Abstract

The study of human genetic evolution and genetics has been a major area of research since the 80's. The year 2003 saw the successful completion of Human Genome Project with 99% of the genome sequenced to an accuracy of 99.99%. With the advancement in science and the advent of new techniques and computational paradigms, scientists have been able to analyze the genome information on a large-scale population level.

This research focuses on providing statistical evidence to the deep research question and the hypothesis that complementary mutations occur in protein structures. The study maps the mutations on the three dimensional structure space of the proteins so as to find interactions between residues in physical space. "The maintenance of protein function and structure constrains the evolution of amino acid sequences. This fact can be exploited to interpret correlated mutations observed in a sequence family as an indication of probable physical contact in three dimensions" [1]

The process involves high-end data integration by generating information on human genetic variation. The variations found between the genomes of more than 2000 individuals are taken through the 1000 Genomes Project consortium (recent release in 2011). Information from ENEMBL, SUPERFAMILY database, DBSNP, ASTRAL database has been collated together to find mappings of mutations at each level of protein, sequence to structure, formation.

There are very rare studies being carried out where the genome level information is processed and mapped until the protein level. So understanding the three dimensional protein structures and studying relationships between residues on the 3D space was a frontier that had not been fully explored. This research puts forward a benchmark investigation by confirming the existence of complementary mutations. A novel method of mappings of mutations to each level of protein structure formation has been followed. The procedure is highly original as it is unlikely that any other research is being carried out at this level of protein structure and variation analysis.

The extensive information integration involved various risks and challenges, bearing in mind that the data being dealt with is biological in nature and that each element has some characteristics which has to be carefully examined and included at each point in the study.

It is proposed that this study would open a new field of analysis of substantial amount of bioinformatics data on the structural level. This would help researchers infer much informative knowledge from the genomes of different organisms by studying interaction of atoms in the structural space. The study generates information, which can be taken as the base to formulate further studies

The contributions of this research related to the study of *Human Genetic Variation* specifically are:

- Evaluation of the variations between human genomes of approximately 2000 individuals.
- Mapping the large-scale variation information on the coding part (exon level) of the genome.
- Evaluating the information of the proteins on which mutated residues are positioned.
- Mapping variation information on the domain level of the protein and the respective HMM models.
- Moreover, the spatial arrangements of the variations in structure space are evaluated to find physical interactions between the variant residues.
- Statistical testing has been carried out which confirms the existence of complementary mutations on protein level.

[1] *Gobel et al., (1994), Correlated mutations and residue contacts in proteins, Proteins: Structure, Function, and Bioinformatics, 18 , pp. 309–317. (doi: 10.1002/prot.340180402)*

List of Figures

List of Tables

Table of Contents

1 Introduction

"The study of complementary changes of volume and/or size in the interior of proteins has been a classical topic in protein analysis. Although complementary mutations do occur . . . they are not the rule." [1]

- *Lesk and Chothia (Mol.Biol. 136:225-270, 1980.)*

Over the past few decades, there has been an explosive growth in the amount of biological information generated by the scientific society due to the advancements in genomic technologies and molecular biology. This enormous amount of biological data generated, needs to be efficiently stored and processed to extract utmost information from it. This information overflow has steered an utter requisite for computerized databases to store the data, organize and index it, and, for specialized tools to visualize and analyze the data.

Various advancements in the field of computational genomics and bioinformatics have been observed earlier in time. Some key advances have been the discovery of the structure of DNA (Franklin, and Watson Crick), sequencing of the human genome (2003), solving the first protein structure (Perutz and Kendrew), as a result of sequencing technology (Sanger).

The complete sequencing of the Human Genome in 2003 - a significant breakthrough in the field of genetic research, paved way to varied studies relating to the genomes and to comprehend information which can be extracted from the massive data generated. In view of knowledge extraction, the '1000 Genomes Project' has given a milestone contribution in the field, by sequencing the genomes of a large number of people and providing an extensive view of analyzing human genetic variation.

This research focuses to the investigate the existence of complementary mutations in protein structures. The existence of such mutations has been a crucial question in the field of biological sciences, and the project aims to provide some substance to reason their existence by acquiring genome variant data from the 1000 Genomes Project. This is achieved by the challenging task of investigating through the stages of DNA and protein 'sequence to structure' formation, and mapping these mutations onto each of the levels of formation. The study then analyzes the mutational mappings and physical contacts by statistical methods to prove or disprove our initial hypothesis of existence of complementary mutations in protein structures.

This section describes aims and objectives of the project and gives an outline of this report.

1.1 Aims and objectives

This research, addresses a deep research question involving the existence of complementary mutations by annotating individual mutations on 3D protein structures to analyze the results by statistical means. The theory of second-site reversion is a point of significant investigation, which not many researchers have looked into so far, with emphasis on the theory. It is believed that a complementary mutation takes place to reverse the deformation of structure created by a point mutation, and regain gene functionality. By mapping single mutations in one person all the way through to their affect on the spatial arrangement of the atoms in the 3D molecule, that is the protein, this study will compare whether mutations are complementary (i.e. next to each other, and interacting chemically or physically) or not.

AIM: To argue the hypothesis that complementary mutations occur in protein structures and see to what extent can we prove or disprove the hypothesis and bring out further relevant studies associated thorough out the process of investigation.

More specifically, the objectives of this project are:

- To investigate the 1000 Genomes Project data by selecting the right source of the dataset.
- Collection and storage of point variations in the genomes of a population of around 2000 humans.
- Analyze variations and investigate those which occur on exonic regions of the DNA sequence.
- Study each protein and map the variants on the sequence of each protein's domains.
- Finding the spatial Cartesian coordinates of variants in 3D space through ASTRAL database.
- Correlating these coordinates for each domain and calculation of Euclidean distances between the points on 3D structure space.
- Evaluating the statistical results to gain distance versus frequency of distances graph.
- Formulating the results of the discovery and stating them with regard to the concept of existence of complementary mutations.

Overall, this project explores the contribution of the studies on human genetic variation by utilizing the results produced by the 1000 Genomes Project. This data is used to evolve mutation studies from genome level, to sequence level all the way through the different levels of protein formation to finally, the structure level. There are very few studies till date wherein scientists have mapped any amount of biological information on the 3D structure level of proteins. This research aims to make this possible and contribute in the area not only by mapping mutations on the structure but also finding relationships between mutations and their co-occurrence by finding complementary mutations or second site reversions.

The existence of complementary mutations has always been an important question in the scientific community and no one has addressed it yet till date. Instead of following the contemporary practices, we aim to give way to structural level research and variation analysis. It is also unlikely that somebody would cater to the answer before us as there are not many people working with proteins as well as genomic data concurrently.

This research will pave a way for many studies relating to the visualization of whole genome variation analysis in humans. Also, the proposed discussion on the concept of complementary mutations shall help the biological society to look at areas like disease occurrence, inter and intra population studies and variation analysis, much deeply and meticulously and provide a means to investigate novel approaches for personalized medicine. I hope that the idea of locating single amino acid substitutions and their correlations with other mutations on the structure level, will give the scientific community a fresh view of analyzing variations in human genomes.

Consequently, the distinct deliverables of this project are:

The "genome" of any given individual is unique and mapping "the human genome" involves sequencing multiple variations of each gene. The project aims at data analysis and mining of a large set of data and the process involves the following phases, each of which would lead us to a step closer to obtain a basis to reason the initial hypothesis.

- Analyzing and detecting mutations (SNP's) occurring in different humans (data taken from experimental results of the 1000 Genome Project) by comparing it to a reference human genome.
- Position mapping of mutations detected on the exons.
- Position mappings of variations on the individual proteins.
- Position mappings of variations on the individual domains.
- Position mapping of variations on the 3D structure of the domains.
- Presenting statistical inference as a basis of argument of the hypothesis.
- Arguing on the hypothesis and investigating the results to uncover the concept of complementary mutations.

2 Conceptual Knowledge

This section briefly defines the key domains and terminology related to this project. The keywords are given in italics followed by their corresponding definitions. The definitions described aim to help the reader understand the areas of study that this project involves. Also, related work attempts to provide a basis for this project's appraisal. This research deals with information mining and integration of data at the genome level, through the DNA to the protein structure level.

