CHALMERS

# MembraneLogic
## A database to couple membrane protein sequences to variation and disease

*Master of Science Thesis*

IMAN POUYA

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden,  March 2011

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.
The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

MembraneLogic
A database to couple membrane protein sequences to variation and disease

Iman Pouya

©IMAN POUYA, March 2011.

Examiner: GRAHAM KEMP

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:
An Apple with a protein sequence carved into it.
The picture represents how we try to turn protein sequence data into knowledge.

Department of Computer Science and Engineering
Göteborg, Sweden March 2011

## Acknowledgments

First I would like to thank Erik Lindahl and Arne Elofsson, my supervisors at CBR for guiding me and pointing me to the right direction during my thesis work.

I would also like to thank Sara Light for providing me specific bioinformatics knowledge and all the feedback throughout my work.

Also a big thanks to Aron Hennerdal who helped me with the membrane protein prediction part in this thesis.

Finally thanks to all colleagues at CBR who introduced me to the field of bioinformatics.

## Abstract

Bioinformatics data is growing faster than exponentially. This large amount of data has opened up many possibilities but it also poses many challenges as it is a tedious task to manually analyze this data.

25% of our genes translates to membrane proteins and they contain 60% of the drug targets. A genetic variation can alter the function of a protein, this can sometimes lead to a disease. Knowing where in the protein structure a disease causing variation occurs will aid drug designers in the drug discovery process.

MembraneLogic is a database that through its automated data gathering process identifies the membrane proteins by prediction and links them to known SNPs and their relation to disease.

The data it uses comes from NCBI Ensembl and PDB and the main prediction method used to identify membrane proteins is SCAMPI.

Its web interface provides functionality to search for membrane proteins and diseases and see what variations are related to disease ,both in sequence and structure.

The application is implemented with JAVA platform related tools and frameworks such as Grails, Spring and Hibernate.

# Table of Contents

# 1. Introduction

## 1.1. Motivation

Membrane proteins constitute 25% of our genes, 60% of the drug targets are towards them. Being able to identify what mutations exist and their linkage to disease both in sequence and structure will be a great aid in drug design and further research.
At the present there exists more than 1000 online resources that provide bioinformatics data. The resources provide different types of data such as nucleotide sequences, protein sequences, protein structures, disease data, data related to human genome variation etc. The bioinformatics data is growing at a high pace, as an example the data growth of base pairs in Genbank is growing faster than exponentially and as the data will continue to grow the need to analyze this in a more systematic manner will grow.
Linking the different types of data is a tedious task that requires knowledge of each individual datatype and different skill levels depending on the data source. Therefore it would be highly useful to create a tool that automatically gathers protein sequences and structures, and links these with variations and known diseases.
This tool could then be used by experimentalists and drug designers to obtain knowledge about proteins linked to human diseases.

## 1.2. Project goals

The main project goal is to create a program that consists of an automated pipeline that can gather bioinformatics data related to membrane proteins. The pipeline will gather protein sequences and structures, disease information, sequence variation data and link these together in a database.
The program will provide a web interface where a user can search for proteins and diseases and be able to see where in the structure and sequence a variation occurs and whether it is linked to a disease.

The outline of the rest of the report is as follows: Chapter 2 gives some background which will aid in understanding the problem domain. Chapter 3 explains the proposed solution and what technologies are used to achieve this. Chapter 4 discusses the results, Chapter 5 explains some of the future work that can be done and Chapter 6 concludes the report.

# 2. Background

## 2.1. DNA

A Genome contain the code that is needed to construct an organism. It contain the biological information that is needed to construct life.
It is divided into subunits called genes. Most genes contain the code that leads to the synthesis of a protein, the functional units of a cell.

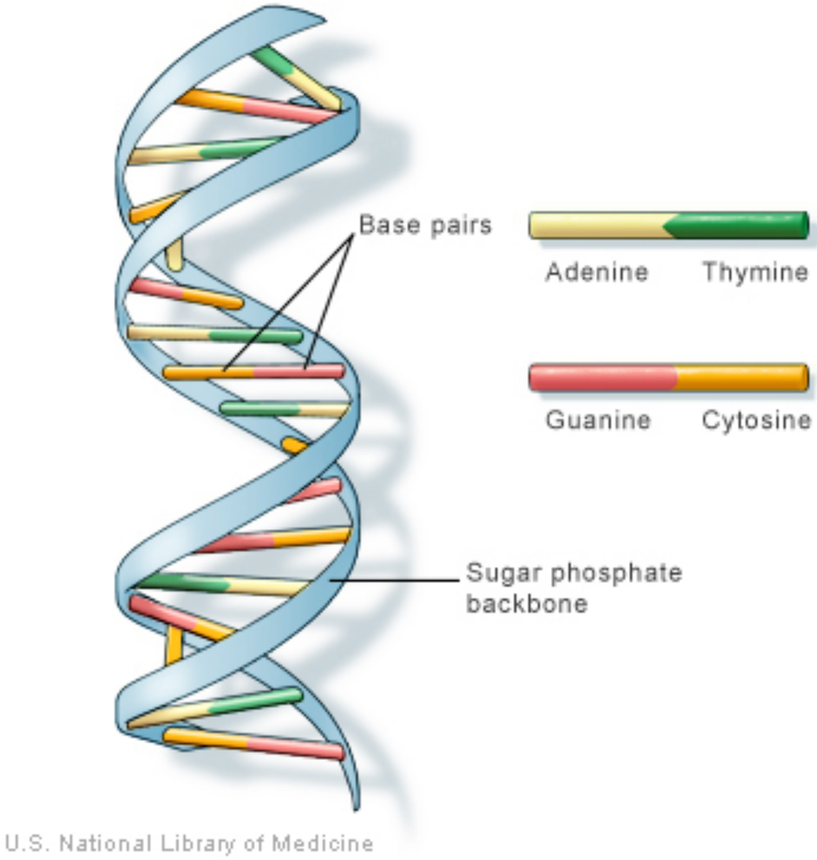


**Figure 2.1 from** structure of DNA from http://ghr.nlm.nih.gov/handbook/illustrations/dnastructure.jpg

The human genome is made of DNA (deoxiribo-nucleic-acid), a double helix structured molecule built up from subunits called nucleotides. Each nucleotide is built up by a sugar phosphate backbone and a base, the bases are Adenine (A), Thymine (T), Guanine (G) and Cytosine (C) [Figure 2.1]. A full human genome sequence consists of roughly 3 billion base pairs, that translates to 3Gb of data [1].

### 2.2.The Human genome project (HUGO)

The Hugo project started in 1990. The main goal was to obtain the complete sequence of the human genome.
Originally the project was expected to end in 2020. Craig Venter a scientist at NIH during the project start proposed a method called shotgun sequencing that would speed up the sequencing process, however other scientists felt that this method was not reliable. Venter went on and started Celera genomics in 1998 with private funding.
With the shotgun sequencing method Celera was able to sequence the genome at a faster pace than the HUGO project. This competition led to a first draft of the genome sequence being published by both projects in 2001 [2] [3] and the full sequence being published in 2003 [4].
A key finding was the number of protein coding genes in the human genome. The estimated number was 30,000-40,000 after the first draft and it came down to 20,000-25,000 after the publication of the full sequence. Before this it was estimated that the human genome contains 10 0,000 genes. This large amount was expected since one thought that humans are much more complex than other species, in fact we don't have that many more genes than other species. For example a worm has roughly 20,000 genes.
The HUGO project gathered information that opened up new research possibilities such as analyzing genetic variations, find the functions of the various genes, comparing differences between species, etc.

### 2.3.The genome of each individual varies

More than 99% of the human DNA sequences are the same but each individual has small variations in their sequence. These variations are usually referred to as mutations, some of the causes of mutation are caused by different events such as radiation,mutagenic chemicals or during the DNA replication process.
Single Nucleotide Polymorphisms (SNP) is a type of variation that is identified at single positions when comparing genome sequences. Some of these have no effect whatsoever (Synonymous SNPs). Others can alter the protein structure (Non synonymous SNPs) and thus alter the biological function of it. These changes have an impact on how we respond to diseases, viruses, toxins and other chemicals that enter our body such as drugs [5, 6].

**Example of SNP**

AAGGTTA --> ATGGTTA

The variation between two human genomes is rather small. Two species like human and chimpanzee do not differ that much either. SNPs difference is about 1.5% but the overall difference in the genome (related to other type of variations such as indels) is 5% [7-10].
In 2007 and 2008 the individual genomes of Venter and Watson were published,
The Watson sequence cost US$1.5 million and took 4 months to complete.
Newer methods for high throughput sequencing will make it possible to sequence individual genomes at lower cost and higher speed [11-14] .In 2006 it cost about US$ 100,000 to sequence a human genome [15]. That is 0.003 cents per base.
In 2009 the company Complete Genomics announced that they can sequence the human genome for US$5000 [16].
In the near future it will be possible to do this for US$ 1,000 and in 1 day.

SciLifeLab is a joint effort between Kungliga Tekniska Högskolan, Karolinska Institutet, Stockholms Universitet and Uppsala Universitet. Its focus is to perform research related to genome and protein profiling and bioinformatics. At the moment they can perform whole genome sequencing at the pace of 32 genomes per month, that is roughly 100Gb of data being gathered every month. Being able to sequence individual genomes has created new opportunities in analyzing variations and disease causing mutations among dozens or hundreds of individuals.

The amount of data in the Watson genome is about 6 million base pairs [17] (twice more base pairs since this was a diploid sequence). Throughout the genome about 10,000 non-synonymous SNPs where identified. Each of these substitutions can cause irregularities in the functions of a cell which leads to a disease. It is very interesting to be able to compare variations and their effects between individuals, however it is not an easy task to perform this systematically.

SNPs can be identified without having to sequence whole genomes. Today one can identify the SNPs of an individual by different SNP genotyping techniques. This is done today by companies like 23andMe at the cost of US$ 500.

### 2.4.Sequence to structure to function

Proteins consist of amino acids. There are 20 different amino acids occurring naturally in our body. Different proteins perform different biological activities in a cell, some regulate the flow of different substances in the cell (ion channels), others perform enzymatic activity, cell signaling or transport activities. Structural proteins can be found in connective tissue and motor proteins can generate mechanical force such as contracting muscles.

The protein synthesis starts with a gene being copied to messenger RNA (mRNA) in a process called transcription, the only difference between DNA and RNA is the base thymine (T) being replaced with uracil (U). The mRNA is then decoded by a ribosome,
It reads each nucleotide triplet in the mRNA and translates it to an amino acid, when the all the mRNA is read an amino acid chain has been synthesized.

To understand the biological function of a protein one cannot just look at the chain of amino acids. After the amino acid chain is synthesized the protein will fold into a three dimensional structure which determines its functionality.

When a SNP occurs this will lead to an amino acid change in the protein and thus the functionality might change.

When talking about the structure of a protein one refers to its primary, secondary, tertiary and quaternary structure.

The primary structure is the amino acid sequence. The secondary structures are regular subelements in the protein, two types of secondary structures are α helices and β sheets,
When drawing a 3d view of a protein the secondary structures are not drawn as molecules instead they are displayed in a schematic view for easier overview (see Figure 2.2 α helices are represented as the coil, and β sheets as arrows). The tertiary structure is the three dimensional packing of secondary structural elements, and the quaternary structure is the arrangement of multiple chains.

