORD, E., AND LUSH, M. 2006. The HUGO gene nomenclature database, 2006 updates. *Nucleic Acids Research 34*, suppl 1, D319.

FOKKEMA, I., DEN DUNNEN, J., AND TASCHNER, P. 2005. LOVD: Easy creation of a locus-specific sequence variation database using an "LSDB-in-a-box" approach. *Human mutation 26*, 2, 63–68.

HAMOSH, A., SCOTT, A., AMBERGER, J., VALLE, D., AND MCKUSICK, V. 2000. Online Mendelian Inheritance in Man (OMIM) Hum. *Mutat 15*, 57–61.

HAPMAP, C. 2003. The International HapMap Project. *Nature 426*, 6968, 789–796.

KENT, W. 2002. Blat the blast like alignment tool. *Genome Research 12*, 656–664.

KHOO, U., NGAN, H., CHEUNG, A., CHAN, K., LU, J., CHAN, V., LAU, S., ANDRULIS, I., AND OZCELIK, H. 2000. Mutational analysis of brca1 and brca2 genes in chinese ovarian cancer identifies 6 novel germline mutations. *Human Mutation 16*, 1, 88–89.

LANGSTON, A., STANFORD, J., WICKLUND, K., THOMPSON, J., BLAZEJ, R., AND OSTRANDER, E. 1996. Germ-line brca1 mutations in selected men with prostate cancer. *American Journal of Human Genetics 58*, 881–885.

MATTICK, J. 2003. Challenging the dogma: the hidden layer of non-protein-coding RNAs in complex organisms. *Bioessays 25*, 10, 930–939.

MATTICK, J. 2004. RNA regulation: a new genetics? *Nature Reviews Genetics 5*, 4, 316–323.

MIKI, Y. ET AL. 1994. Brca1 mutations in primary breast and ovarian carcinomas. *Science 266*, 5182, 120–122.

OKAYAMA, T., TAMURA, T., GOJOBORI, T., TATENO, Y., IKEO, K., MIYAZAKI, S., FUKAMI-KOBAYASHI, K., AND SUGAWARA, H. 1998. Formal design and implementation of an improved ddbj dna database with a new schema and object-oriented library. *Bioinformatics 14*, 6, 472.

PANGULURI, R., DUNSTON, G., B^{RODY}, L., MODALI, R., UTLEY, K., ADAMSCAMPBELL, L., DAY, A., AND WHITFIELD-BROOME, C. 1999. Brca1 mutations in african americans. *Human Genetics 105*, 1-2, 28–31.

PASTOR, O. 2008. Conceptual modeling meets the human genome. *Conceptual modeling-ER 2008 5231*, 1–11.

PASTOR, O., CASAMAYOR, J., CELMA, M., PASTOR, M., MOTA, L., AND LEVIN, A. 2010. The conceptual schema of the human genoma: Looking at bioinformatics from an information systems perspective. Tech. Rep. TECPROS-12-01, PROS Research Center, Camino de Vera S/N, 46022, Valencia, Valencia, Spain. Sept.

PASTOR, O., LEVIN, A., CASAMAYOR, J., CELMA, M., VIRRUETA, A., AND ERASO, L. 2010. *Model driven-based engineering applied to the interpretation of the human genome*, 1st ed. Lecture Notes in Computer Science, vol. 6520. Springer-Verlag, Chapter 10.

PASTOR, O., LEVIN, A., CELMA, M., CASAMAYOR, J., SCHATTKA, L. E., VILLANUEVA, M., AND PEREZ-ALONSO, M. 2010. Enforcing conceptual modeling to improve the understanding of human genome. In *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on*. IEEE Press, 85–92.

PASTOR, O. AND MOLINA, J. 2007. *Model-driven architecture in practice: a software production environment based on conceptual modeling*. Springer-Verlag. Berlin-Heidelberg.

PASTOR, O., PASTOR, M., AND BURRIEL, V. 2010. Conceptual modeling of human genome mutations: a dichotomy between what we have and what we should have. In *Proceedings of Bioinformatics 2010*. BIOSTEC Bioinformatics, 160–166.

PATON, W., KHAN, S., HAYES, A., MOUSSOUNI, F., BRASS, A., EILBECK, K., GLOBE, C., HUBBARD, C., AND OLIVER, S. 2000. *Proceedings of the IVth Int. Conference on Research Challenges in Information Science*. Vol. 6. Bioinformatics, Chapter Conceptual Modeling of Genomic Information, 548–557.

PERTSEMLIDIS, A. AND FONDON, J. 2001. Having a blast with bioinformatics (and avoiding blastphemy). *Genome Biology 2*, 10, 1–10.

PRUITT, K., TATUSOVA, T., AND MAGLOTT, D. 2006. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic acids research*.

SHATTUCK-EIDENS, D. ET AL. 2009. Brca1 sequence analysis in women at high risk for susceptibility mutations. *The Journal of the American Medical Association 278*, 15, 1242–1250.