2.1 Overview and definitions

The explosion of information is a major challenge. We need faster, automated analysis to process large amounts of data. There is a need for integration between different types of information (sequences, genomes, annotations, protein levels, RNA levels etc). Also, there is a need for "smarter" software to identify interesting relationships in very large data sets.

***Data* :** Data are any facts, numbers, or text that can be processed by a computer. It can exist in any form, usable or not. It does not have meaning of itself.
***Information*:** The patterns, associations, or relationships among all this data can provide information.
***Knowledge*:** Information can be converted into knowledge about historical patterns and future trends.

Genetics is the branch of biology that deals with heredity and variation. Genomes are modular and thus allow rapid evolution. It is important to have the knowledge of genetics and the processes which occur inside an organism's body to understand or study variations. Biological processes involve molecules that can form complex biological structures. These molecules are present in the organism ultimately due to expression of information residing in the genetic material.

Bioinformatics is the application of *Information Technology* to *Biology* using mathematical and statistical principles. It is the field of endeavor that relates to the collection, organization and analysis of large amounts of biological data using networks of computers and databases (usually with reference to the genome project and DNA sequence information). It provides computational solutions to biological problems and explores the hidden biological information

Chromosomes : The structure in the cell nucleus that contains all of the cellular DNA together with a number of proteins that compact and package the DNA.

The flow of information from a genome level to a protein structure level is depicted below in *Figure 1*. ***DNA*** molecules encode the biological information fundamental to most life forms whereas ***Proteins*** are the primary unit of biological function.

Figure 1: *Flow of information from genome level to protein structure level*

The genome is made up of 30 billion base pairs. It is spread across 22 chromosomes (with an addition of X and Y chromosomes). Not all parts of the genome contain information. The exons are certain coding regions which code for a gene. Genes are encoded through parts of the genome which contain elements of the DNA. The DNA contains information about the proteins and its amino acids. Moreover, these proteins are made up of a chain of amino acid residues. This chain fold up to form a 3 dimensional structure where it is stable and attains the form of its natural existence.

The detailed information content is described in the following sections. The DNA level information is discussed first (Section 1.2) , followed by its information flow into the protein level (Section 1.3). The structural formation of proteins is discussed as this research focuses on dealing with the folded structure level of proteins. Further, the genome level information is discussed about as it consists of a large scale more wider aspect of consideration. The nature of mutations and how evolution occurs, is described at the genome level in section 1.5.

2.2 DNA Level

DNA molecules encode the biological information fundamental to most life forms. *DNA- Deoxy Ribose Nucleic Acid* is made up of a chain of building blocks namely: G (guanine), A (adenine), T (thymine), and C (cytosine). These 4 building blocks are called nucleotides. A n*ucleotide* is composed of three components – a Nitrogenous Base, a Pentose sugar and a Phosphate group.

DNA is a double stranded helix composed of A-T and G-C complementary bases. The opposing strands of DNA are not identical, but are complementary. This means: they are positioned to align complementary base pairs. The order of nucleotide sequences determine which proteins are synthesized, as well as when and where the synthesis occurs.

A DNA sequence contains the 4 nucleotides A, T, C and G in random patterns to form ***nucleic acid sequences***. The human genome contains 3 billion of these base pairs.
Example of a DNA sequence: ...*AATGGTACCGATGACCTGGAGCTTGGTTCGA*...

Genes are sequences of DNA that encode proteins. One gene may be responsible for coding more than one protein. Also there may also be 1 or more genes responsible for the synthesis of a single protein. Genes contain regions of ***introns*** (non coding regions) and ***exons*** (coding regions). The following *Figure 2* explains the assembly of coding parts and the non coding parts in a DNA.Only the coding parts (exons) code for genes and have some functions assigned to them.

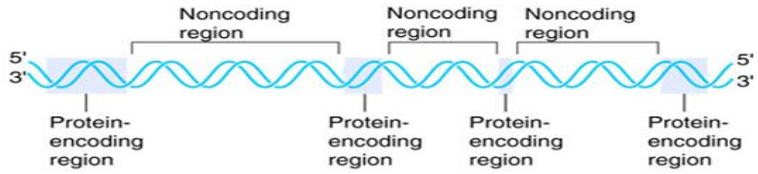


Figure 2: *DNA coding and non-coding information*

Genetic code and Codon pairs: Genetic Code is a list of 3 bases (nucleotides) refered to as codons, which encodes a specific amino acid. There are 20 amino acids involved in protein synthesis and there are only 4 bases in the DNA coding for all the amino acids. The code is a triplet: The triplet code was first suggested by Gammow in 1954. In a triplet code, three DNA bases code for one amino acid. The code is universal: The same genetic code is applicable to all forms of organisms.

2.3 Protein level

Proteins are linear chains of amino acids (a.a.) of differing sequences and lengths. They are polymers of amino acids which evolve into three dimensional structures.

Information in the DNA dictates the sequence of its amino acids. Each a.a. consists of the following elements:

- Amino group (**NH_2**) - amino terminal end
- Carboxyl group (**COOH**) - Carboxyl terminal end
- The central carbon atom C-alpha (**C_α**)
- A variable side chain **R** different for each amino acid.

The structure of an amino acid at the sequence level and its structural level is represented in Figure 3.

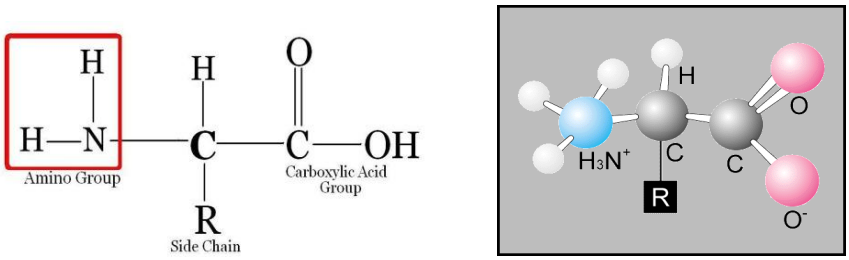


Figure 3: *Backbone of an amino acid residue at sequence(left) and structure(right) level.*

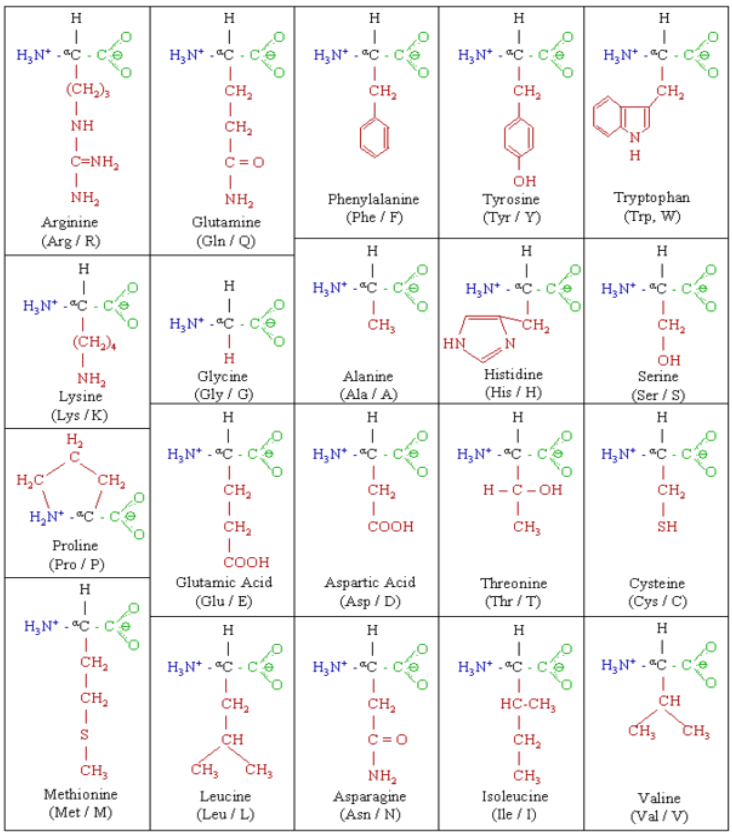


Figure 4: *The 20 amino acids with their one and 3 letter code*

***Amino acids**:* There are 20 standard amino acids namely Alanine, Arginine, Asparagine, Aspartic Acid, Cysteine, Glutamic Acid, Glutamine, Glycine, Histidine, Isoleucine, Leucine, Lysine, Methionine, Phenylalanine, Proline, Serine, Threonine, Tryptophan, Tyrosine and Valine. These amino acids are grouped according to the characteristics of their side chains (R groups). The structures of these amino acids is given in Figure 4.