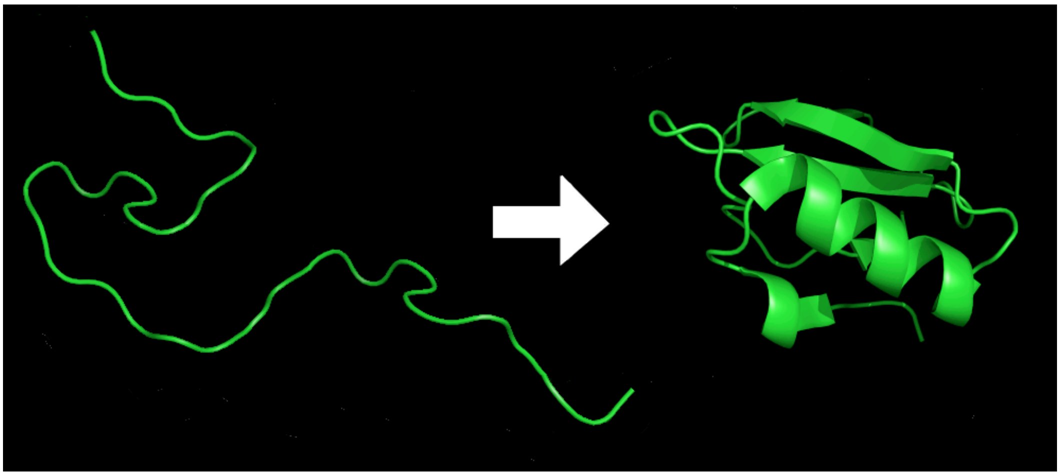


**Figure 2.2** An amino acid chain before and after folding, The α helix is represented as a coil and the β sheets as arrows. From http://commons.wikimedia.org/wiki/File:Protein_folding.png

## 2.5.Membrane proteins

The membrane protein class defines the proteins associated with the cell membrane. The cell membrane is the boundary of the cell and communication layer with the outside. They are responsible for functions such as cell to cell contact, transporting substances, enzymatic activity and signaling. Almost everything that enters or leaves the cell is determined by the membrane proteins.

25% of the human genes consists of membrane proteins [18]. 60% of all drug targets are toward this class [19]. This makes it very important to identify the SNPs for these and identify what functional changes they lead to.

Much research is directed toward membrane proteins as it is interesting to understand specific functions and characteristics such as the topology. The topology determines in what parts of the membrane the protein operates in. It can be predicted by using tools such as SCAMPI [20] or TOPCONS [21] .

As it has been difficult to understand the interaction between a protein and a membrane experimentally the membrane proteins have been a good target for computer simulations [22] .

The goal of drug designers is to find molecules that affects a protein. One method of drug design relies on the knowledge of three dimensional structure of a protein, the main goal is to identify a binding site (the receptor) and then design a molecule that binds to it (the ligand). The three dimensional structure is usually identified by crystallography, however membrane protein structures are very hard to identify due to their membrane association. The average time it takes to identify an eukaryotic protein is about one to three years. Most of the drugs that are designed today try to interact with membrane proteins. For example to treat ulcers proton pumps must be targeted to inhibit the acid production in the stomach. These types of drugs are referred to as Proton Pump Inhibitors (PPI). Some people react differently to PPIs, common side effects are headache, diarrhea, constipation and abdominal pain. Knowing what SNPs occur in an individual and how they relate to drug response might give us more information on how to design better drugs with less side effects.

## 2.6.Existing data sources

During the last decade the bioinformatics community has done a comprehensive job gathering data about genome sequences, protein sequences, protein structures and SNPs. most of this data is available to the public through different web resources. The amount of data is very extensive and keeps growing. As an example dbSNP [23] contains about 10 million entries, and sequence and base pair data in Genbank has grown faster than exponentially between 1982 and 2008 [24] [Figure 2.3, 2.4].

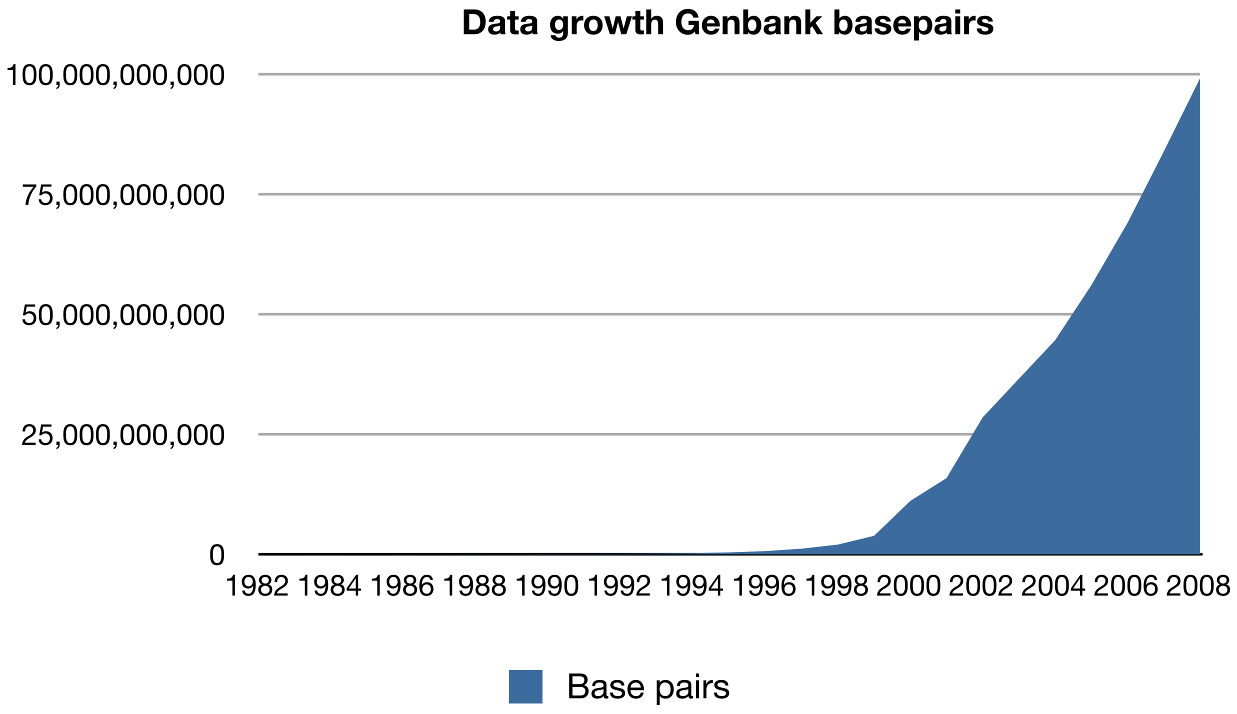


**Figure 2.3 derived from** *[24]*

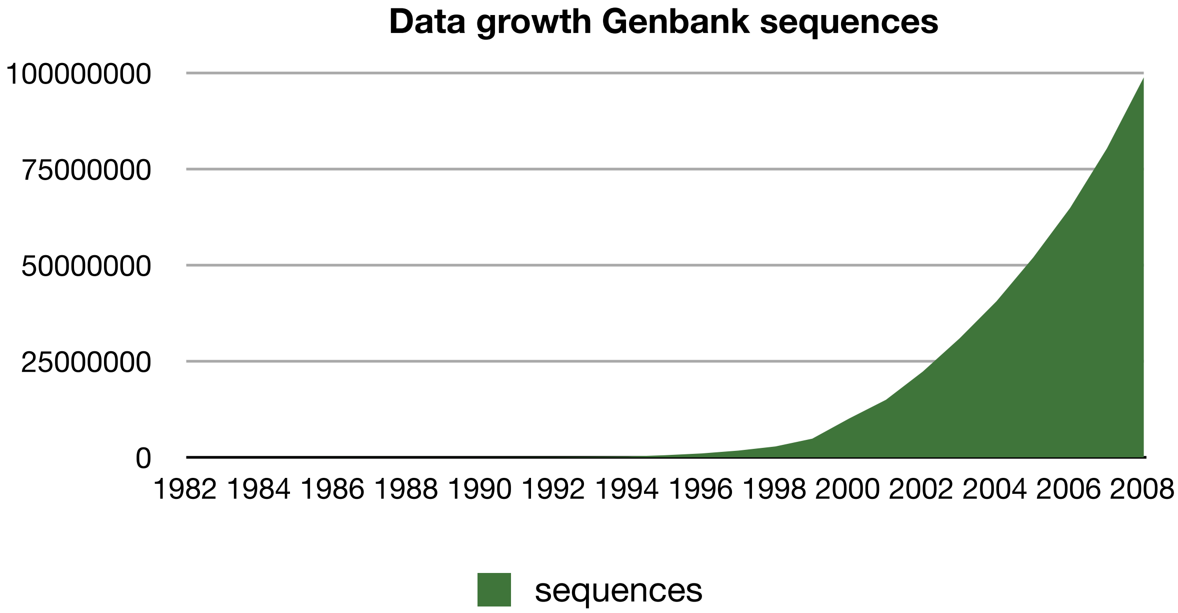


**Figure 2.4 derived from** [24]

Apart from Genbank there are over 1000 other online resources that researchers can use to gather genome data [25]. Handling this amount of data in a traditional way such as buying and manually handling it in private databases is not sustainable as it requires a lot of maintenance as the amount of data grows continuously. Therefore online resources is the common approach today as it is the simplest way to distribute data.
National Center For Biotechnology Information (NCBI) Provides 30 different databases, among those one finds genome and protein sequence databases, SNP database (dbSNP), and a catalogue of Mendelian disorders found in the human genome (OMIM).
Ensembl [26] is a joint project between European Bioinformatics Institute (EMBL-EBI) and Wellcome Trust Sanger Institute. It contains data from various eukaryotic genomes. As the name implies it specializes in combining data from many different sources and visualizes it in a genome browser.

It is not always an easy task to identify what sources to use as the sources vary in reliability, usability, relevancy and how well curated the data is. Many of the sources are not well maintained, the result of this is that they contain stale data [27]. As it is shown in Figure 2.5 there are quite many sources to choose from.
Linking the data between the different sources can be a very time consuming task and nearly impractical as one wants to compare thousands of instances.