SHERRY, S., WARD, M., KHOLODOV, M., BAKER, J., PHAN, L., SMIGIELSKI, E., AND SIROTKIN, K. 2001. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research 29*, 1, 308.

SMITH, B., WILLIAMS, J., AND SCHULZE-KREMER, S. 2003. The ontology of the gene ontology. *American Medical Informatics Association Annual Symposium Proceedings,* 609–613.

STEIN, L. 2002. Creating a bioinformatics nation. *Nature 417*, 6885, 119–121.

STENSON, P., BALL, E., MORT, M., PHILLIPS, A., SHIEL, J., THOMAS, N., ABEYSINGHE, S., KRAWCZAK, M., AND COOPER, D. 2003. The human gene mutation database (hgmd): 2003 update. *Human Mutation 21*, 6, 577–581.

VAN DER HOUT, A. ET AL. 2006. A dgge system for comprehensive mutation screening of brca1 and brca2: application in a dutch cancer clinic. *Human Mutation 27*, 7, 654–666.

VAN DER KROON, M., RAMIREZ, I. L., LEVIN, A., PASTOR, O., AND BRINKKEMPER, S. 2009. Mutational data loading routines for human genome databases: the brca1 case. Report UU-CS-2009-020, Department of Information and Computing Sciences, Utrecht University.

ZHAO, Z., FU, Y., HEWET-EMMETT, D., AND BOERWINKLE, E. 2003. Investigating single nucleotide polymorphism (snp) density in the human genome and its implications for molecular evolution. *Gene 312*, 207–213.

Matthijs van der Kroon obtained his BSc in Information Science at Utrecht University, The Netherlands. He was granted to do his Bachelor thesis at the Centro de Investigación en Métodos de Producción de Software (ProS), a research center of the Technical University of Valencia. In this work, he used conceptual modeling techniques applied to Human Genetics. After graduating he moved to Valencia and started working as a junior researcher at the Genoma team in ProS center. At the same time he is writing his MsC thesis: “Applying conceptual modeling to the Single Nucleotide Polymorphism (SNP) concept”.

Ignacio Lereu Ramirez graduated in Science of Mathematics at the University of Valencia in 1997. He entered the Centro de Investigación en Métodos de Producción de Software (ProS) in 2009. Here he studied the process of loading the genomic database (HGDB) that came to be as a result of the Conceptual Model of the Human Genome (CSHG), primarily focusing on the case study of the NF1 gene. He obtained his BSc in Informatics in july 2010 at the Technical University of Valencia for his merits in this field. For the creation of his Bachelor thesis he collaborated on a strong basis with the researchers in the same domain: Matthijs van der Kroon, Ana M. Levin and Óscar Pastor. The fruits of this collaboration and the realized labor, are solidified in the here presented article.

Ana Levin obtained her MsC in Biology by the Universidad de Valencia (Spain) in 1999. After her graduation she moved to The Netherlands and worked as a Junior Scientist at TNO Voeding and WCFS in Zeist (NL). In 2007 she obtained her PhD for the work "Differentiation in colonies of *Aspergillus niger*" promoted by Prof. Dr. Han Wösten at Utrecht University, Molecular Microbiology Department. That same year she moved back to Spain. From 2008, she works as Project Manager of the Genoma team at Centro de Investigación en Métodos de Producción de Software, a research centre of Universidad Politécnica de Valencia. Ana Levin has several publications in international journals, book chapters, conferences and an industrial patent from her collaboration with the multinational DSM during her PhD.

Óscar Pastor is full Professor and Director of the "Centro de Investigación en Métodos de Producción de Software (PROS)" at the Universidad Politécnica de Valencia (Spain). He received his Ph.D. in 1992. He was a researcher at HP Labs, Bristol, UK. He has published more than two hundred research papers in conference proceedings, journals and books, received numerous research grants from public institutions and private industry, and been keynote speaker at several conferences and workshops. Chair of the ER Steering Committee, and member of the SC of conferences as CAiSE, ICWE, CIbSE or RCIS, his research activities focus on conceptual modeling, web engineering, requirements engineering, information systems, and model-based software production. He created the object-oriented, formal specification language OASIS and the corresponding software production method OO-METHOD. He led the research and development underlying CARE Technologies that was formed in 1996. CARE Technologies has created an advanced MDA-based Conceptual Model Compiler called OlivaNova, a tool that produces a final software product starting from a conceptual schema that represents system requirements. He is currently leading a multidisciplinary project linking Information Systems and Bioinformatics notions, oriented to designing and implementing tools for Conceptual Modeling-based interpretation of the Human Genome information.

Sjaak Brinkkemper is full professor of organisation and information at the Department of Information and Computing Sciences of the Utrecht University, the Netherlands. He leads a group of about thirty researchers specialized in product software development and entrepreneurship. The main research themes of the group are methodology of product software development, implementation and adoption, and business-economic aspects of the product software industry.