Protein formation and linkages in peptide bonds

The order of amino acids determines the type of a protein and its structure. Amino acids are joined together by ***peptide bonds.***

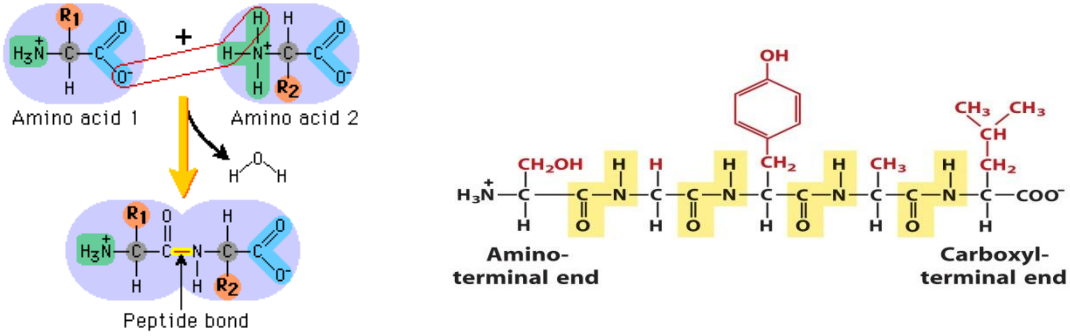


Figure 5: *Formation of a protein chain by peptide bonding between amino acids.*

2.4 Protein structure complexity

This section describes the levels of protein structure formation and its complexity. Since this research focuses on studying the structural level relationships, the structure formation is explained in more detail.

Primary Structure: The sequence of amino acids in the polypeptide chain. It is described as a string from a finite alphabet (AA). The primary structure of a protein contains all the necessary information required for the three-dimensional levels of structure.

Secondary Structure**:** Secondary structure is the spatial arrangement of a polypeptide's backbone atoms without regard to side-chain conformations (R groups). The 3D arrangement is determined by hydrogen bond interactions between adjacent amino acid residues in the polypeptide chain. Polypeptide chain can arrange itself into characteristic repeating patterns such as helices, or pleated sheets. Two basic secondary structures that are stabilized by hydrogen bonds are **α-helix** and **ß-pleated sheet.** They also consist of ***loops*** and turns which help in quickly changing directions and keeping a compact shape of the structure.

Super-secondary structure : Secondary structure elements are observed to combine in specific geometric arrangements known as **motifs or super-secondary structures**. α-helices and ß-strands are combined through loops to make recognizable higher order structures that commonly occur in proteins.

Domain* :** Several motifs usually combine to form compact globular structure called as ***domains. It is generally associated with a function. A domain is considered the fundamental unit of protein structure which is compact, stable and has a specific function.

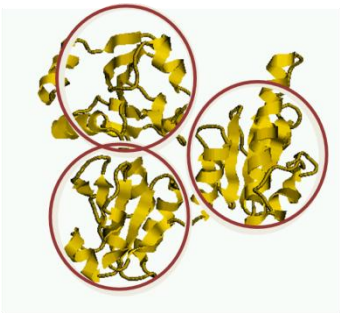


Figure 6: *3 distinct domains in the same protein structure*

Tertiary Structure: It is the folding of the total chain, the combination of the elements of secondary structure linked by turns and loops. Its stability is determined by non-bonding interactions & the disulfide bond. The tertiary structure of proteins is characterized by tightly folded structure with polar groups on the surface and non-polar groups buried.

Quaternary Structure: Spatial arrangement of subunits (2 or more polypeptide chains). It is the overall structure of a protein resulting from the interactions between different subunits in proteins.

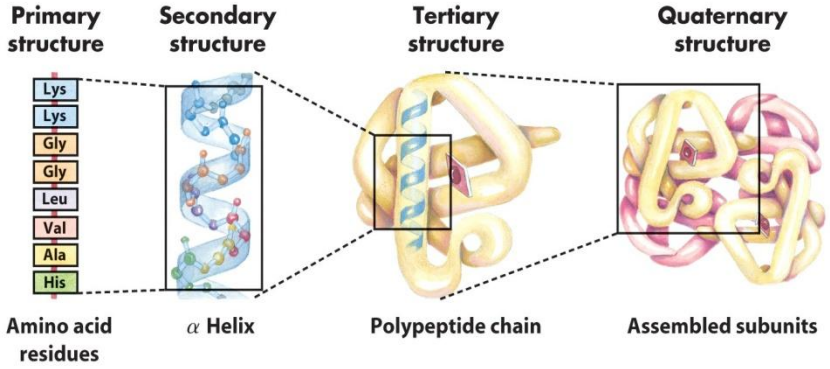


Figure 7: *The levels of protein formation from sequence to structure level.*

2.4.1 Studying protein structures

Scientists who are interested in protein structures can be interested in -
(a) How structures form - FOLDING
(b) What structures are like - APPEARANCE
(c) What structures do, and how - FUNCTION

The following Figures 8, 9 and 10 depict how the protein peptide chain evolve from a sequence level, to folding gradually in space because of bonding and forces and finally evolve into a whole protein structure.

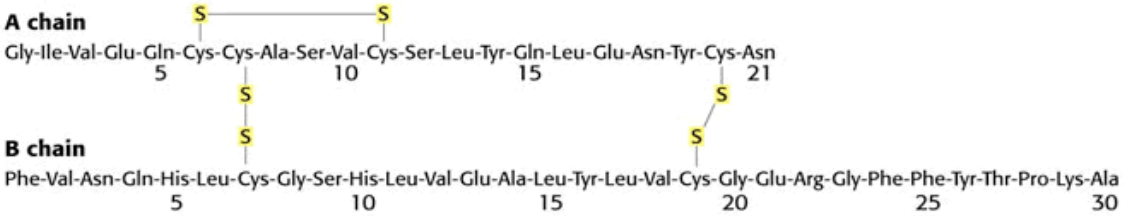


Figure 8: *Sequence level bonding towards formation of a protein*

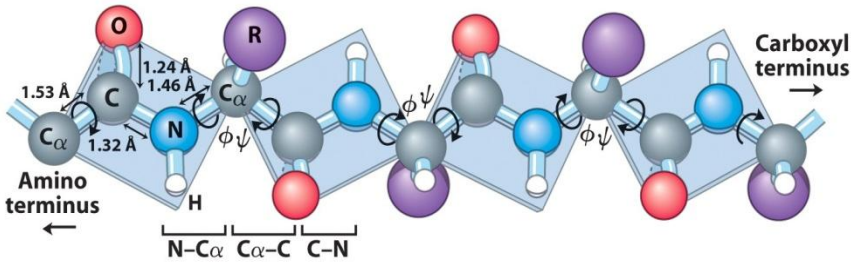


Figure 9: *Visualization of 3D structure of a protein chain*

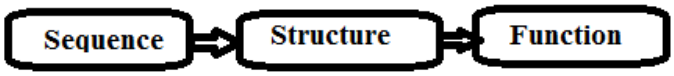
The sudden folding of the protein chain occurs due to forces acting upon the protein residues which facilitate bonding and inter-molecular forces. The torsion angles between atoms of a reside can be seen in Figure 9.

Folding gives rise to three-dimensional structures that are exquisitely adapted to function. Figure 10 gives a view of fully formed protein structures on the 3 dimensional space.