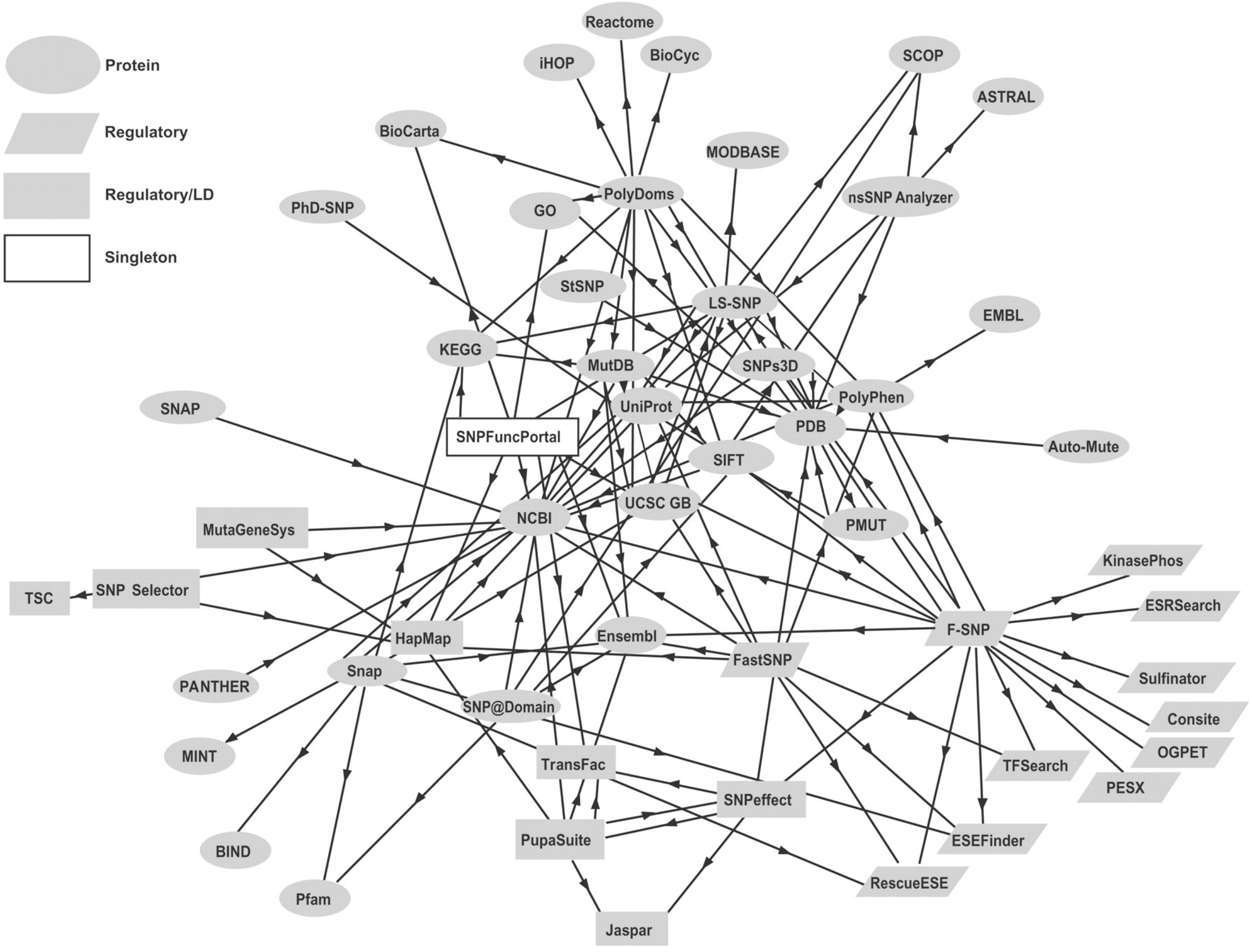


**Figure 2.5 from [27]** SNP Prediction servers and their relation to other bionformatics sources

Suggestions have been made to consolidate the genome data from current databases, clinical laboratories and scientific papers in a common knowledge database, a Genome Commons [28]. Apart from the benefit of not having to check hundreds of different databases for information it will also open up possibilities to analyze data the E-science way, that is more data driven research instead of hypothesis driven.

The Genome Commons would be a project that needs many collaborators both from the research community and the enterprise. Projects such as SciLife Lab could play a large role in contributing information and tools for diagnostic use.

Some of the sources such as NCBI [29][Figure 2.6] are linking their databases by using uniform data standards. However one needs programming experience to link this data together.

Other sources are using the metaserver approach i.e. consolidating data from various sources.

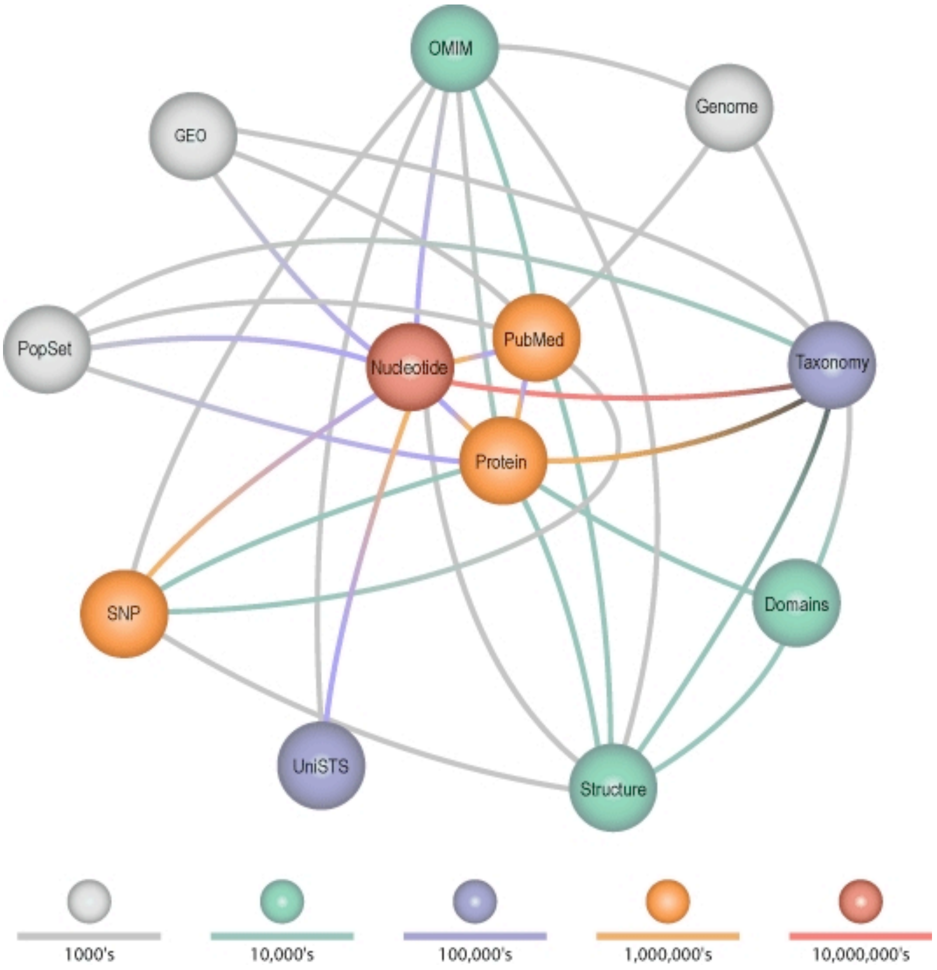


**Figure 2.6 from** [30] The various NCBI databases and their linkage

## 2.7. Problem

The large amount of information on the web is only useful if it can be converted to knowledge. The scale of the data has opened up many possibilities but it is also a problem. For example the 10,000 non synonymous SNPs in the Watson genome are not possible to analyze manually by searching through each of the databases separately, second we might not know the SNPs but we start analyzing a protein or a structure and want to gather data related to it such as SNPs, and diseases.

It is a very time consuming task to identify which databases contain reliable information. Many of the databases are not maintained very well and can contain data that is many years old.

The skills needed to use these are very different but mostly one needs expert knowledge to use them.

Linking data between the different sources is yet another time consuming task as one tries to analyze variations among thousands of individuals.

All of these hurdles force one to divert focus from gathering results to just perform tedious tasks. As one is eager to get hold of the results one comes up with so many ad hoc solutions that often can make the results irreproducible.

It would be much more effective if one can do all this in a systematic manner.

Most of the sources such as Ensembl and NCBI are gene oriented, that is the whole browsing and using is related nucleotides. This of course gives a huge flexibility but yet again endless flexibility makes things more complicated to use.

Protein sequences can be retrieved through the sources mentioned above but the membrane proteins are not easily identified

The different data sources contain information that can be difficult to link together as there one often must find data and then look up details in an article. One example of this is linking protein sequence variations with a known disease. Being able to link the data together will make it easier to gather knowledge from the data that leads to biological conclusions, for example how are SNPs related in the structure? Assume we have two SNPs one of which we know the function for, they are far away in the sequence but close in the structure can we perhaps come to the conclusion that they have functional similarities?

It is evident that one needs a tool that makes the whole data retrieving process easier. This would open up possibilities to analyze the data in a more systematic manner, be able to directly see what SNPs are disease related in a protein, and link the sequence data to structure.

# 3. System design and Implementation

## 3.1.Overview

To be able to gain further knowledge about membrane proteins we need a way to handle this large amount of fast growing data. Gathering this amount of data once to evaluate a hypothesis is not sustainable in the long run as it is very time consuming.
Creating a process that automatically gathers the data periodically will speed up this process and open up possibilities to handle and analyze data the E-science way.

The overall goal of the present is to create a database consisting of membrane proteins linked to their known diseases, both in sequence and structure.
By creating a data retrieval pipeline we have an automated process that can gather and link data from various sources.
The pipeline collects human protein sequences and identify the membrane proteins by a prediction method called SCAMPI. In addition to the membrane proteins this will give us the topology locations of the proteins.
The pipeline also gathers all the available protein structures. For each membrane protein sequence identified it will also find all the non-synonymous SNPs and their relation to disease.
A web user interface was created where one can search for proteins or diseases. The user interface provides a view where one can see a protein and all its related data. One will be able to see the identified SNPs, see where in the topology they are located. In the cases a SNP is related to a disease this relation will be visible. For proteins with structures we will be able to visualize the relation between sequence and structure, for example one will be able to see where in the structure a SNP exists.
Most of this will be implemented using Java based tools and frameworks such as Hibernate, Spring and Grails.

## 3.2.The main components in the database

The key components in the database will be the protein sequences, the variations and the diseases. Any other feature that will be added in the future will rely on at least one of these components. Since we want to gather data from many different sources the data structure of each of these components must designed in such a way that linking between different external sources is possible, thus we cannot just have one type of id, instead we must be able to add different ids that the various sources are using.

## 3.3.The data gathering pipeline

In order to keep the database up to date we need to have an automated pipeline that can gather data from different sources regularly.
The pipeline should be designed in a way to support gathering data from many different sources for each component, it should also be extendable so that new components can be added over time. In practice a component will correspond to a data model in the software implementation, even when we get data from different sources that sends us different formats the data model for a component will ensure that we have uniform data [Figure 3.1].

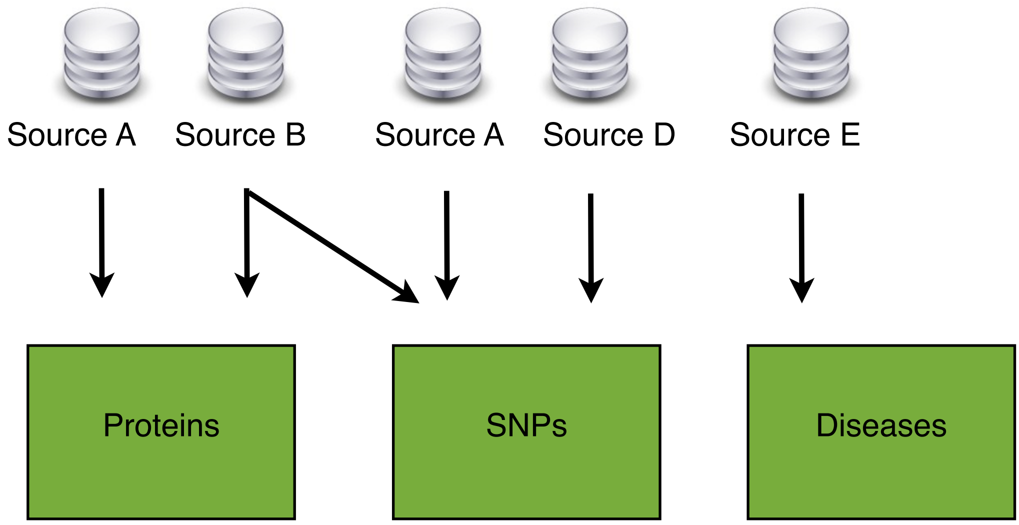


**Figure 3.1** Data can come from many sources same source can provide data to many components (like source B). Each component ensures that the data is saved in uniform manner even if the sources send data in different formats

Secondary components such as structures and topology data depends on a main component.
The main components (proteins sequences, variations and the diseases) depend on each other and have to be gathered in a sequential manner. However this does not mean that one should need to start from step 1 each time when gathering data.

**Gathering order of the main data components**

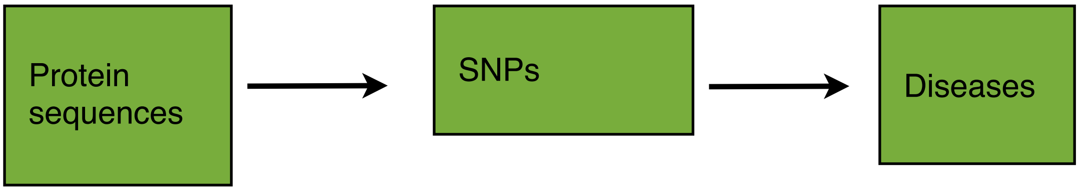


**Figure 3.2** Gathering order

As long as the dependent data is in place the pipeline can update one data component as many times as needed.  This provides a robustness in cases where just a certain step has gone wrong and a level of flexibility when one just need to update the data of one component (for example when more diseases have been found and we only wish to update these) [Figure 3.2].

