Diagen: A Model-Driven Framework for Integrating Bioinformatic Tools

Maria José Villanueva, Francisco Valverde, Ana M. Levín,
and Oscar Pastor Lopez

Software Production Methods Research Center
Universitat Politècnica de València
Camino de Vera S/N 46022, Valencia, Spain
{mvillanueva,fvalverde,alevin,opastor}@pros.upv.es

Abstract. Nowadays, the diagnosis of disease based on genomic information is feasible by searching genetic variations on DNA sequences. However, geneticists struggle with bioinformatic tools that are supposed to simplify DNA sequence analysis. As a universal tool to support every requirement is far from be implemented, geneticists themselves must solve the data exchange among several tools. Due to the fact that there are no standards to support this integration task, it must be managed in every analysis. This paper addresses this integration by means of a model-driven framework. The Diagen framework is a software implementation based on conceptual modeling principles that formalizes data exchange and simplifies bioinformatic tool integration. First, we analyze how conceptual modeling can be used to deal with data exchange among tools. Then, the presented framework is used to search for variations on the BRCA2 gene using real DNA samples and a set of specific bioinformatic tools.

Keywords: Model-Driven Development, Tool Integration, DNA sequence analysis.

1 Introduction

Recent genetic discoveries have opened the door to personalized disease diagnosis based on DNA sequence analysis. A DNA sample, extracted from i.e. blood, is treated by sequencing machines and, then, a DNA sequence is obtained. Afterwards, the resulting sequence is compared against a reference sequence in order to obtain the differences between them. Geneticists name these differences as genetic variations and use them to assess their effects in the humans' health. Nowadays, it is possible to predict the risk of getting a certain disease by searching for specific genetic variations on the DNA sequence [1].

Geneticists perform DNA sequence analysis aided by bioinformatic tools. Even though these tools are functional and useful for reducing time and complexity, none of them completely fulfill all the geneticists' requirements [2]. As a consequence, geneticists are forced to use several tools in order to gather all the functionality and, eventually, accomplish the complete DNA sequence analysis.

S. Nurcan (Ed.): CAiSE Forum 2011, LNBIP 107, pp. 49–63, 2012.

One important issue regarding these tools is that data exchange among them is required. The problem lies in the fact that each of these tools is isolated and uses its own data format to report the computed information. For this reason, data exchange among tools is a non-trivial task that geneticists must address in each analysis. An example procedure to fulfill this task is described as follows:

1. Data exportation: After the tool task is completed, the results are exported into a file following one of the available export formats.
2. Format comprehension: It is required to understand the semantics and the syntax of the tool-specific data formats. The concepts related with the information that has to be exchanged should be identified in both involved formats.
3. Format manipulation: A relation between both formats is established for each required concept. Then, all the data is translated from the source format into the target format.
4. Data importation: Once all the information is expressed using one of the available import formats, it can be imported in the target tool.

As geneticists usually lack Software Engineering knowledge, most of them perform this task manually or by means of programming scripts. Although these specific scripts are useful in solving minor problems, they are far from being compliant with good practices of Software Engineering. The implemented scripts to support data exchange are often coupled solutions that integrate only two specific tools. In the end, these solutions cannot be reused and compromise the geneticists flexibility for using other tools.

In order to achieve a higher effectiveness in the execution of DNA sequence analysis tasks, the availability of structured genomic information systems, and software tools to exploit them, is very important. The work presented in this paper is part of the Diagen Project, a research project created to address these goals. The Diagen Project is a collaborative project among experts of different domains: 1) Geneticists from the Instituto de Médicina Génomica (IMEGEN), experts in DNA sequence analysis; 2) Software Engineers from the Centro de Investigación en Métodos de Producción de Software (ProS), experts in the application of Model-Driven Software Development in different domains; and 3) Computer Scientists from the Grid y Computación de Altas Prestaciones (GryCap), experts in parallel computation and supercomputation.

The first objective of the project (Figure 1) was the design of the Conceptual Schema of the Human Genome (CSHG) [3], a conceptual schema whose main aim is the formalization of the concepts related with the human genome.

The second objective of the project was the development of an Information System based on the CSHG, the Human Genome Database (HGDB). This database has been populated with structured genomic information by means of loading routines that understand, transform and load the genetic data obtained from different heterogeneous genomic databases.