Figure 10 : *Examples of protein 3 dimensional structures*

2.4.2 Sequence, Structure, Function Relationship

Loss of protein structure results in loss of function. A loss of three-dimensional structure sufficient to cause loss of function is called ***denaturation.*** A good protein structure minimizes disallowed torsion angles, maximizes number of hydrogen bonds, maximizes buried hydrophobic AA and maximizes exposed hydrophilic Amino acids.



2.5 Genome level

Mutations (which are any change in the DNA base sequence), occur constantly in all cells and organisms. Offspring's rarely get a perfect copy of the DNA from its parents. But mutations are rare: about 1 DNA base change per 10^9 bases each cell generation. (Humans have about 3 x 10^9 bases). Some mutational changes are much larger: chromosome rearrangements that include genes torn in half and moved to new locations, sometimes combined with other genes.

2.5.1 SNP

An ***SNP*** is a single base change / mutation in a DNA sequence that occurs in a significant proportion (more than 1 percent) of a large population. A Single Nucleotide Polymorphism (SNP) is a source of variation in a genome. SNPs are the most simple form and most common source of genetic polymorphism in the human genome (90% of all human DNA polymorphisms). SNPs are found in coding and (mostly) non coding regions. They occur with a very high frequency of about 1 in 1000 bases to 1 in 100 to 300 bases.

- The abundance of SNPs and the ease with which they can be measured make these genetic variations significant.
- SNPs in coding regions may alter the protein structure made by that coding region.

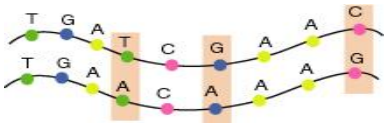


Figure 11 : *Single Nucleotide polymorphism (SNP) mutation among 2 genomes.*

2.5.2 Genetic variation

Mutations are the changes which alter the chemical structure of a gene at a molecular level. Gene mutations or point mutations are the mutations produced by alterations in base sequences of the concerned genes. This can take place either by base substitution, base addition or base deletion.[2] Any two copies of the human genome differ from one another by approximately 0.1% of nucleotide sites (that is, one variant per 1,000 bases on average)[3]

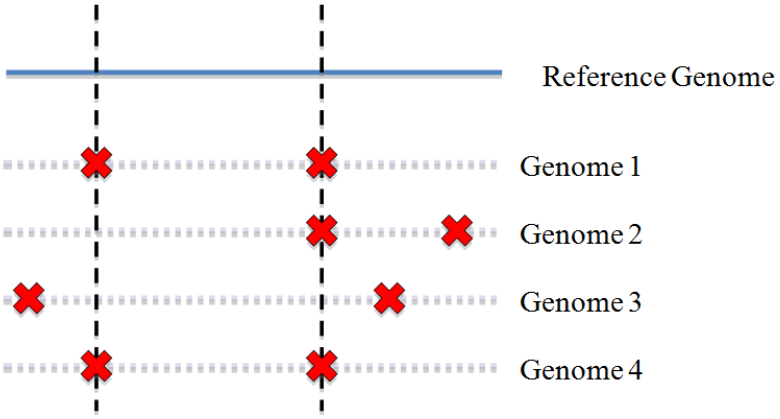


Figure 12 : *Mutations detected on residue positions via alignment of genomes.*

Alignment*:** The result of a comparison of two or more gene or protein sequences in order to determine their degree of base or amino acid similarity. Sequence alignments are used to determine the similarity, homology, function or other degree of relatedness between two or more genes or gene products. ***Figure 12 explains the comparison of 4 genome sequences with the reference human genome and how alignment is used in order to infer the mutational changes in the sequences.

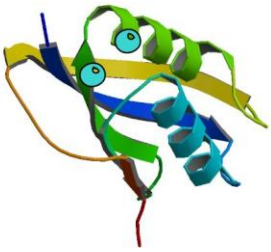
A protein folds into a 3D structure to minimize its free potential energy. Global energy minimum should be achieved to get the right structure. When mutations occur, the sequence is related to the structure and thus the function of the protein, so the structure changes, the energy changes. It is so predicted that the protein tries to re-stabilize the energy by introducing complementary mutations (The hypothesis which is being studied in the project).

2.5.3 Complementary mutations

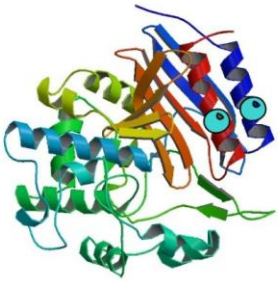
When a mutation occurs in a protein, the change in a single amino acid in the sequence affects the changes in the structure. This change in structure alters the normal functioning of the protein. It is believed that a complementary mutation takes place, at a position very near to the mutated position on the structure, to reverse the deformation of structure created by a point mutation, and regain gene functionality.

2.6 Summary and challenges

The challenge is to find the existence of these mutations (section 2.5.3). As shown in the figures below, the 2 mutations might occur very close to each other on the structure level, but it does not mean that these positions would be adjacent to each other in the sequence.



The structure on the left (PDB ID - 1csq) depicts two mutations occurring very close to each other. It can be well deciphered that these two mutations are also very near on the sequence level (the green chain continues and links both the mutations).



But in other cases, like the structure on the right (PDB ID- 2jqk), even though the mutations seem to be in physical contact with each other on the structure, when observed carefully, these two mutations are actually on the opposite ends of the sequence.

Thus the effect of these mutations are carefully examined from the genome to the protein structure level so as to get precise results and accurate information on their existence. If the mutations are complementary they would be observed to occur next to each other, and interacting chemically or physically on the 3 dimensional structure.

As mentioned, the primary structure of the protein contains all relevant information for the building of the protein structure. It is thus that this study formulates a knowledge based approach by constituting the detailed aspects of each entity and their characteristics while deciphering the results.

3 Background

3.1 Human Genome Project [4]

The Human Genome Project (HGP) is an international scientific research project with a primary goal of determining the sequence of chemical base pairs which make up DNA, and of identifying and mapping the approximately 20,000-25,000 genes of the human genome from both a physical and functional standpoint. [4]

In 2000, a draft about the human genome project was announced which was then completed and accepted in 2003. It is still being worked upon, and further analyses are being published. The project goals were to:

- identify all the approximately 20,000-25,000 genes in human DNA,
- determine the sequences of the 3 billion chemical base pairs that make up human DNA,
- store this information in databases,
- improve tools for data analysis,
- transfer related technologies to the private sector, and
- address the ethical, legal, and social issues (ELSI) that may arise from the project.

The sequencing of the human genome was carried out by Human genome Sequencing Consortium and Celera Genomics run by Craig Ventor. Upon publication of the majority of the genome in February 2001, Francis Collins, the director of NHGRI, noted that the genome could be thought of in terms of a book with multiple uses. After the complete published results of the Human genome project, all genes were "known", scientists now needed to understand their functions!

The next steps to be taken were to elucidate and comprehensively identify the structural and functional components of human genome and to elucidate the organization of genetic network and protein pathways and develop a detailed understanding of the heritable variation in human genome.

The study basically finds a relation between genotypes and phenotypes i.e. the genetic code or the making up of the genome and the physical properties and appearance of an organism. The human genome project gave a basic genome structure of the human genome. This genomic sequence would be used in our research as a reference sequence, against which all other data sequences would be compared to find variations.

The Human Genome:

- The entire collection of chromosomes in each cell of an organism is called a genome.
- Humans have 46 (2n) chromosomes.
- The human genome has about 3×10^9 base pairs and around 25,000 genes.

3.2 Correlated mutations and residue contacts[5]

As stated above, the sequence-structure and function relationship clearly states that if the amino acid sequence varies, the 3D protein structure would take a different shape. (Section 1.1.2)

The publication – "Correlated mutations and residue contacts in proteins" [5], states that correlated mutations do occur in protein structures. The research studies the structural evolutionary constraints and analysis the effects of mutations which occur together when variation occurs. This is one of the reasons which could lead to understanding that complementary mutations occur. For the analysis, each protein family is multiple aligned to study the conserved and the non conserved regions. Then, mutations are studied by firstly, studying their behavior at one particular position throughout the alignment. This is carried out by constructing a similarity matrix for analysis. Secondly, by comparing the mutation behavior at two random positions analyzed by studying the correlation coefficients. Thirdly, the alignments are analyzed for pair contacts and finding contact clusters. This is done by assuming a certain threshold value, which, if exceeded by the mutation correlation factor, the two positions being analyzed would be considered to be in direct contact with each other.