### 3.4.Choices of data sources

To ensure good quality of our database we must in turn use data sources that fulfil certain requirements.

We wish to use data sources where people often submit new data to and avoid the ones that contain stale data. The data sources should also have common standards so it will be easy to link the data between them.

The sources should be well maintained, and already have a broad user base. These two factors ensures us that the data has a certain level of quality, and that the source will stick around for some time.

To sum up we want the data sources to be:

- Updated regularly
- Using common standards
- Well maintained
- Generally accepted and used by the bioinformatics community

NCBI and Ensembl are two sources that fulfil our requirements above.
From Ensembl we chose to retrieve protein sequences and their SNPs and from NCBI we chose to retrieve disease information from OMIM.
NCBI also have protein sequences and SNPs however it was easier to retrieve data from Ensembl in those two cases. The structures was retrieved from PDB [Figure 3.3].

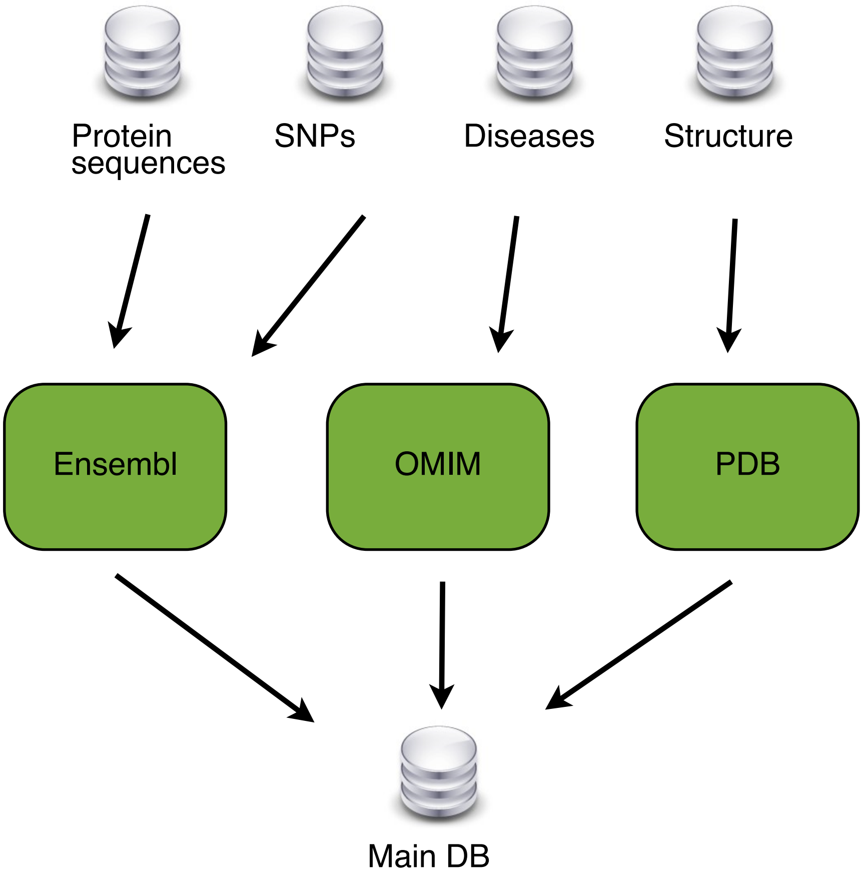


**Figure 3.3** Choices for data sources. green boxes represent the sources, and silver discs represent the data types retrieved from the different sources

Sources can return data in various ways. One can usually connect directly to databases or retrieve data through a web service or download flat text files (FASTA, XML etc).

When using the sources one should as much as possible avoid connecting directly to the SQL databases since this requires one to find out the database structure.
The reason is that the data there is normalized and you will end up having to join many different tables to get hold of the data. This requires you to get a good understanding of the database structure. To save time one should try to use web services as much as possible.
The web services have already done most of this work for you so you will only need to parse the received data.

Ensembl provides its web services via a tool called Biomart [31], and NCBI provides it via Entrez Programming utilities (Eutils) [32].

### 3.5.Understanding how the data is linked between sources

To link data between different sources one must identify the common ids they share between each other.
Proteins and SNPs in Ensemble are linked together by a specific Ensembl protein id and transcript id.
SNPs in Ensembl have two different id formats, one is prefixed with rs (reference cluster id) and the second is prefixed with ENSSNP. The rs id is the DBSNP reference identifier thus when referencing SNPs in the OMIM catalogue the rs is is used [Figure 3.4].

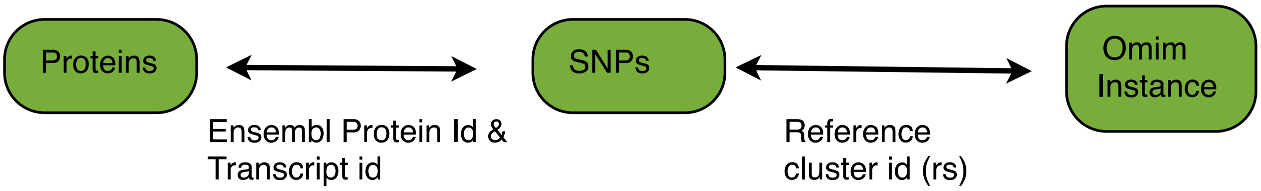


**Figure 3.4** Id linkage between the different data types

### 3.6.Retrieving the data

Identification of membrane proteins was done by running the longest protein transcript for all genes through SCAMPI, the topology prediction tool. The proteins sequences where retrieved from Ensemble as FASTA flat files. Besides classifying membrane proteins SCAMPI also generates a topology sequence for the protein. With the topology sequence one can identify where in the membrane a specific amino acid is located. This part was set up as a separate service from the rest since the identification programs where already in place. Previous work had been done on this by Aron Hennerdal where he created a script to download all the proteins sequences from Ensembl and run topology prediction on this. During this thesis work a web service was wrapped around this script. The purpose of this was to be able to retrieve the data output in a structured way and possibly by many different servers. With the web service one can send a command to run a new topology prediction and return the output.

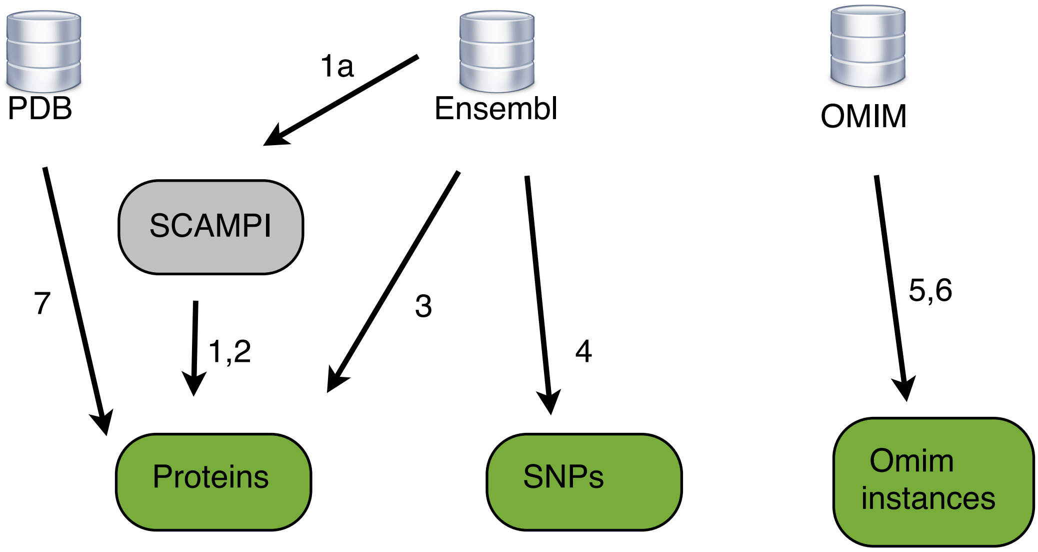


**Figure 3.5** All the current steps in the pipeline

1. Starting by retrieving the membrane protein sequences from the SCAMPI web service, The web service first retrieves all protein sequences from Ensembl (1a) and then determines which ones are membrane proteins.
2. As a second step we call the Scampi web service to retrieve the topology location sequence for each Membrane protein.
3. Call Ensembl to retrieve some additional information about each protein such as a basic description and gene name.
4. Retrieve all the SNPs for the proteins.
5. Find all SNPs linked between dbSNP and OMIM.
6. Retrieve all the OMIM instances received in Step 5.
7. Gather PDB files for each protein available in the database.

After retrieving the membrane proteins from the above described service we then make another call to Ensembl to get some further info such as a description and the name of the protein and the gene it is linked to.

The Biomart service requires one separate call to receive the variations. After retrieving those one can now link to OMIM via the variation ids. Eutils require one to first query the web service to get all links between dbSNP and OMIM. Once that is done, one can retrieve each OMIM instance. Each OMIM instance is actually a journal, and there is only one specific section of that journals that references the variation i.e. what disease it relates to. To find the related disease name one has to parse each document and see what parts are relevant to the SNP. Figure 3.5 has a graphical representation of the flow.

## 3.7.Representing the data

The representation layer should be usable without any specialized bioinformatics knowledge.

The main workbench should give an overview of a specific protein, its variations and what diseases are possibly is linked to them. One should also, at a minimum, be able to see where in the sequence a variation occurs. A view where one can see where in the structure a variation occurs is also highly desirable.

To reach the main workbench the user should be able to search for a protein or a disease. A free text search where users can type in an Ensembl protein id, protein name or disease name will solve this.

When the search results in a protein, users are directly shown the protein workbench overview, when searching for a disease, one is presented with a list of proteins related to the disease and choosing a protein will direct the user to the protein workbench.

## 3.8.Choices of technology

The main work of the application is to pull data from various sources on the web. This type of work often involves parsing data in different formats such as XML, JSON or plain text. The representation part consists of a web interface where one can query for proteins and diseases.

The design of the application is extensible so that new capabilities can be enabled in the future such as providing web services that can serve raw data.

This requires an implementation that is easy to maintain and built on sound development practices and design patterns. Modularized code is an obvious way to go, we can achieve this by using well known design patterns such as Model View Controller (MVC) [33], write test code, using Object relational mapping (ORM) [34] to remove the need of data access layer code and remove the dependency of the underlying database.

### 3.8.1.Hibernate

Hibernate is an implementation of ORM for Java.

Traditionally a data access layer was needed to interface with the underlying database. The responsibility of the data access layer was to query the database and map the data onto a data structure or class that the rest of the application will use.

Hibernate is a replacement for the data access layer. It allows a class to be mapped directly to a database table. It also provides data access methods and a SQL inspired query language called HQL (Hibernate Query Language). This makes an application less dependent on the underlying database technology and allows one to change the underlying database technology without having to re-implement custom SQL queries.

In this application Hibernate is used indirectly via Grails and allows us to design a class architecture which Hibernate will translate to a database schema and handle communication and queries to it.

### 3.8.2.Spring Framework and Grails

The Spring Framework [35] is a set of tools that simplifies application development on the Java platform. The framework was released 2003 and its aim was to simplify J2EE application development. The tools are designed in a way that they are easy to introduce in a current application without having to modify large parts of an application.
Among its toolset is a flexible MVC framework that simplifies web application development.Testing features that simplifies creation of unit and integration tests and an Aspect Oriented Programming framework.