The last objective of the project, explained in this paper, was the identification of the effectiveness problems of current software tools available for DNA sequence

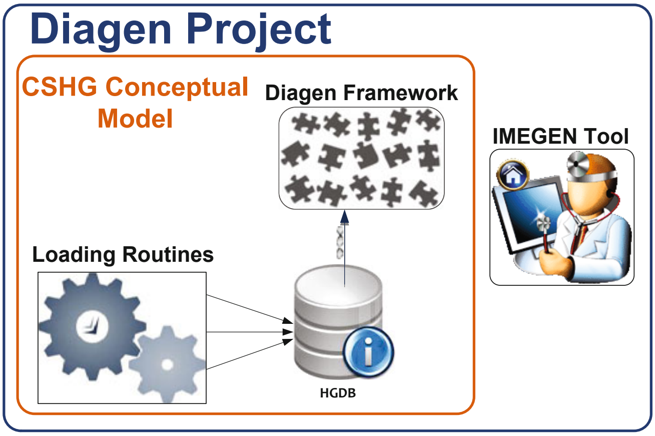

Fig. 1. Diagen Research Project

analysis and the elaboration of a tool that overcomes these problems. This study revealed that the problems were not in the tools functionality itself, but in the lack of support for data exchange among them.

As a solution, this paper proposes the application of conceptual modeling to develop a model-driven framework that formalizes data exchange and simplifies tool integration. The Diagen framework gathers the required conceptual models with the aim to establish a common domain specification for expressing genetic data precisely and unambiguously. Moreover, it provides a suitable mechanism to manage data flow among tools to achieve the tool integration. As a proof of concept, the proposed framework integrates several tools that are used by IMEGEN geneticists in their daily routine to search for genetic variations, using real DNA samples of the BRCA2 gene (a gene related to Breast Cancer).

The rest of the paper is organized as follows: Section 2 presents a brief summary of other proposed solutions to solve the tool integration problems in DNA sequence analysis. Section 3 explains the proposed model-driven framework for integrating bioinformatic tools. Section 4 presents how the framework is used for disease diagnosis support using samples of the gene BRCA2 and a set of bioinformatic tools. And finally, section 5 presents the conclusions and future work.

2 Related Work

Several works have attempted to overcome current DNA sequence analysis tool issues. These proposals follow two different approaches.

Several sequence file formats for expressing bioinformatic tools results have emerged. Examples of these formats are: 1) Variant calling formats, such as the Variant Call Format (VCF) proposed for the 1000 Genomes Project [4]; and,

2) Alignment results formats, such as the Genome Variation Format (GVF) [5], which provides a textual format using the Sequence Ontology [6] or the Sequence Alignment/Map Format (SAM) [7], which provides a compressed textual representation of read alignments against a reference. The SAM authors already proposed a set of utilities, named SAM tools, to provide post-procesing functionality to the SAM format (such as viewers). The SAM format expresses large amounts of alignments in small size and allows an efficient access to them. However, the interpretation of this format is not easy and requires a good knowledge of the format in order to identify the configuration and codification of their different fields.

All these formats have been defined for the purpose to provide interoperability among different DNA sequence analysis tools. The implementation of decoupled data exchange mechanisms is feasible using any of the above examples as a standard format. However, their main drawbacks are the complexity of each textual format and the mandatory implementation of a low-level mechanism to extract the data. As a consequence, none of them have become a widely applied standard and are only used in the research context where they have been proposed.

Several bioinformatic development frameworks have also been implemented that address the integration issue. Some examples of these frameworks are Biojava [8], BioPython [9], or BioPerl [10]. These frameworks provide an API that supports common functionality for DNA analysis tasks. These frameworks have been defined to provide geneticists with the freedom to implement their personalized tools. All of them allow geneticists to develop programs written in different programing languages (Java, Python or Perl) offering implemented methods that can be called inside their programs. Additionally, as the sequencing domain lacks of standard nomenclature to express the output results, they provide several format conversion operations to transform file formats among different tools. However, although geneticists are able to customize their programs for DNA sequence analysis, they still have to worry about low-level programming details and integration issues.

Another example of a development framework is the Taverna Tool [11], a framework for the design, edition and execution of workflows based on the integration of web services. Taverna is specially focused on the biological domain, providing the interoperability with biological resources such as myExperiment or the BioCatalogue. The Taverna tool is a very well-known tool among biologist to design in-silico experiments. However, geneticists still have to worry about the inputs and outputs that each web service produces, that is, they must map which output is required as input for the next task.

Concerning the academic environment, some approaches have considered the use of conceptual modeling to improve the quality of software tools for DNA sequence analysis. On the one hand, the framework Pierre [12] is used for the partial generation of user interfaces for browsing through genomic repositories.This automatic generation is based on the specification and composition of different genomic services. On the other hand, the framework MEMOPS [13] presents an architecture for the retrieval and storage of biological information. This framework uses UML as modeling language in order to define a data model that is used to generate automatically the documentation, the programing interfaces and the storage code. Both approaches follow model-driven principles in order to create usable interfaces, but they addressed how to present and retrieve genomic information, instead of processing tasks to obtain additional analytical information or integration with another bioinformatic tools.