The research mentions Lesk and Chothia stating that even though complementary mutations do occur, they would not necessarily be the rule. [5] Other parallel approaches conclude that it depends on the number of sequences considered, and the range of similarity considered for the sequences that these results may vary.

3.3 Novelty

With the analysis being carried out in this research, we go in depth of this hypothesis by mapping genomic mutations i.e. point mutations – SNP's to predict if the hypothesis is proved or disproved. Also, our approach would be completely different than what has been done before by the following means:
1) The research analysis genomes of a large population (dataset larger and more dependable).
2) The variations are mapped not only on the sequence level or the alignments, but also on the HMM models and the protein structures on 3D space and validated through spatial distance calculations.

There are not many studies being carried out where the genome level information is processed along with the protein level, and if so, till the three dimensional protein structural level and then studying relationships between entities on the 3D space. This research puts forward a benchmark investigation on confirming the existence of complementary mutations. The procedure followed is highly original as it is unlikely that any other research is being carried out at the structure level, and if so, with regard to arguing the existence of complementary mutations.

3.4 Project Protocol

The following figure depicts the research protocol carried out throughout this study which is explained throughout the further chapters.

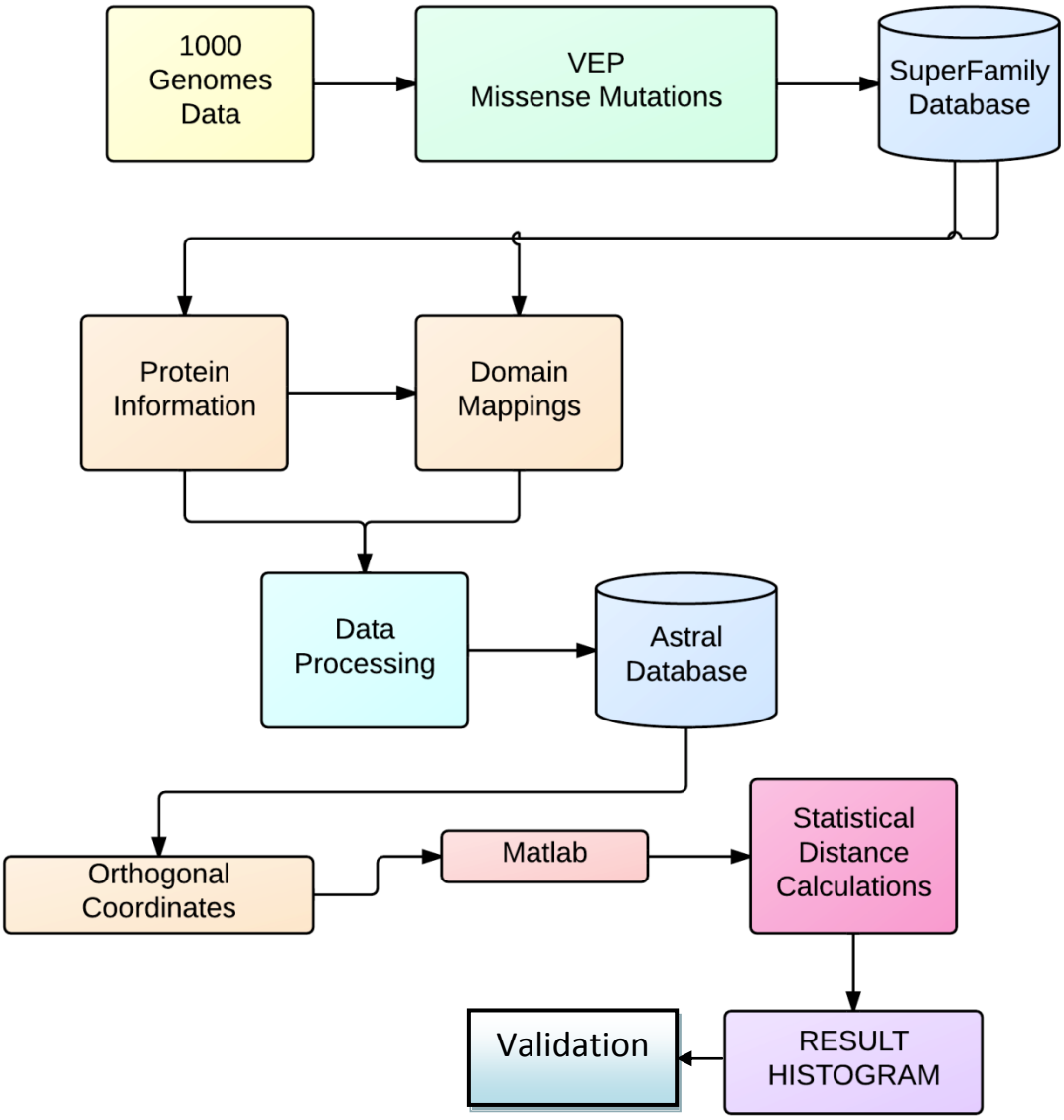


Figure depicts : Flowchart of the Project Protocol

4 Tools and techniques

4.1 Data Mining, Integration and Information Retrieval:

Data mining (also known as Knowledge Discovery in Databases - KDD) techniques are used to retrieve potentially useful information from data. It uses machine learning, statistical and visualization techniques to discover and present knowledge in a form which is easily comprehensible to humans. Also, since we are dealing with a large dataset and the process would yield millions of domains and structures to be interpreted, we need data integration tools in this regard. We refer to databases and online repositories to extract data needed for mapping mutations.

4.2 Database Sources

4.2.1 1000 Genomes Database[6] :

The 1000 Genomes Project is a consortium of scientific researchers around the world who have got together to use DNA sequencing to examine human genetic variations. The goal of the 1000 Genomes Project is to find most genetic variants that have frequencies of at least 1% in the populations studied. This means to find places where our genomes are different. If we take 2 people, our DNA sequences are very long, around 3 billion bases and they are almost the same as each other. But about one every thousand places there is a difference, which makes about 3 million differences between each pair of people.

The people who have been selected to serve as samples for the 1000 genomes project are anonymous individuals, and are selected to be fully representative of their population. There are approximately 2000 individuals, (as reported by the 2011 update) the genomes of which have been taken for the study till now. The project consortium is a publically available resource which can be used by any individual and can be of most value to human geneticists. To make new discoveries, to study human evolution.

The plan for the full project is to sequence about 2,500 samples at 4X coverage. The first set of samples for sequencing includes 1167 samples that already existed or could be collected quickly, from 13 populations, for sequencing in 2010 and early 2011. The second set includes 633 samples that are being collected, from 7 populations, for sequencing in early 2011. The third set, consisting of 700 samples is also available.

The project sequence data allows us to investigate fundamental processes that shape human genetic variation including mutation, recombination and natural selection. The data is present in the form of downloadable MySQL databases and also provided through the FTP mirrored sites. The data not only consists of the genome information of the populations but also gives the mutational information of the amount of variations occurring at respective positions on the chromosomes, genes, transcripts and proteins.

4.2.2 ENSEMBL genomic database

The Ensembl project produces genome databases for vertebrates and other eukaryotic species, and makes this information freely available online [7]. For homo-sapiens (humans): The site provides a data set based on the February 2009 Homo sapiens high coverage assembly GRCh37 (GCA_000001405.6) from the Genome Reference Consortium. The database management system used is MySQL. The data is available both in the form of a web based viewer or downloadable flat files.

It consists of genome annotations (to identify and describe all the physico-chemical, functional and structural properties of genes in a genome) for ORF's (Open Reading Frames), transcription sites, genes, introgenic regions (introns and exons in each gene), splice sites, promoter regions, and it is constantly being updated. Ensembl is a huge data source, of which we can extract all the exonic regions where mutations are detected for mappings in our research. The sites are properly labeled by amino acid residues and numbers i.e. in genomic coordinate systems (e.g. chromosome, clone, contig), annotations are positional. We would take the appropriate site information for each genome and their coding regions.