Grails [36] is another framework that utilizes the tools available in Spring to increase the productivity in web application development. The framework is designed around current agile development practices such that it includes tools for the whole process from project setup to application deployment. The strategy of Grails is to rely on conventions over configurations which is a big reason for the high productivity gain.
Convention over configuration means that the framework can assume a certain structure in the application, this enables the framework to know how to prepare the application for a development or a production without having to rely on external build scripts, create test skeletons automatically when Domain classes are created and the MVC structure makes a clean separation between application logic, data models and generation of HTML views. Other tools that Grails provides simplifies the consumption web services and parsing of XML and JSON messages. It also provides simple mechanisms to create custom web services and powerful methods to map a class directly to XML or JSON.

One often choses a web framework to speed up the development but gets stuck in the limitations of the framework, thus time is consumed by working around these limitations. The benefit of using Grails is that you are never stuck with just the tools the framework provide, tools deeper down the framework stack are always available [Figure 3.6] and use Hibernate or Spring or just plain Java when needed.

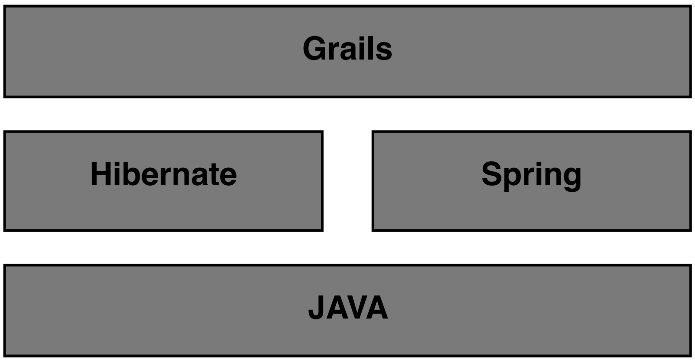


**Figure 3.6** The framework stack hierarchy, Grails works on top of Hibernate and Spring, if limitations are found in Grails one can go further down the hierarchy and use Hibernate, Spring or Java directly

Grails, Spring and Hibernate combined fulfil the requirements for this application very well, The simplify usage of web services, insertion of data into a database and creation of a web application.

Other popular web programming languages such as PHP is discouraged when dealing with applications that need to gather large amounts of data. PHP has no concept of threading or parallel processes.
This would be a limit in the future when one wants to optimize the data retrieval process by running tasks in parallel. A PHP script also has a fixed execution time and a fixed memory limit which one ends up tweaking from time to time.

A benefit of running on the Java platform is also its server technologies with very good queuing functionalities and rollback of transactions once a server goes down. This will allow one to build a robust application.

### 3.8.3. Jmol - molecular representation

Being able to represent sequence variations on the molecular structure was a highly desirable feature in the web interface. In order to achieve this a molecular viewer was needed.
Jmol is a molecular viewer that has the capability of representing PDB files in 3D.
One if the modules it provides is a Java applet. An applet is a small Java program that can run in a web browser. Jmol also provides a Javascript interface to communicate with the rest of the web application. This interface provides simple javascript methods to initialize the applet and load PDB files, it also provides methods to send Jmol scripts directly to the applet. These capabilities allows a web application to integrate with the Jmol viewer.

When loading the main protein workbench the Java applet will be initialized and instructed to load a PDB file.
The Javascript command is wrapped around an HTML element in order to define the layout.

```
<div class="viewer">
  <script>
     jmolInitialize("/thesis/js/jmol-12.0.12",true);
     jmolApplet(["100%","100%"], "load 'http://www.rcsb.org/pdb/files/
     1WB0.pdb';set measurementUnits ANGSTROMS; select all;spacefill off; wireframe
     off; backbone off; cartoon on; color a-helix #800000 ; select
     ligand;wireframe 0.16;spacefill 0.5; color cpk ; select all; model 0;set
     antialiasDisplay true; ;save STATE state_1");
  </script>
```

The function call *jmolInitialize* makes the applet available to the web application. After initialization we load the applet with the function call *jmolApplet* along with the options and settings. Among the more important options used in the call above is the instructions to fill 100% of the div element and which PDB file to load into the viewer.

To let the rest of the application communicate with the applet we apply event listeners that when triggered send Jmol scripts to the applet.
In particular when a user clicks on a letter in the protein sequence or a SNP the viewer should highlight the corresponding part in the structure.
Part of the javascript implementation can be seen above. When clicking on a letter, we need to have some bookkeeping to understand if we selected or deselected a letter and also know what letters have been selected prior to the current click.
A query script is prepared for Jmol that instructs the viewer on what residues to select and how to highlight them. In the code above the query script is stored in the variable scriptString. The first part `"select "+residues.join(' OR ')"` explains what residues we want to be highlighted by the viewer and the second part `selectionHalos ON` explains how the selected residues should be highlighted.

```
//when clicking on a sequence or structure this function is called
function markStructure(residueNum){
....  //some logic to find out if the residueNum should be selected or deselected
.... //an array is created with the residues that have been selected

 highlightResidue(selectedResidues);
}

//prepares a script string to send to the Jmol applet
function highlightResidue(residues){
     var scriptString = '';
     if(residues.length==0) //if nothing is selected
            scriptString = 'select none';
     else
            scriptString = "select "+residues.join(' OR ')+" ;selectionHalos ON";

     jmolScript(scriptString); //sending the script to the Jmol applet
}
```

# 4. Results

The amount of data gathered by the pipeline can be seen in Figure 4.1.

| Data type | Number of instances |
|---|---|
| Proteins | 7164 |
| SNPs | 43439 |
| Diseases | 572 |
| Disease related SNPs | 984 |
| PDB structures | 3051 |
| Proteins having PDB structures | 724 |

**Figure 4.1**

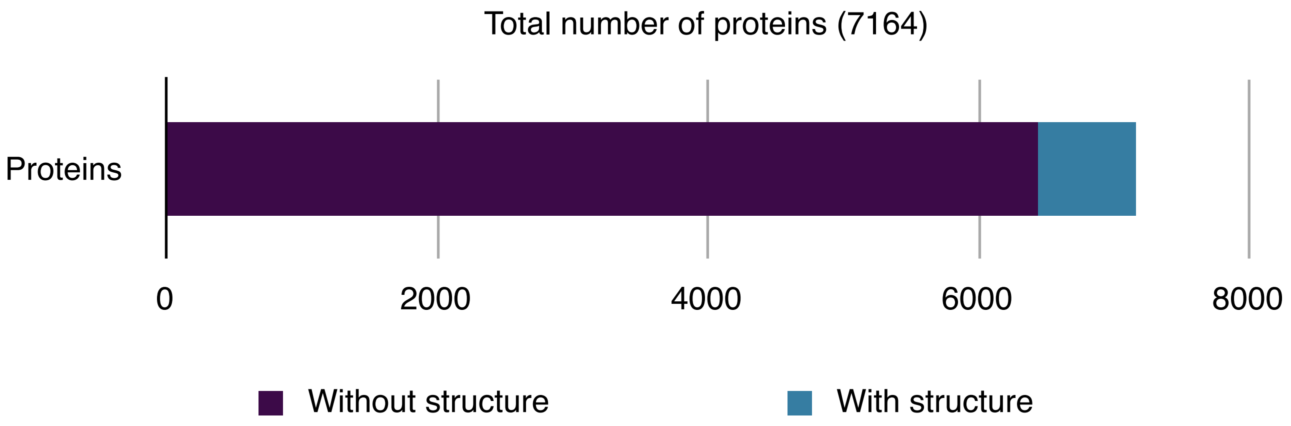


**Figure 4.2**

In total 7164 proteins where identified as membrane proteins. This value is higher than found in studies. It is assumed that we have an overprediction of membrane proteins. To make this more exact a better prediction method such as Topcons can be used, however this would take about 1 minute for each protein compared to SCAMPI that took about 15 minutes for all 25000 human proteins.
3051 structures were found in the PDB, but one protein can have many structures, thus the amount of unique structures is 724. The reason a protein has many structures is that each structure might actually only represent parts of the whole protein, a second entry might contain a missing portion of a molecule, a third entry might contain a structure with higher resolution (the atomic level picture of the molecule, higher resolution means more correct atomic coordinates measured in Ångström i.e. less errors in the crystalized structures)

In the web interface we currently chose to display only the structure with the highest resolution.

10% of the proteins have identified structures.
The amount of structures found was quite surprising as we expected pretty much no structures to be found for human membrane proteins.
The structures identified where not whole structures, but just portions of the whole molecule. More complete structures can be found by using protein structure prediction methods or using meta servers such as pcons.net [37, 38] that combine different structure prediction methods.
As this was done late in the project not much information was gathered about this beforehand, the concept still works but one needs to rely on another way to find structures instead of PDB, perhaps structure prediction methods.

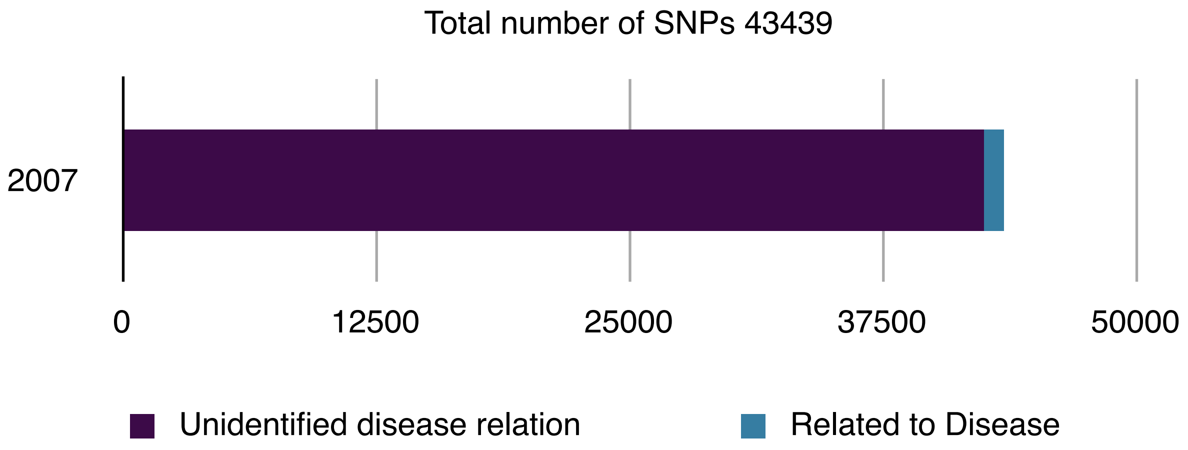


**Figure 4.3**

It is just 2.2% of the SNPs [Figure 4.3] that have already been identified as disease related.
Regarding the SNPs one could argue that perhaps there is not that many disease causing mutations happening in the membrane proteins but this is very unlikely. It is more likely that data around diseases and SNPs have been gathered individually but there still remains work to link these two together.
Using SNP effect prediction [39] could help expanding the knowledge not only by identifying new possible disease causing SNPs but also finding out which SNPs affect individuals’ response to chemicals.