The Diagen framework allows the integration of bioinformatic tools in order to design computational processes to obtain new genetic knowledge from their data. Moreover, the framework is designed to implement high-level services to avoid specific tasks and input/output formats issues.

3 An Integrative Framework for Bioinformatics

This work presents a model-driven framework for the integration of DNA sequence analysis tools and the retrieval of genetic information. Diagen is classified as a model-driven framework because each of its components (classes, data entities, and operations) is a projection of the Conceptual Schema of the Human Genome (CSHG).

The CSHG is a conceptual model created in close collaboration with geneticists, where biological concepts related to the human genome have been precisely addressed and defined. The CSHG has been designed through the definition of four views: 1) The structural view, which defines the internal composition of different genetic elements; 2) The variation view, which defines the knowledge related to the differences in DNA sequences among individuals; 3) The transcription view, which defines the functional effects through the production of proteins; and 4) The pathway view, which defines the metabolic reactions that occur inside the cell. These views contain the concepts related to different perspectives of the human genome properties.

The Diagen framework uses three views (structural, transcription and variation) of this conceptual model to support the following DNA sequence analysis tasks (Figure 2):

1. Sequence Treatment (T1): Due to technological limitations of sequencing techniques, the process is fragmented and error prone. As a consequence, the final DNA sequence is made up of small and redundant fragments that have to be assembled by a specific program using a reference sequence; eventually a consensus sequence is derived.
2. Sequence Alignment (T2): The resulting DNA sequence is aligned to a reference sequence in order to determine the differences between them. Each difference is classified according to the change that has occurred in the DNA sequence.
3. Variation Knowledge (T3): Each difference is characterized as a genetic variation. Moreover, using data gathered in genomic databases, each variation is associated with complementary information and reported if it is associated to a disease.

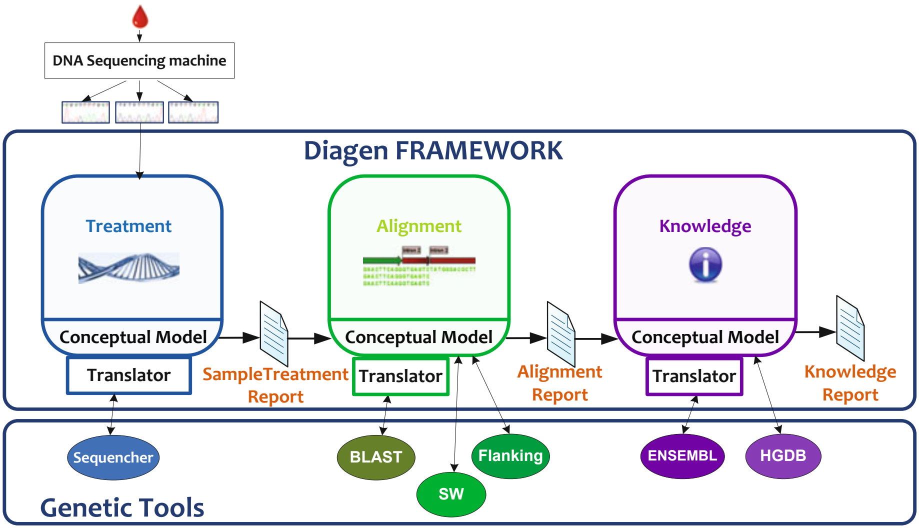

Fig. 2. General View of the Framework

In order to accomplish a complete DNA sequence analysis, these tasks must be executed sequentially and exchange information among them. Taking into account that data exchange is required when a tool calculates data that another tool requires, it can be assumed that both tools must share a set of common concepts. Therefore, it is possible to define a conceptual model that represents those shared concepts and establishes well-defined boundaries and vocabularies. These conceptual models are called *Reports* in the framework context, as they gather all the concepts related to the information that is reported in each task.

Diagen establishes a common context to guide data exchange among tools defining a conceptual model for each task transition. Focusing on the DNA sequence analysis process, three conceptual models have been defined:

1. The Sample Treatment Report conceptual model (Figure 3): This conceptual model defines all the concepts related to the reconstructed sequence obtained after the sequence treatment task (T1). This sequence is analyzed in the sequence alignment task (T2).
   The entity *Gene* represents the DNA region that is sequenced by geneticists. A *Gene* is identified by the *id*, that is a well known term in the genetics community. The attribute *transcript_id* identifies the transcription sequence that has been used to determine the set of exons to be sequenced. On the one hand, a *Gene* has a set of *Exons*, the regions of the genes that codify for proteins. An *Exon* is identified by a number *num*, that locates its position inside the *Gene*. The attribute *consensus* contains the validated DNA sequence of the *Exon* after the sample treatment process. On the other hand, a *Gene* is made up by *Segments* that are the fragments obtained by the sequencing machines. A *Segment* is identified by the attribute *id*, that differ-