The Ensembl Perl API
Ensembl uses MySQL relational databases to store its information. A comprehensive set of Application Programme Interfaces (APIs) serve as a middle-layer between underlying database schemes and more specific application programs. The APIs aim to encapsulate the database layout by providing efficient high-level access to data tables and isolate applications from data layout changes. [18]

It displays genomic data by comparing it to the reference genome. The very well structured MySQL database helps the users to not start by organizing the raw sequence, but by simply taking the pre-computed data. Since Ensembl is a huge data repository, we would access this database to get the initial mutation data on the genomic sequences in our research.

4.2.2.1 Variant Effect Predictor [8]

Ensembl provides the facility to predict the functional consequences of known and unknown variants using the Variant Effect Predictor (VEP). There are three primary ways to use the functionality of the VEP: through web interfaces, using standalone perl script, using Ensembl's perl API.

The file contains 38.2M SNPs, 3.9M Short Indels and 14K Deletions.
The three variant types are annotated in the 8th column of the file (INFO) as such
VT=SNP, indicates the variant is a snp.
VT=INDEL, indicates the variant is an indel,
VT=SV, indicates the variant is a deletion.

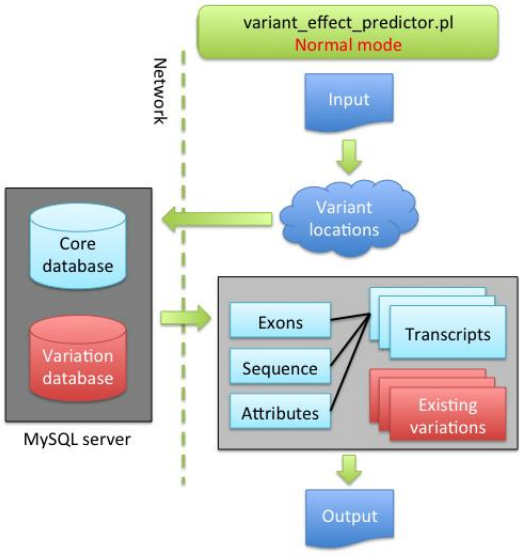


***Figure 13**: The working of the VEP perl script*

This research uses the VEP tool to output information about the mutations acquired through the 1000 genomes database. The perl script is used for this research which returns the output in the form of VCF files. The detailed description of how the tool works is explained in the data integration and collection chapters.

4.2.3 SUPERFAMILY database [9][10]

SUPERFAMILY provides structural, functional and evolutionary information for proteins from all completely sequenced genomes. [10] The database can be used for several applications, one of which is 'Sequence Search'. It finds the domain annotations for DNA sequences or protein sequences over SCOP (Structural Classification of Proteins) [11]. The sequences given are checked to find the domain assignments specific to the input sequences. If the assignment doesn't exist, a BLAST [12] search is carried out against the ASTRAL [13] database. If there is no result found till here, the next step is searching the HMM (Hidden Markov Models) [14] and SAM (Sequencing and alignment Modeling) for the input sequences.

The SUPERFAMILY database would be used in the project to map protein coding regions onto protein domains. The database has data for over 900 organisms, and approximately 60% models which are produced, produce the accurate domain assignments, making the database extensive as well as reliable for our study. Also, the database provides 3D visualizations of the structure of the predicted domains, which would also be used for mapping mutations on the 3D domain structure in the project.

The database also provides the functional annotation of domain superfamilies. This aspect can be made use of if we want to extend our work and look for functional changes taking place due to mutational changes in the protein sequences.

4.2.4 SCOP **[15]**

The SCOP (Structural Classification of Protein) database is a comprehensive ordering of all proteins of known structure according to their evolutionary, functional and structural relationships. The basic classification unit is the protein domain. This database provides a detailed and comprehensive description of the structural and evolutionary relationships of the proteins of known structure. The classification of protein structures in the database is based on evolutionary relationships and on the principles that govern their three-dimensional structure. [15]

4.2.5 Protein Data Bank (PDB) [16]

PDB is an international data repository which contains dimensional structures of protein complexes. The structures are determined by Xray and NMR techniques. It currently consists of 81553 entries. The structures are assigned a PDB ID which is used for accession of the structural data. Also the database is connected to PubMed (Pubmed comprises of approximately 21 million citations for biomedical literature) and thus consist of a PubMed ID through which the related publication can be referred.

Users can download PDB files which have all the coordinates of each atom listed in them for visualization through bioinformatics softwares such as Schrodinger, JMol etc. Also, users can

download FASTA file formats which have the amino acid sequence of the protein. The EC numbers are also stated if there exists an enzyme in the structure. Enzymes are proteins that catalyze (*i.e.*, increase the rates of) biochemical reactions. The Enzyme Commission number (EC number) is a numerical classification scheme for enzymes, based on the chemical reactions they catalyze. As a system of enzyme nomenclature, every EC number is associated with a recommended name for their respective enzyme.

The PDB files are processed and filtered to present formal coordinate files and these are grouped in the ASTRAL database. For this research, the ASTRAL files are consulted to find the coordinates of the respective mutations on 3D space.

4.3 Data Formats

4.3.1 VCF and BAM file formats

1000 Genomes project stores files in VCF file formats. **VCF (Variant Call Format)** is a text file format (most likely stored in a compressed manner). It contains meta-information lines, a header line, and then data lines each containing information about a position in the genome. The VCF format is a tab delimited format for storing variant calls and and individual genotypes. It is able to store all variant calls from single nucleotide variants to large scale insertions and deletions. There is an option whether to contain genotype information on samples for each position or not. More information is available on the 1000 genomes website [http://www.1000genomes.org/about]

4.3.2 PDB file format [16]

Each PDB file for a structure contains a header with the description of which structure it is, has information about when the structure was published and which organism does it belong to. It states the X, Y and Z coordinates for each atom in the structure (each atom would be an individual amino acid in the peptide chain of the protein sequence). It also contains the PSI and PHI angle descriptors, which are the torsion angle values. These are the angles at which each atom is rotated, which states the building of the 3D structure of the protein though a single polypeptide chain. Thus the file contains the whole stereochemistry of the structure. Also, it consists of ligand particles or external hetatoms as well which have been detected during the XRAY and NMR techniques of structure determination.

The PDB files which have been cleaned, processed and approved are kept under the ASTRAL database. Thus, PDB / ASTRAL files mainly consist of the atomic coordinates for standard residues and the occupancy and temperature factor for each atom.

4.4 Statistical Calculations

Euclidean distances are used to calculate distance matrices. Distance matrix gives distances in Å (angstrom) between all atoms. It's main use is in comparison of atoms in 3D structures. Given an

amino acid, the centre of mass is found for each of the two atoms Euclidean distances are calculated between them. The analysis of the interactions between the nearest atoms is carried out to see if they have any relationship such as occurring correlatedly.

The distance formula for N dimensions is given by : $d = \sqrt{\sum_{i=1}^{N} |I_i - J_i|^2}$.

4.5 MATLAB [17]

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java.

MATLAB is used for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

This software is used in this research to calculate statistical relationships between the elements of the coordinates in the 3 dimensional space

4.6 Data Specification

The human genome project has found all human genes and has provided tools to understand their functions. A chromosome has 23 pairs, which means there are approximately 3,000,000,000 (3 billion) base pairs in one single genome sequence. This would correspond to around 30-40 thousand genes in each genome. We in the research would be considering genomes of about 179 individuals. For each of the individual's genome, there are complex exon coordinated which we intend to find and map mutations on. Next we try to map between the exons and proteins. Each exon may be coding for 100's and 1000's of proteins. Out of which we identify the domain regions and then map the mutations on the 3D structures of the domains. A 3D structure has coordinates and torsion angles for each atom or each amino acid present in the protein sequence.

The total amount of data which we are dealing with is gigantic and we are analyzing the modular base level of each mutation i.e. each amino acid, which makes it all the more non-trivial.