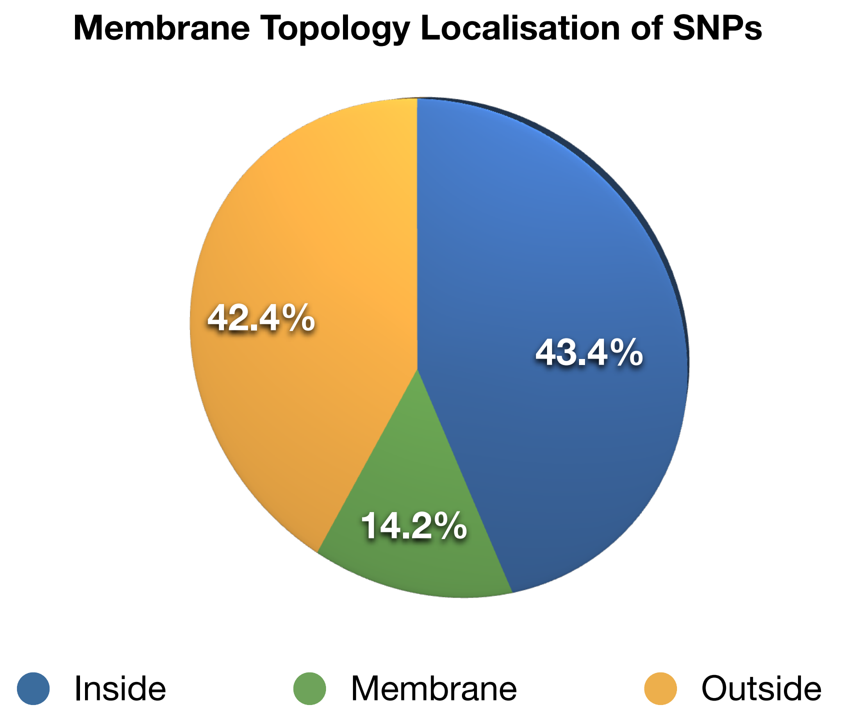


**Figure 4.4**

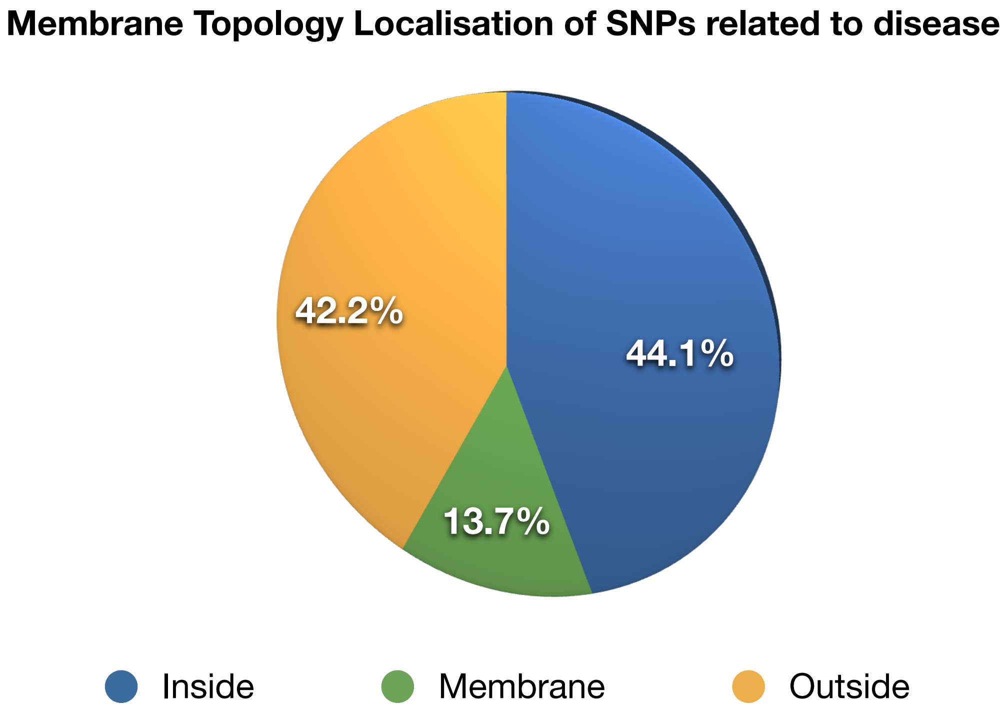


**Figure 4.5**

The distribution of SNPs in the membrane topology is similar when comparing all identified non-synonymous SNPs to the ones related to disease [Figure 4.4, 4.5].
It would be interesting to classify the membrane proteins by transmembrane spanning domains to see if the topology distribution differs. On single protein instances like Apolipoprotein E it is very clear that the most SNPs are located in one area [Figure 4.6]

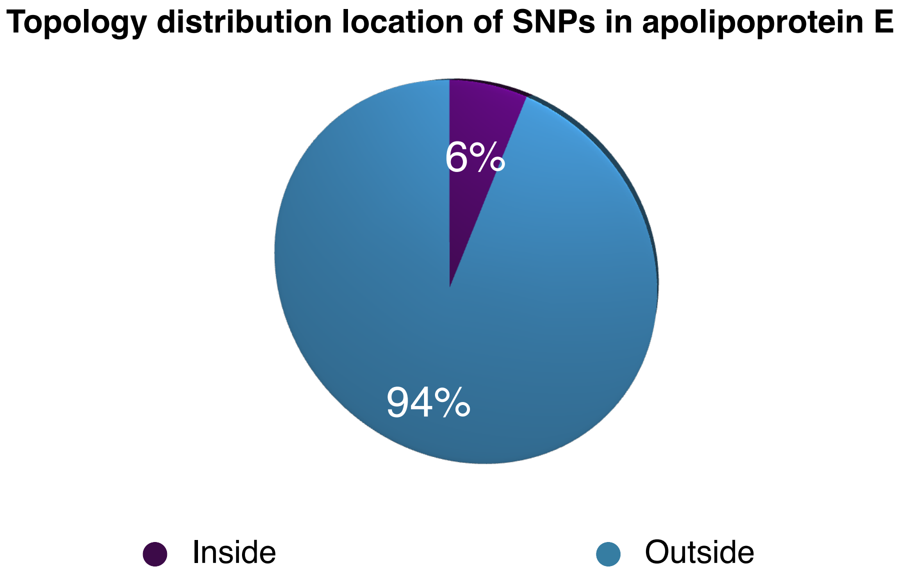


**Figure 4.6**

When gathering and relating this data a couple of things had to be improved during development. The first thing was that the data gathering pipeline was not general enough. For example when Ensembl performs updates and their data returns in another manner it propagates to the pipeline thus creating an error. Some improvements were done to make the pipeline more robust. The main strategy was to use as general data as possible, for example avoiding to use the filtering options provided by Ensembl and instead filter manually in the code as the Ensembl filters could change the return parameters between releases or simply vanish.

It is also fairly difficult to verify the data integrity just by performing small tests. Because people will probably gather further results based on this data it is imperative that a high level of data integrity is a must. This part of testing and validating the pipeline with respect to data integrity was the most time consuming part since most of the errors showed up once we started to insert larger amounts of data. A lot of assumptions had been made about the different data structures but many special cases had not been considered, as a full update of more than 5000 proteins including all of its variations and diseases could take 1.5 h it took a couple of iterations before all the special cases were found.

The special cases were added to the unit tests but also data integrity checks were performed dynamically while inserting data. This was very crucial as we want to avoid halting the pipeline. A benefit of this is that we can still continue updating the data and only log the instances that had errors.

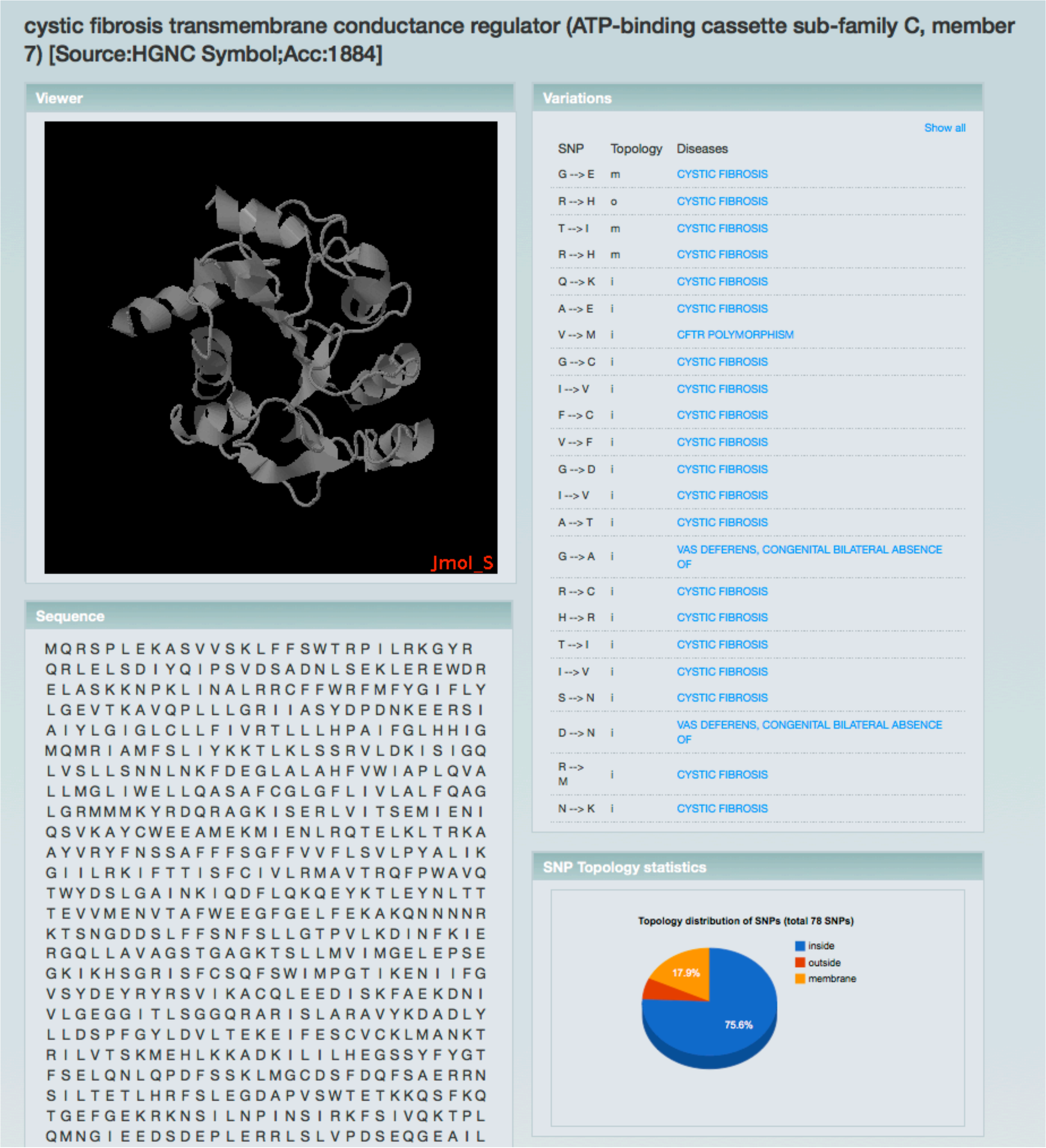
An example of an incorrect assumption was the ids of SNPs which were thought to be unique, but this was not true. An id of a SNP describes a shift in a gene. A gene can have many different transcripts but the SNP can occur in all of them but the positions will differ. All of these SNPs had the same ID which caused errors. When one asks for a SNP from Biomart you get the same SNPs for all transcripts of the gene. In those cases we had to filter the response and retrieve the SNP for the correct transcript.

The Hibernate persistence framework was used to handle database operation. Although it has high productivity benefits there are some best practices one has to consider especially when inserting large amounts of data at once. During the initial testings it was observed that it took longer time to insert single instances as we went further along the pipeline.
In the Hibernate session there exists a first level cache that is used to store the objects that will persist in the database. When performing batch inserts like we do in the pipeline this first level cache keeps growing which makes the insertion process slower as the amount of inserts occur [40].
Clearing the cache after each 100th insert solved the problem.

## 4.1.The web interface

The web interface provides a simple way to see the proteins and to see what variations and diseases are related to them. One can perform full text search on a protein name or its associated gene name or just browse the whole protein index.