entiates one from another. The attribute *electropherogram* contains the result obtained directly from sequencing machines and the attribute *sequence* contains the interpretation of this *electropherogram* into nucleotides (A,T,G,C). The *startpos* and *endpos* indicate the range of the fragment inside the *Gene*. Finally, the *Reference* entity represents the sequence used by geneticists as a reference to carry out different analysis. In this task, the *Reference* is used for the assembly of all the *Segments* of the *Gene* in order to obtain the *consensus* sequences. The attribute *id* identifies the entity using a well known nomenclature in the genetics community. The attribute *sequence* contains a general sequence, that is a computed representation of all the sequences of the human beings. This conceptual model uses a few concepts of the structural and transcription views, such as the *Gene* entity, the *Exon* entity, or the *transcript_id* attribute. Other concepts are related with technological details, such as the *Segment* entity or the attribute *electropherogram*.

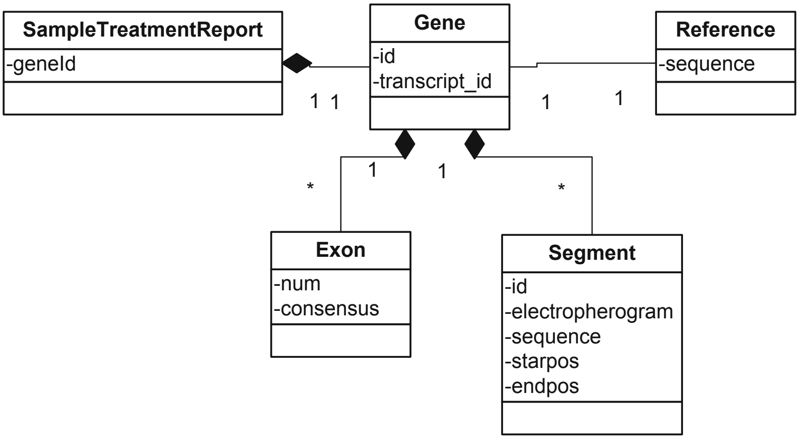

Fig. 3. Sample Treatment Report Conceptual Model

2. The Alignment Report conceptual model (introduced in [14], Figure 4): This conceptual model defines all the concepts related to the differences found in the sequence alignment task (T2) to be characterized in the variation knowledge task (T3).

   Gene and *Reference* are the entities that are compared in the Alignment task. A *Gene* represents the DNA sequence that geneticists want to analyze, and is the generalization of the Gene and Exon entities from the Sample Treatment conceptual model. The *Reference* is the reference sequence used for comparison and is the same as the one described in the Sample Treatment conceptual model. As geneticists are only interested in differences, the Alignment Report contains a list of the *Differences* found. A *Difference* is identified by *startPos* and *endPos* that locates it according to the *Reference*' sequence. Moreover, a *Difference* may be identified as well using their flanking sequences: 20 nucleotides that are delimiting both sides of the difference, the *fsright* and *fsleft* attributes. The attribute *isHeterozygous* is a biological concept that indicates if the difference has occurred in one (homozygous) or both DNA strands (heterozygous). A *Difference* is categorized in three types: *Insertion*, *Deletion* or *Substitution* if some *characters* have been introduced, deleted or substituted in the sequence in comparison with the reference. This conceptual model uses a few concepts of the structural view, such as the entities *Gene* and *Reference.* Other concepts are related with string sequences comparison, such as the entities *Insertion*, *Deletion* or *Substitution*, and the attribute *characters.*

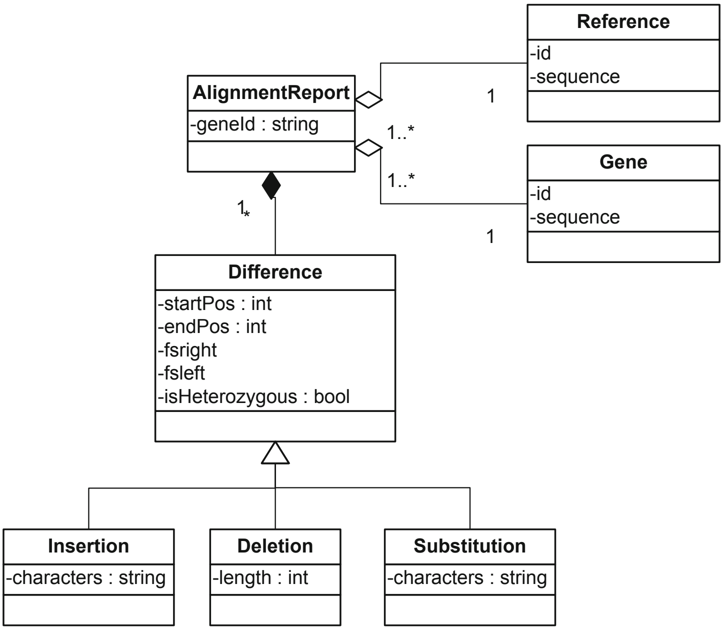

Fig. 4. Alignment Report Conceptual Model

3. The Knowledge Report conceptual model (Figure 5): This conceptual model defines all the concepts related to the characterized variations in the knowledge task (T3) to be used for another task, for example, to visualize a diagnosis report.