DETAILS

The number of missense variants extracted and located on the exons are : **868,139,** which consists of information relating to distinct **81047** proteins.
The variant information processed and collected through querying the SUPERFAMILY database and mapping over the domains: **136,090** records.

5 Data Collection

5.1 Introduction

Collecting the correct type of data for a study is the initial requirement of any research which entails data. The identification of respective sources and evaluating them against each other to acquire the finest data is necessary. To proceed with a data integration project, it becomes all the more vital to identify the right data source which would contain the characteristics of the type of data to be used further in the research.

To collect the data for the Human Genetic Variation Project, we used the following approaches, the characteristics and limitations of which have been described further in this chapter.

5.2 Basic data requirements

- **Data from 1000 genomes project**
  To study human genetic variation, we acquire data coming out from the 1000 genomes project. Details in Section (4.2.1)
  1000 genomes Projects makes available public sequences of DNA's of all individuals.
  Also, it provides the variant data, wherein the variant data, i.e. mutated residues which deflect the normal evolutionary behavior of a reside position in a particular column of amino acid residues, observed between the genomes have been annotated and listed.

- **Exon / coding regions considered**
  A genome consists of various intronic and exonic regions. Since we are concerned with the analysis of protein data, we would look into only the exonic regions, which are regions which actually code for a gene and the gene, therefore, for a protein.

- **Chromosomes 1-22 and X** Humans have a set of 23 chromosomes plus an additional X and Y chromosome. The X chromosome is present much widely than the Y chromosome and in almost all individuals. Thus, while collecting the data, the X chromosome has been considered.

I. Approach 1:

The Human genetic Variation data was acquired from ENSEMBL

To get a list of variants in Ensembl proteins the Ensembl database was referred. The database only includes "known" mutations i.e. novel/individual mutations are included (there are some 1000 Genome mutations which have not made it into the database yet). The database was downloaded locally from the ENSEMBL website-

(http://www.ensembl.org/info/data/ftp/index.html). The tables that were used were the transcript_variation, variation, variation_feature and variation_synonym tables. The database expands to around 37 GB's.

Also, the VCF files for the 1000 Genomes data were downloaded from (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20110521/) - but these files only give nucleotide changes.

Limitation off approach 1:

The data was loaded, preprocessed and viewed to formulate some mappings within the domains. The limitation with this process was that :

- The source data is still the 1000 genomes data, and the mutations are identified but, the database does not have any information about the position of the mutation at any point in the sequence or in the proteins.
- To proceed with the research we need the positions of the mutations so that we can map these residdue points onto the 3 Dimensional structure space.

Thus this idea of extracting the data was discarded and other means of data collection were looked into.

II. Approach 2:

The ENSEMBL also provides an ENSEMBL Biomart wherein the custom database elements can be searched and downloaded. The database details were downloaded from is available at the website :ENSEMBL genome browser: http://www.ensembl.org/Homo_sapiens/Info/Index. The biomart is available at

http://www.ensembl.org/biomart/martview/71c3817f8db4dbb1dafeeebeb8bb574a.

The data was extracted to confirm that users could retrieve all the information related to a variation at the protein and domain level, but te position of the mutation was still not known. Thus this data source was also discarded and we proceeded wth the final selection of data source - Approach 3.

III. Approach 3 - Selected:

Variant Effect Predictor (VEP -http://www.ensembl.org/info/docs/variation/vep/index.html) program can read VCF files and calculate the effects of mutations on Ensembl proteins even when they do not map onto a SNP.

The database:

1000 genomes data was downloaded through the public ftp site, vol1, release 20110521 available at ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20110521/. [6]

The data is accessed through EBI (European Bioinformatics Institute) for European users. This release represents version 3 of the integrated variant call set based on both low coverage and exome whole genome sequence data.

The data was downloaded from the following FTP site on 5th August 2012 - ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20110521/README.phase1_integrated_release_version3_20120430]

5.3 Summary

To take the study forward and continue with mapping of each mutation on the genome at different levels it is essential that the dataset used has all the required information for the processing of the data at further steps. The data processed by Variant Effect Predictor comprises of the detailed position mappings of mutations on each protein in the genome. Thus we take this data for further research.

6 Data Integration

The process of finding a basis of arguing over the existence of complementary mutations is carried out in different phases. The process involves mapping the mutations at each phase and integrating all this data together to finally achieve a perspective which allows us to prove or disprove our hypothesis. This chapter contains information about how the data was extracted, stored, mapped onto each element of the sequence and how the process of amalgamation of the immense information was carried out to attain that basis.

We start with the human genome sequences, extract mutations occurring on each of the genomes, map these mutations on the transcript protein models, next to the protein domains, then to the exact possible position of the residue in the domain 3 dimensional structure, and finally extracting the 3- dimensional Cartesian coordinates of each mutant residue. Visualizing these coordinates, calculating the distances between mutations occurring in the same domain, and interpreting these results through graphical representations is done to achieve results which lead us to deduce the final premise.

6.1 Data extraction

Data from the 1000 Genomes Project (Section 4.2.1) is taken for the project. The 1000 Genomes VCF files were downloaded from the EBI FTP site

ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20110521/ (dated 16.7.2012).

The data consists of whole genomes sequenced for 2500 unrelated individuals across populations. The exon data, or the deep exon coverage data is considered for our research. The data consists of a total of around 2.5 million SNP variants, out of which the missense variants are filtered out for this research by the VEP tool. (Section 4.2.2.1) Data associated with chromosomes 1 to 22 and X has been collected.

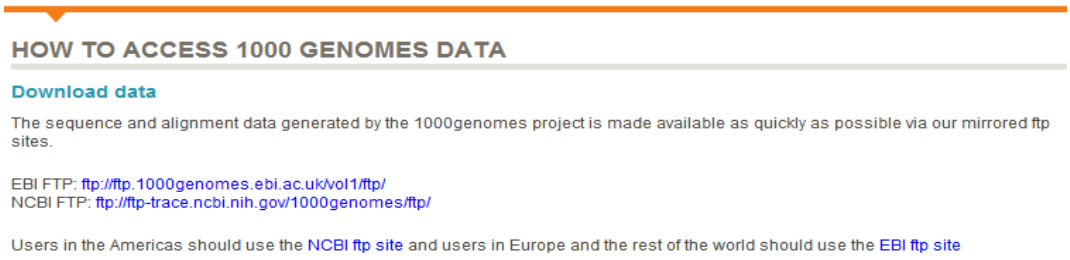


Figure : *Screenshot of data location from 1000 Genomes website*

After collecting the data which entails our data requirements, we now use the tool **Variant Effect Predictor** [8] provided by ENSEMBL [7] to extract the functional characteristic features of the downloaded variants. The standalone perl script (preferable for handling queries which involve processing a large amount of data) is used to run the tool.
The perl VEP script called variant_effect_predictor.pl, is run on the command line to extract information about all the variants downloaded from the raw 1000 Genomes data files.

perl variant_effect_predictor.pl -i [VCF File] -o [Results File] -fork 6 --protein

- **--protein** :
  The --protein parameter adds the corresponding Ensembl protein ID to the output (it reports the transcript ID by default). The protein ID's are recognized by their stable ENSP identifiers.
- **-fork** :
  Forking the process greatly improves the speed of the program.

VEP provides the output with all types of variants namely - SNP's, Indels and Deletions. The command used gives the consequences for all variant ID's in the VCF files, including amino acid changes and position where the mutation leads to one (labeled as missense variants). The raw VFC files have been parsed to extract genotypes relating to missense variants. Refer to Figure 12 for the alignment view of how mutations are detected and aligned on genomes.

The mutated residue, along with its position on the genome sequence and the protein identifier (ENSP number) is returned which are needed for further analysis. The output obtained and its interpretation is explained in the following section 4.2.

6.2 Decoding data

The output file generated by VEP script is taken and the missense variant data is extracted from it. The file is named missense.txt and is saved as a tab delimited text file. The retrieved output file contains information about a total of **868,139** missense variants.