An example of the protein view for CFTR is shown in figure 4.7



cystic fibrosis transmembrane conductance regulator (ATP-binding cassette sub-family C, member 7) [Source:HGNC Symbol;Acc:1884]

Viewer

Variations

Show all

| SNP | Topology | Diseases |
|---|---|---|
| G --> E | m | CYSTIC FIBROSIS |
| R --> H | o | CYSTIC FIBROSIS |
| T --> I | m | CYSTIC FIBROSIS |
| R --> H | m | CYSTIC FIBROSIS |
| Q --> K | i | CYSTIC FIBROSIS |
| A --> E | i | CYSTIC FIBROSIS |
| V --> M | i | CFTR POLYMORPHISM |
| G --> C | i | CYSTIC FIBROSIS |
| I --> V | i | CYSTIC FIBROSIS |
| F --> C | i | CYSTIC FIBROSIS |
| V --> F | i | CYSTIC FIBROSIS |
| G --> D | i | CYSTIC FIBROSIS |
| I --> V | i | CYSTIC FIBROSIS |
| A --> T | i | CYSTIC FIBROSIS |
| G --> A | i | VAS DEFERENS, CONGENITAL BILATERAL ABSENCE OF |
| R --> C | i | CYSTIC FIBROSIS |
| H --> R | i | CYSTIC FIBROSIS |
| T --> I | i | CYSTIC FIBROSIS |
| I --> V | i | CYSTIC FIBROSIS |
| S --> N | i | CYSTIC FIBROSIS |
| D --> N | i | VAS DEFERENS, CONGENITAL BILATERAL ABSENCE OF |
| R --> M | i | CYSTIC FIBROSIS |
| N --> K | i | CYSTIC FIBROSIS |

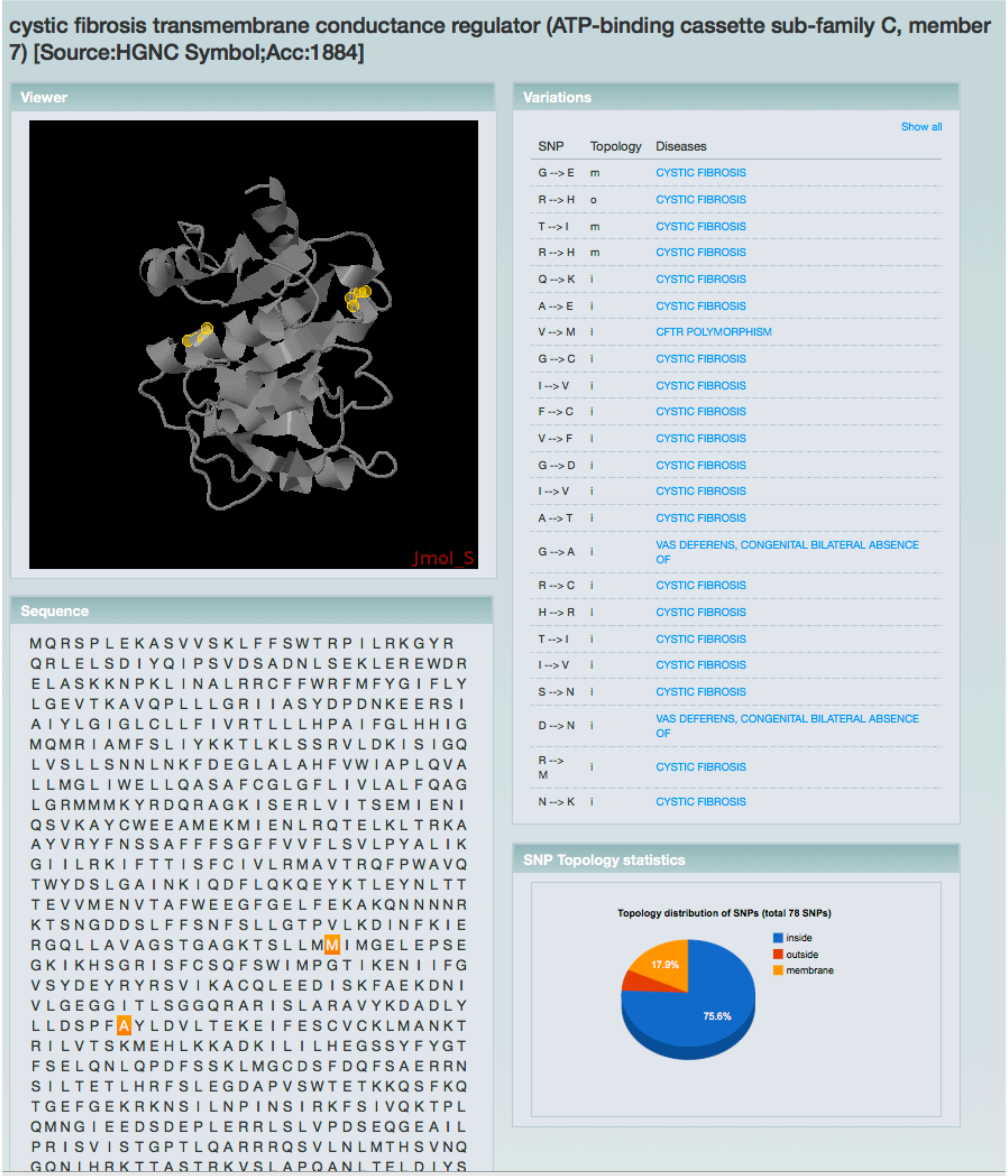
Sequence

```
MQRSPLEKASVVSKLFFSWTRPILRKGYR
QRLELSDIYQIPSVDSADNLSEKLEREWDR
ELASKKNPKLINALRRCFFWRFMFYGIFLY
LGEVTKAVQPLLLGRIIASYDPDNKEERSI
AIYLGIGLCLLFIVRTLLLHPAIFGLHHIG
MQMRIAMFSLIYKKTLKLSSRVLDKISIGQ
LVSLLSNNLNKFDEGLALAHFVWIAPLQVA
LLMGLIWELLQASAFCGLGFLIVLALFQAG
LGRMMMKYRDQRAGKISERLVITSEMIENI
QSVKAYCWEEAMEKMIENLRQTELKLTRKA
AYVRYFNSSAFFFSGFFVVFLSVLPYALIK
GIILRKIFTTISFCIVLRMAVTRQFPWAVQ
TWYDSLGAINKIQDFLQKQEYKTLEYNLTT
TEVVMENVTAFWEEGFGELFEKAKQNNNNR
KTSNGDDSLFFSNFSLLGTPVLKDINFKIE
RGQLLAVAGSTGAGKTSLLMVIMGELEPSE
GKIKHSGRISFCSQFSWIMPGTIKENIIFG
VSYDEYRYRSVIKACQLEEDISKFAEKDNI
VLGEGGITLSGGQRARISLARAVYKDADLY
LLDSPFGYLDVLTEKEIFESCVCKLMANKT
RILVTSKMEHLKKADKILILHEGSSYFYGT
FSELQNLQPDFSSKLMGCDSFDQFSAERRN
SILTETLHRFSLEGDAPVSWTETKKQSFKQ
TGEFGEKRKNSILNPINSIRKFSIVQKTPL
QMNGIEEDSDEPLERRLSLVPDSEQGEAIL
```

SNP Topology statistics

**Figure 4.7** Protein view for CFTR

In this view one can see the protein's structure if one exists, in the case of CFTR a full structure is not represented but only a structure of parts of the sequence.
In the viewer one can rotate and zoom as well as choosing to represent the structure in other styles. The variations table shows all non-synonymous SNPs related to the protein. In the case above the protein had 79 SNPs so they variations table is filtered to only show the disease related ones. An option to choose all SNPs is available by clicking "show all" SNP associated with a disease also have a link provided that refers to the corresponding OMIM article where one can read more about the disease [Figure 4.8].



cystic fibrosis transmembrane conductance regulator (ATP-binding cassette sub-family C, member 7) [Source:HGNC Symbol;Acc:1884]

Viewer

Variations

Show all

| SNP | Topology | Diseases |
|---|---|---|
| G --> E | m | CYSTIC FIBROSIS |
| R --> H | o | CYSTIC FIBROSIS |
| T --> I | m | CYSTIC FIBROSIS |
| R --> H | m | CYSTIC FIBROSIS |
| Q --> K | i | CYSTIC FIBROSIS |
| A --> E | i | CYSTIC FIBROSIS |
| V --> M | i | CFTR POLYMORPHISM |
| G --> C | i | CYSTIC FIBROSIS |
| I --> V | i | CYSTIC FIBROSIS |
| F --> C | i | CYSTIC FIBROSIS |
| V --> F | i | CYSTIC FIBROSIS |
| G --> D | i | CYSTIC FIBROSIS |
| I --> V | i | CYSTIC FIBROSIS |
| A --> T | i | CYSTIC FIBROSIS |
| G --> A | i | VAS DEFERENS, CONGENITAL BILATERAL ABSENCE OF |
| R --> C | i | CYSTIC FIBROSIS |
| H --> R | i | CYSTIC FIBROSIS |
| T --> I | i | CYSTIC FIBROSIS |
| I --> V | i | CYSTIC FIBROSIS |
| S --> N | i | CYSTIC FIBROSIS |
| D --> N | i | VAS DEFERENS, CONGENITAL BILATERAL ABSENCE OF |
| R --> M | i | CYSTIC FIBROSIS |
| N --> K | i | CYSTIC FIBROSIS |

Sequence

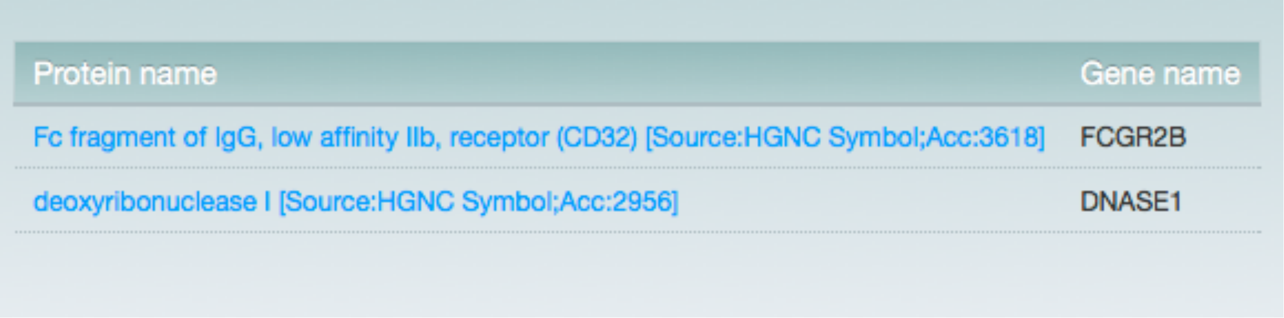
```
MQRSPLEKASVVSKLFFSWTRPILRKGYR
QRLELSDIYQIPSVDSADNLSEKLEREWDR
ELASKKNPKLINALRRCFFWRFMFYGIFLY
LGEVTKAVQPLLLGRIIASYDPDNKEERSI
AIYLGIGLCLLFIVRTLLLHPAIFGLHHIG
MQMRIAMFSLIYKKTLKLSSRVLDKISIGQ
LVSLLSNNLNKFDEGLALAHFVWIAPLQVA
LLMGLIWELLQASAFCGLGFLIVLALFQAG
LGRMMMKYRDQRAGKISERLVITSEMIENI
QSVKAYCWEEAMEKMIENLRQTELKLTRKA
AYVRYFNSSAFFFSGFFVVFLSVLPYALIK
GIILRKIFTTISFCIVLRMAVTRQFPWAVQ
TWYDSLGAINKIQDFLQKQEYKTLEYNLTT
TEVVMENVTAFWEEGFGELFEKAKQNNNNR
KTSNGDDSLFFSNFSLLGTPVLKDINFKIE
RGQLLAVAGSTGAGKTSLLMMIMGELEPSE
GKIKHSGRISFCSQFSWIMPGTIKENIIFG
VSYDEYRYRSVIKACQLEEDISKFAEKDNI
VLGEGGITLSGGQRARISLARAVYKDADLY
LLDSPFAYLDVLTEKEIFESCVCKLMANKT
RILVTSKMEHLKKADKILILHEGSSYFYGT
FSELQNLQPDFSSKLMGCDSFDQFSAERRN
SILTETLHRFSLEGDAPVSWTETKKQSFKQ
TGEFGEKRKNSILNPINSIRKFSIVQKTPL
QMNGIEEDSDEPLERRLSLVPDSEQGEAIL
PRISVISTGPTLQARRRQSVLNLMTHSVNQ
GQNIHRKTTASTRKVSLAPQANLTELDIYS
```

SNP Topology statistics

**Figure 4.8** View of CFTR, letters sequence are highlighted with orange, corresponding letters are also highlighted in the structure

To see the location of a SNP in the structure one can click on it. Yellow orbs will highlight the location of the amino acid in the viewer, also the sequence view will highlight the peptide shift in orange. This can be interesting in cases where one wants to see if two SNPs might be far away in the sequence but close in the structure.
The topology distribution of the SNPs are also displayed In this case 75% of the SNPs are located in inside region of the membrane.