   Reference and *Gene* are the entities that have been analyzed in order to create a disease diagnosis report. From this analysis a list of genetic variations have been detected. Each difference described in the Alignment conceptual model becomes a *Variation* when it has some genetic knowledge associated. A *Variation* is characterized by the same attributes than the differences: *startPos* and *endPos* according to the *Reference*; the *fsright* and the *fsleft* flanking sequences of the *Variation*; and the *isHeterozygous* attribute indicating the affected strands of the variation. Additionally, a *Variation* is characterized by the attributes *HGVSGenomic*, *HGVSCoding* and *HGVSProtein* that represent the standard nomenclature for expressing genetic variations [15]. A *Variation* can be categorized in three types: *Insertion*, *Deletion* or *Indel* if some *nucleotides* have been inserted, deleted or changed. The entity *Knowledge* retrieves the additional information that has been found in genetic databases.The attribute *phenotype* expresses the external feature associated to the variation. The attribute *isSNP* means that is a common variation present in at least 2% of the population. The attribute *certainty* offers an assessment of the veracity of the information. Finally, the *source* refers to the data origin. Sometimes, this information is claimed to be true by a research publication. The entity *Bibliography* shows the information about this publication: the *title* of the paper, the *authors* involved, the *abstract* or summary of the contents, the media of *publication* (for example, a conference or a journal), and the *URL* where it can be found. This conceptual model uses a few concepts of the structural view, such as the entities *Gene* and *Reference*; the variation view, such as the *Variation* entity, or the attributes *phenotype* and *isSNP*; and the transcription view, such as the attributes *HGVScoding* and *HGVSprotein.*

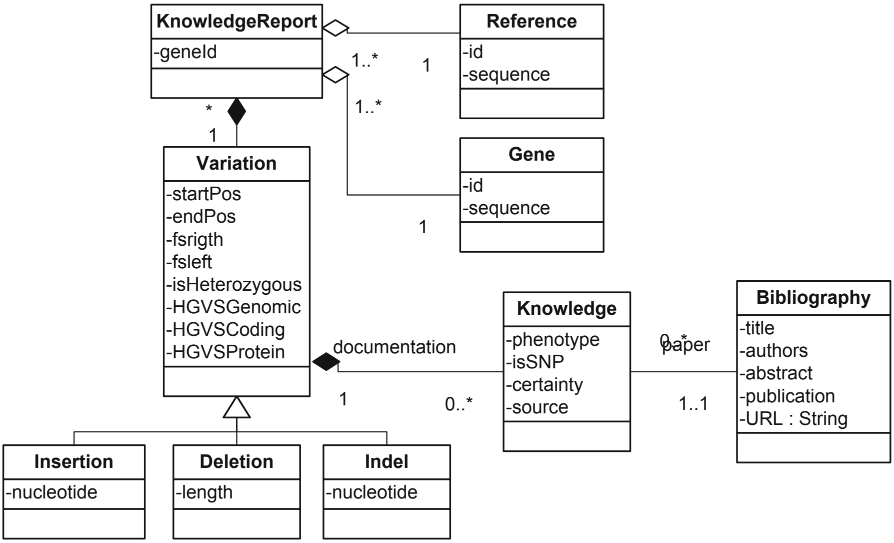

Fig. 5. Variation Knowledge Conceptual Model

Data exchange among tools that perform these tasks, usually requires the implementation of a translation mechanism to understand each other. In that case, data expressed in a concrete format needs to be translated into a different format. The implementation of this translation mechanism is a highly consuming task, and the solution is not flexible enough if some tool has to be introduced or changed in the process.

Instead, Diagen avoids coupled implementations thanks to the use of conceptual models, in a higher level of abstraction. A tool can be completely integrated in the process incorporating a simple translator in the framework. This translator is easier to implement since it only requires establishing the relationships between the output and the conceptual model. This translator should express the outputs of the tool in terms of the underlying conceptual model. Hence, the implementation of this translator is completely independent of other tools and formats.

The Diagen framework has been implemented using the Java language. Additionally, each conceptual model involved in data exchange has software correspondence with a set of Java classes and a XML representation. In order to manage both representations (Java and XML) JAXB (Java Architecture for XML bindings) [16] has been used. This is a specific API that allows Java objects to be parsed in XML data and vice versa. The implementation of both configurations provides a better flexibility to use each of the components into another environments.