Another feature in the search is that you can search for a disease. In this case all proteins that are related with that disease will show up. Clicking on them will take you to the protein view.
In the example below a search has been made for “lupus” and we hope to find results related to Systemic Lupus Erythematosus.
In this case 2 proteins were found [Figure 4.9], clicking on the protein will take you to the main protein workbench



| Protein name | Gene name |
|---|---|
| Fc fragment of IgG, low affinity IIb, receptor (CD32) [Source:HGNC Symbol;Acc:3618] | FCGR2B |
| deoxyribonuclease I [Source:HGNC Symbol;Acc:2956] | DNASE1 |

**Figure 4.9**

# 5. Future work

Based on the results new ideas came up that extend the functionality and will help in finding new results. Some of those ideas are discussed in this section.

It was expected that classifying SNPs by topology location in membrane proteins would lead to a conclusion that we can find more diseases in one location. Most of the disease related SNPs where located in the inside or outside region of the membrane but these numbers where similar to the topology distribution of all identified SNPs thus we cannot pinpoint a location in the topology where diseases occur more frequently.
However looking at single protein instances we see that the SNPs occur more often in one specific region. This could tell us that further classification of membrane proteins might show us that specific classes contain more SNPs in a specific topology region which could mean that certain membrane protein classes  can be related to disease more often than others.

The pipeline concept can actually be extended to include all proteins in the human geneome. Things such as topology data would only apply to the membrane proteins which would be identified as a class of their own in the full protein set.
It is straightforward to increase the dataset of variations and diseases by extending the pipeline to gather data from other sources than NCBI and Ensembl. Any new source that is chosen must however conform to certain standards to guarantee reliable data.
Other types of variations such as Multi Nucleotide polymorphisms (MNP), insertions, deletions, duplications, insertion-deletions is yet another way to extend the variation data.
For users that want to use the data in their own databases or other projects a web service can be provided where one can query and retrieve the data in a standard format such as XML or JSON.

As more and more individual genomes will get sequenced one will be able to gather biological conclusions in an easy manner. The possibility to submit the fully sequenced genome to the database will show what variations exists and their linkage to diseases.
Extending the ways to search data in more intuitive ways such as “find all diseases in the outer membrane regions” or “find all proteins having more than 90% of their SNPs in the membrane region” will also give a broader overview and hopefully provide more conclusions about the membrane proteins.

The standard way of finding disease related SNPs is through genome wide association studies (GWAS) [41, 42], but there could be other ways to analyze current data and conclude if a variation could be disease related or not.
One possible strategy is to keep a conservation score on each protein sequence. The regions that have a high conservation could represent very important parts of a protein thus if changed could result in a disease.
Another strategy to find more disease related SNPs is to find amino-acids that co-evolve as there could be many variations in different proteins that lead to a disease.
Relevance of non-synonymous SNPs can also be analyzed to conclude if it changes the function of a protein. A peptide shift that changes the amino acid from leucine to isoleucine might not lead to a functional change as compared to one shifting to alanine.
Protein function prediction on SNPs is also another way to identify possible disease related SNPs.

# 6. Conclusions

MembraneLogic is a program that consists of a database, an automated data gathering pipeline and a web interface.
The automated pipeline gathers and identifies membrane protein sequences, protein structures, SNPs and Disease information from various databases such as Ensembl, OMIM and PDB, links these together and inserts this information in a database of its own. The pipeline is designed in such way that other data sources can be added further on.
A web interface was created where one can search for proteins and diseases. With the help of the web interface it is possible to see where in the sequence and structure a variation or disease occur.
The database is a MySQL database and the pipeline and the web interface was implemented using Java based tools and frameworks such as Grails, Hibernate and Spring framework.

About 2.2% of the non-synonymous SNPs were related to disease. Does this mean that genetic variations is not a cause for disease? It is clear that changing peptides can alter the functions for a protein, It is more likely that linking diseases and SNPs is only at its early stages and the amount of data here will grow in time.
A disease is not caused by a variation in a single gene, interactions in other genes, environmental and lifestyle factors are also factors that one should count in. But currently it is difficult to analyze all these different factors as one would need lots of data from many individuals throughout their lives.

The pipeline now provides a solid platform to build upon further.
Apart from providing researchers with easy accessible data this database can during the drug design process aid in understanding where in the protein structure a variation occurs.In the area of personalized medicine a database like this will be able to aid in finding impacts of genetic variation medication response. Personalized medicine is a market that will expand as proactive medical treatment grows.
MembraneLogic has provided a concept that will shorten the step between data gathering and biological conclusions. It can be accessed at http://grenache.theophys.kth.se:8080/thesis.

# Bibliography

1. Brown, T.A, *Genomes*, in *Genomes*. 2002, BIOS scientific publishers Ltd. p. 1-13.
2. Consortium, The International Human Genome Mapping, *Humane Genome.* Nature, 2000(409): p. 934-941.
3. Venter, J. C., et al., *The sequence of the human genome.* Science, 2001. **291** (5507): p. 1304-51.
4. International Human Genome Sequencing Consortium, *Finishing the euchromatic sequence of the human genome.* Nature, 2004. **431**(7011): p. 931-45.
5. Chakravarti, Aravinda, *Single nucleotide polymorphisms: . . .to a future of genetic medicine.* Nature, 2001(409): p. 822-823.
6. Stoneking, Mark, *Single nucleotide polymorphisms: From the evolutionary past. . .* Nature, 2001(409): p. 821-822.
7. Rubin, Gerald M., *The draft sequences: Comparing species.* Nature, 2001(409): p. 820-821.
8. Demuth, J. P., et al., *The evolution of mammalian gene families.* PLoS One, 2006. **1**: p. e85.
9. Hallast, P., et al., *High divergence in primate-specific duplicated regions: human and chimpanzee chorionic gonadotropin beta genes.* BMC Evol Biol, 2008. **8**: p. 195.
10. Chimpanzee Sequencing and Analysis Consortium, *Initial sequence of the chimpanzee genome and comparison with the human genome.* Nature, 2005. **437** (7055): p. 69-87.
11. Wheeler, D. A., et al., *The complete genome of an individual by massively parallel DNA sequencing.* Nature, 2008. **452**(7189): p. 872-6.
12. Sundermann, U., S. Kushnir, and F. Schulz, *The Development of DNA Sequencing: From the Genome of a Bacteriophage to That of a Neanderthal.* Angew Chem Int Ed Engl.
13. Kidd, J. M., et al., *Mapping and sequencing of structural variation from eight human genomes.* Nature, 2008. **453**(7191): p. 56-64.
14. Wang, J., et al., *The diploid genome sequence of an Asian individual.* Nature, 2008. **456**(7218): p. 60-5.
15. *Solexa's Progress Is In The Genes* 2006 [cited 2010; Available from: http://www.businessweek.com/magazine/content/06_45/b4008109.htm.
16. *Complete Genomics Drives Down Cost of Genome Sequence to $5,000* [cited 2010; Available from: http://www.bloomberg.com/apps/news?pid=newsarchive&sid=aEUlnq6ltPpQ.
17. Wadman, Meredith, *James Watson's genome sequenced at high speed.* Nature, 2008(452).
18. Granseth, E., et al., *Experimentally constrained topology models for 51,208 bacterial inner membrane proteins.* J Mol Biol, 2005. **352**(3): p. 489-94.
19. Overington, J. P., B. Al-Lazikani, and A. L. Hopkins, *How many drug targets are there?* Nat Rev Drug Discov, 2006. **5**(12): p. 993-6.
20. Bernsel, A., et al., *Prediction of membrane-protein topology from first principles.* Proc Natl Acad Sci U S A, 2008. **105**(20): p. 7177-81.
21. Bernsel, A., et al., *TOPCONS: consensus prediction of membrane protein topology.* Nucleic Acids Res, 2009. **37**(Web Server issue): p. W465-8.
22. Lindahl, E. and M. S. Sansom, *Membrane proteins: molecular dynamics simulations.* Curr Opin Struct Biol, 2008. **18**(4): p. 425-31.
23. Adrienne Kitts, Stephen Sherry, *The Single Nucleotide Polymorphism Database (dbSNP) of Nucleotide Sequence Variation.* 2009.

24. *Growth of Genbank*. Available from: http://www.ncbi.nlm.nih.gov/genbank/genbankstats.html.
25. Galperin, M. Y. and G. R. Cochrane, *Nucleic Acids Research annual Database Issue and the NAR online Molecular Biology Database Collection in 2009.* Nucleic Acids Res, 2009. **37**(Database issue): p. D1-4.
26. Fernandez-Suarez, X. M. and M. K. Schuster, *Using the ensembl genome server to browse genomic sequence data.* Curr Protoc Bioinformatics. **Chapter 1**: p. Unit1 15.
27. Karchin, R., *Next generation tools for the annotation of human SNPs.* Brief Bioinform, 2009. **10**(1): p. 35-52.
28. Brenner, Steven E., *Common sense for our genomes.* Nature, 2007. **449**: p. 783-784.
29. Tatusova, T., *Genomic databases and resources at the National Center for Biotechnology Information.* Methods Mol Biol. **609**: p. 17-44.
30. *Entrez Nodes*. Available from: http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=handbook&part=ch15.
31. *Ensembl Biomart*. Available from: http://www.ensembl.org/biomart/martview.
32. *Entrez Programming Utilities*. Available from: http://eutils.ncbi.nlm.nih.gov/.
33. *Model-View-Controller design pattern*. [cited 2010; Available from: http://publib.boulder.ibm.com/infocenter/wchelp/v6r0m0/index.jsp?topic=/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm.
34. Bauer , C. King G., *Hibernate in Action*. 2005.
35. *Spring framework*. Available from: http://www.springsource.com/developer/spring.
36. *GRAILS*. Available from: http://grails.org/.
37. Wallner, B., P. Larsson, and A. Elofsson, *Pcons.net: protein structure prediction meta server.* Nucleic Acids Res, 2007. **35**(Web Server issue): p. W369-74.
38. Larsson, P., et al., *Assessment of global and local model quality in CASP8 using Pcons and ProQ.* Proteins, 2009. **77 Suppl 9**: p. 167-72.
39. Pauline C. Ng , Steven Henikoff, *SIFT: predicting amino acid changes that affect protein function.* Nucleic Acids Res. **31**(13).
40. *Hibernate Community Documentation*. Available from: http://docs.jboss.org/hibernate/stable/core/reference/en/html/batch.html.
41. *Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls.* Nature, 2007. **447**(7145): p. 661-78.
42. Manolio, Teri A., *How to Interpret a Genome-wide Association Study.* The Journal of American Medical Association, 2008. **299**.