Each task that is supported by the framework has been implemented to be independent from the others, therefore it can be used separately. Thanks to this modularity, it is possible, for example, to use the alignment task in another environment (Figure 6). In this case, the input data must be provided in terms of the input conceptual model (Sample Treatment Report) and the output report must be read in terms of the output conceptual model (Alignment Report). Both conceptual models can be expressed using the Java language or the XML representation. In the case of XML, the JAXB framework manages to the correspondence with the Java language.

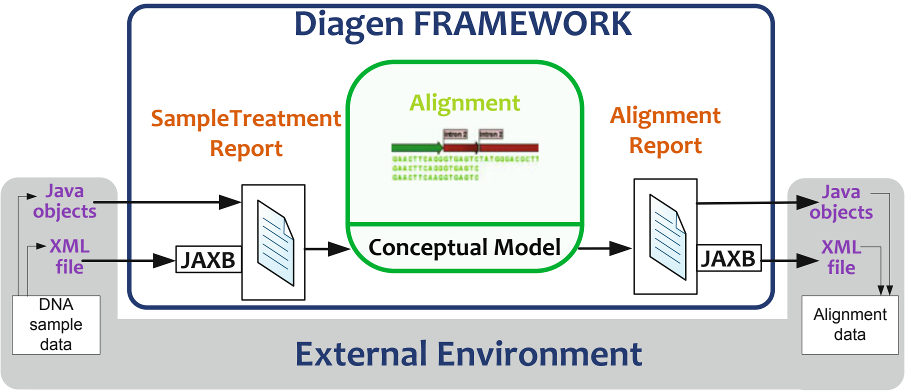

Fig. 6. Integration Mechanism for Alignment Task

4 Using Diagen for Disease Diagnosis: The BRCA2 Case

As a proof of concept, the framework has been used to develop a prototype for disease diagnosis of Breast Cancer. The prototype has been designed as a web application in order to offer geneticists a higher flexibility and avoid them installation issues. This specific framework configuration integrates several bioinformatic tools that are used daily by the geneticists of IMEGEN.

The prototype architecture consists of four layers (Figure 7):

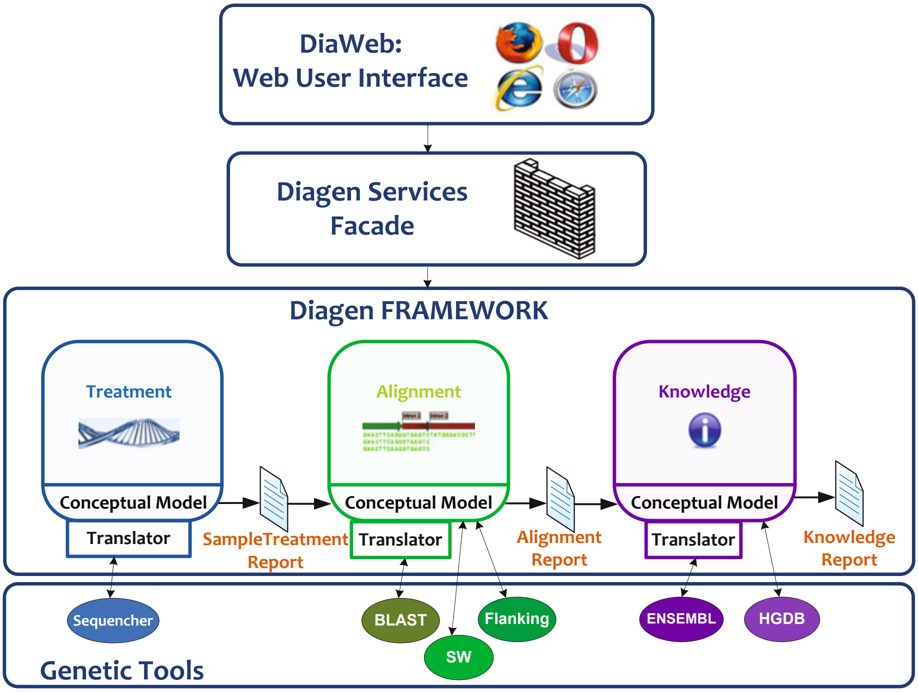

Fig. 7. IMEGEN Prototype using Diagen

1. DiaWeb: This layer is responsible of the interaction with geneticists through several web pages that capture data (such as the sequencing machine output or the selection of the algorithm for aligning the sample) and offer a diagnosis result.
2. Diagen Facade: This layer offers to upper layers the Diagen functionality through an API. This interface is defined as several methods that encapsulate the available services of Diagen.
3. Diagen Framework: This layer includes the presented framework and the tool translators. These translators are responsible of translating the tool format into the conceptual model, and vice versa.

4. Genetic Tools: This layer collects the tools that geneticists from the Diagen research project usually use in order to accomplish the DNA sequence analysis. It also includes some additional tools designed in the context of this project with the aim of improving the efficiency of these tasks. Hence, the framework has been applied to integrate:
   (a) Sequence treatment task: The Sequencher tool [17] is used to rebuild the sequence from the segments provided by sequencing machines. A translator, which expresses the calculated consensus sequences in terms of the Sample Treatment Report conceptual model, has been implemented and incorporated inside the framework.
   (b) Sequence Alignment task: The implementation of the BLAST algorithm from NCBI [18] is used to search for differences in the sequence. In this case, the incorporated translator expresses the found differences in terms of the Alignment Report conceptual model. Due to the fact that BLAST is an heuristic algorithm, it does not always detect the best solution to the alignment problem. Hence, an algorithm based on the Smith-Watterman has been implemented (SW Tool). This algorithm, although is more accurate than BLAST, it is still not efficient enough to be used in practice. For these reasons, another alignment approach has been proposed: to look for known-variations in the sequence by aligning the flanking sequences of each variation against the DNA sequence (Flanking Tool). Regarding both tools, the integration does not required the implementation of an additional translator because both use the proposed conceptual model and their results are already expressed using the Alignment Report conceptual model.
   (c) Variation Knowledge task: Variation characterization is performed manually by geneticists searching in several databases. As a better solution, the Diagen framework integrates two mechanisms for genetic knowledge retrieval. The first mechanism obtains some genetic data (such as structural information about genes or transcription data required to calculate the HGVS notation) from the ENSEMBL database [19]. In this case, a translator has been included to express this data using the conceptual model. However, the information retrieved by ENSEMBL is not enough to execute the complete variation knowledge task. The second mechanism retrieves genetic information (such as phenotype information or bibliographic references) from the HGDB database [20]. As the HGDB gathers information from different genomic repositories that geneticists usually check, it is possible to execute the variation knowledge task successfully. Moreover, as HGDB is based on the Conceptual Schema of the Human Genome (CSHG) [3] it does not require an additional translator.

The prototype supports the three defined tasks to perform a DNA sequence analysis. As a result, it retrieves a personalized report containing the genetic variations and the potential diseases according to an individual sample.

The use of the framework provides two main advantages: 1) The development time of specific translators decreases because the core functionality can be reused among them; 2) The framework also provides common functionality for managing DNA sequences (comparison, retrieval of reference sequences, nomenclature and so on). Moreover, regarding the use of the IMEGEN tool itself the main advantages are: 1) A reduction in the efforts needed for data exchange among tools, as the Diagen framework manages data flow transparently for geneticists; 2) The elimination of the need to search for variation data in the huge set of databases spread around the Web, as the information system HGDB gathers this data in a structured way and the genetic data retrieval is easily performed; and 3) A decrease in the execution time, as manual data flow and manual search are avoided.

The prototype has been tested with real samples of the BRCA2 gene (Table 1). The test was carried out analyzing the BRCA2 gene sample from ten different patients (P1-P10). For each patient, the table shows the number of variations characterized by IMEGEN, the number of variations characterized by Diagen, and the accuracy that Diagen offers compared with the IMEGEN manual process. IMEGEN performs the analysis in approximately four hours (depending on the success achieved while searching for a difference in the genetic repositories).

The preliminary test showed that Diagen offers the results almost instantly and with an accuracy rate of between 60-90%. It is also important to emphasize that the variations not characterized by Diagen were always the same variations (7 variations in total) that appeared repeatedly in all the analyses.

Table 1. Preliminary BRCA2 tests

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Characterized Var. IMEGEN	7	10	8	8	8	13	9	10	9	8
Characterized Var. Diagen	6	6	7	6	5	8	6	7	6	5
Accuracy rate %	86	60	88	75	63	62	67	70	67	63

5 Conclusions and Future Work

This work proposes a model-driven framework that is based on a well-defined conceptual model of the human genome in order to address DNA sequence analysis. As a proof of concept, the Diagen framework is configured for the development of a disease diagnosis support and is tested by means of real DNA samples of the BRCA2 gene.

We have realized that the available tools actually accomplish some of the geneticists' goals. The problem lies in the fact that geneticists' activities, specifically in the DNA analysis domain, lack standard methodologies, well-defined processes, fixed vocabularies, and unified knowledge sources. As a consequence, the execution of a DNA sequence analysis cannot be performed efficiently or without geneticists' intervention.

The solution to these problems is not to reinvent new DNA sequence analysis tools but to integrate the most suitable tools according to geneticists' needs. The presented framework applies conceptual modeling to integrate different bioinformatic tools and to provide a common context to exchange data with each other. The main advantage of the presented framework, over other integration approaches, is that Diagen is a high-level abstraction framework that provides concise and significant tasks to geneticists (such as "sequence treatment", "alignment" or "variation knowledge retrieval") instead of low-level tasks (such as "run Blast algorithm", "translate format to Fasta", or "obtain HGVS nomenclature"). Moreover, with this framework, geneticists can perform a DNA sequence analysis and forget about the data formats of different tools.

As genetics is a very innovative field that is constantly evolving with new discoveries, all concepts must be well-defined without ambiguity. The CSHG was the first step in order to establish the domain specification of the different human genome concepts. Thanks to this conceptual schema is possible to specify the concepts related to DNA sequence analysis. Thus, the involved concepts in the different tasks have been precisely formalized. As a consequence, it is easier to adapt the tasks to changes or to support new concepts.

The preliminary results are promising, but there is room for improvement. The low accuracy detected is because the missed variations were not described in the integrated sources. As these sources are constantly improving, it is expected that future versions of the IMEGEN prototype will solve these issues.

As future work, the framework will be extended to support other bioinformatic tasks. The main goal of this extension is to design a complete framework that supports other functionality besides DNA sequence variation analysis.

Additionally, the next step is to apply the service-oriented paradigm to provide a more flexible development environment. With this approach, geneticists could select only the required functionality, defined as services, and easily create a personalized tool.

Acknowledgments. Thanks to the Instituto de Medicina Genómica (IMEGEN, http://www.imegen.es) for its support and the provision of data. This work has been developed with the support of MICINN under the FPU grant AP2010-1985 and the project PROS-Req (TIN2010-19130-C02-02), and co-financed with ERDF.

References

1. Hamburg, M.A., Collins, F.S.: The Path to Personalized Medicine. New England Journal of Medicine 363(4), 301–304 (2010)
2. Rusk, N.: Focus on Next-Generation Sequencing Data Analysis. Nature Methods 6(11s), S1 (2009)
3. Pastor, O., Levin, A.M., Celma, M., Casamayor, J.C., Virrueta, A., Eraso, L.E.: Model-Based Engineering Applied to the Interpretation of the Human Genome. In: Kaschek, R., Delcambre, L. (eds.) The Evolution of Conceptual Modeling. LNCS, vol. 6520, pp. 306–330. Springer, Heidelberg (2011)
4. Nayanah, S.: 1000 Genomes Project. Nat. Biotech. 26(3), 256 (2008)

5. Reese, M.G., et al.: A Standard Variation File Format for Human Genome Sequences. Genome Biology 11(8), R88 (2010)
6. Eilbeck, K., et al.: The Sequence Ontology: A Tool for the Unification of Genome Annotations. Genome Biology 6(5), R44 (2005)
7. Li, H., et al.: The Sequence Alignment/Map Format and SAMtools. Bioinformatics 25(16), 2078–2079 (2009)
8. Holland, R.C.G., et al.: BioJava: An Open-Source Framework for Bioinformatics. Bioinformatics 24(18), 2096–2097 (2008)
9. Cock, P., et al.: Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. Bioinformatics 25(11), 1422–1423 (2009)
10. Stajich, J.E., et al.: The Bioperl Toolkit: Perl Modules for the Life Sciences. Genome Research 12(10), 1611–1618 (2002)
11. Hull, D., et al.: Taverna: a tool for building and running workflows of services. Nucleic Acids Research 34(Web Server issue), 729–732 (2006)
12. Garwood, K., et al.: Model-driven user interfaces for bioinformatics data resources: regenerating the wheel as an alternative to reinventing it. BMC Bioinformatics 7, 532 (2006)
13. Fogh, R.H., et al.: MEMOPS: Data modelling and automatic code generation. Journal of Integrative Bioinformatics 7 (2010)
14. Villanueva, M.J., Valverde, F., Pastor, O.: Applying Conceptual Modeling to Alignment Tools: One Step towards the Automation of DNA Sequence Analysis. Bioinformatics 2011 (2011)
15. Dunnen, J.T., Antonarakis, S.E.: Mutation nomenclature extensions and suggestions to describe complex mutations: A discussion. Human Mutation 15, 7–12 (2000)
16. Ort, E., Mehta, B.: Java Architecture for XML Binding (JAXB). Technical Report Sun Developer Network (2003)
17. Curtis, P., Bromberg, C., Cash, H., Goebel, J.C.: Sequencher, Gene Codes Corporation, Ann Arbor, Michigan (1995)
18. NCBI BLAST (Basic Local Alignment Search Tool), `http://blast.ncbi.nlm.nih.gov`
19. Hubbard, T., et al.: The ENSEMBL Genome Database Project. Nucleic Acids Research 30(1), 38–41 (2002)
20. Pastor, O., et al.: Enforcing Conceptual Modeling to Improve the Understanding of Human Genome. In: Research Challenges in Information Science (RCIS), pp. 85–92 (2010)