The data attributes are explained below [8] :

1. **Uploaded variation** - as chromosome_start_alleles. This is the variant ID.
2. **Location** - in standard coordinate format (chr:start or chr:start-end); chr stand for chromosome number.
3. **Allele** - the variant allele used to calculate the consequence.
4. **Gene** - Ensembl stable ID of affected gene, denoted by unique ENSG numbers for each gene.
5. **Feature** - Ensembl stable ID of feature, denoted by ENST numbers for each transcript affected.
6. **Feature type** - type of feature. Currently Transcript.
7. **Consequence** - consequence type of this variation - missense_variants only have been filtered.
8. **Relative position in cDNA** - base pair position in cDNA sequence
9. **Relative position in CDS** - base pair position in coding sequence
10. **Relative position in protein** - amino acid position in protein.
11. **Amino acid change** - only given if the variation affects the protein-coding sequence
12. **Codons** - the alternate codons with the variant base highlighted as upper case.
13. **Corresponding variation** - identifier of existing variation (observed as NULL)

14. **Extra** - this column contains *ENSP* - the Ensembl protein identifier of the affected transcript only because while running the VEP script , --protein command was used. This column has identifiers on the basis of the data processing run via the script.

Empty values are denoted by '-'.

The 14 columns and their data (sample 10 rows for depiction) have been depicted below in Table 1a and Table 1b. The complete table with sample rows data can be viewed in Appendix I.

Uploaded_variation	Location	Allele	Gene	Feature	Feature_type
rs186355291	10:286889	A	ENSG00000015171	ENST00000381591	Transcript
rs186355291	10:286889	A	ENSG00000015171	ENST00000535374	Transcript
rs186355291	10:286889	A	ENSG00000015171	ENST00000381604	Transcript
rs186355291	10:286889	A	ENSG00000015171	ENST00000509513	Transcript
10_329338_C/T	10:329338	T	ENSG00000151240	ENST00000280886	Transcript
rs181035038	10:370907	G	ENSG00000151240	ENST00000434695	Transcript
rs189598710	10:390828	C	ENSG00000151240	ENST00000280886	Transcript
10_390969_C/T	10:390969	T	ENSG00000151240	ENST00000280886	Transcript
10_390983_G/A	10:390983	A	ENSG00000151240	ENST00000280886	Transcript
rs12358220	10:402317	T	ENSG00000151240	ENST00000540204	Transcript

Table 1a : *Missense data output from VEP (Columns 1-6)*

Consequence	cDNA_ position	CDS_ position	Protein _ position	Amino _ acids	Codons	Existing _ variation	Extra
missense_variant	1063	810	270	D/E	gaC/gaA	-	ENSP00000371003
missense_variant	326	195	65	D/E	gaC/gaA	-	ENSP00000439587
missense_variant	925	690	230	D/E	gaC/gaA	-	ENSP00000371017
missense_variant	1016	807	269	D/E	gaC/gaA	-	ENSP00000424205
missense_variant	4256	4168	1390	G/R	Gga/Aga	-	ENSP00000280886
missense_variant	393	395	132	M/T	aTg/aCg	-	ENSP00000414462
missense_variant	3462	3374	1125	I/S	aTc/aGc	-	ENSP00000280886
missense_variant	3401	3313	1105	V/I	Gtc/Atc	-	ENSP00000280886
missense_variant	3387	3299	1100	A/V	gCg/gTg	-	ENSP00000280886
missense_variant	1221	997	333	A/T	Gcc/Acc	-	ENSP00000443826

Table 1b : *Missense data output from VEP (Columns 7-14)*

In the first column, the uploaded variation id's are in the format " rs186355291" which are the ID's allotted by dbSNP (Database for single nucleotide polymorphisms SNP's and multiple small

scale variations). DBSNP database is a collection of the list of all variants and allots them a unique identification number starting with 'rs'. It is a public-domain archive for a broad collection of simple genetic polymorphisms. [18]

Where the variant is novel as it has not been identified yet, i.e. no dbSNP 'rs ID' is assigned, VEP automatically creates one in the following format:
[CHROMOSOME]_[POSITION]_[REFERENCE RESIDUE]/[MUTANT RESIDUE]
Eg: 10_329338_C/T

The data clearly depicts the following information for each variant - a) the protein **ENSP** identifiers b) the **position** from which it is known. It is so, that we now know which mutation maps to which protein and at what position. Also, additional knowledge on what the initial nucleotide was and what has it mutated to, is also given in the data..

6.2.1 Specifications

A. For an ENSP number, there are more than one variants recognized. It means that one protein may consist of several variant mutations.

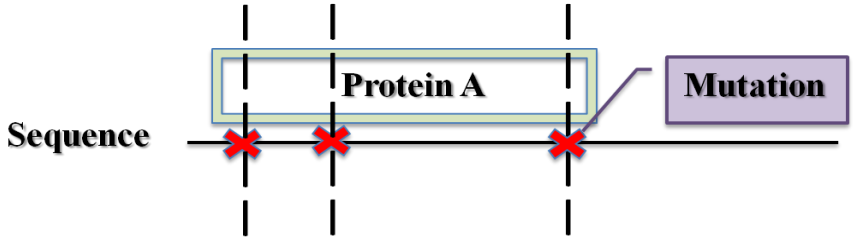


Figure 14 : *A protein can consists of many mutations*

B. For each uploaded variation or SNP there may be multiple transcripts effected , reported via multiple lines). Eg: variation - rs186355291 effects ENSP ID's - ENSP00000371003, ENSP00000439587, ENSP00000371017, ENSP00000424205. It means that a single mutation can be a part of more than one proteins (overlapping sequences). This can be predicted by mapping each variation to the position at the respective protein sequences. **This is depicted in Figure 15.**

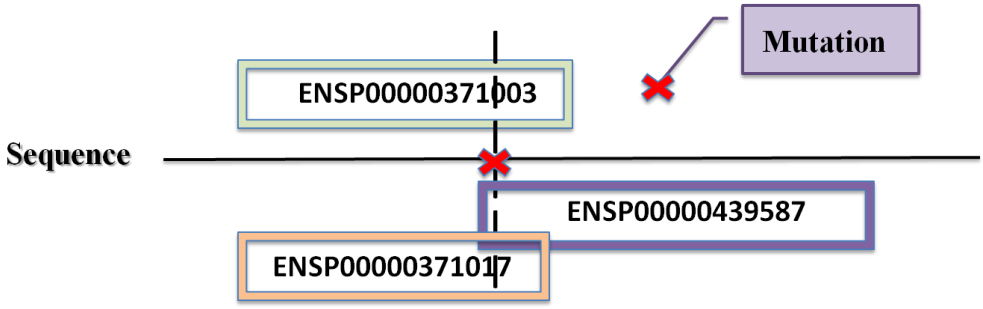


Figure 15 : *A single mutation can belong to many proteins*

These details are carefully considered while extracting the sequence information of each protein ENSP, through the Superfamily Database further in the research.

6.2.2 Significance

The 868,139 variants represent the base information. The position of the variant missense residues are obtained on the proteins and this is noted by the ENSP numbers. These will be mapped by the following steps onto the 3D protein structures:
a) respective proteins,
b) the domains in the proteins,
c) the region in the domain,
d) a residue position on the 3-dimensional protein structure .

6.3 Data storage

The data is stored in MySQL database named as 'HGV'. The MYSQL servers are browsed and managed through the GUI tool **SQLYog**. The codes exected for creating the database and the tables, to load the data into the tables and access remote servers are provided in **Appendix B**.

1. The Human Genetic Variation database named **HGV** is created in MySQL.

2. The table **missense** is created in the HGV database which consists the variant information, the ENSP numbers and the position of the variant residue, obtained via VEP output.

3. The data from the tab delimited output text file is loaded into the table.

Once the data is stored we sight the attribute dependencies. Each ENSP protein consists of one or more variants, and also, variants belong to one or more proteins. This means that the data is highly correlated. Next we process through the data thorough the Superfamily database for domain mappings.

6.4 Information retrieval from the Superfamily Database

The database dump was downloaded, extracted and the database was extracted through the command line on the local machine. The database files were very large in size to have been loaded and executed on the local machine. The MySQL database files were downloaded from the site: *http://supfam2.cs.bris.ac.uk/SUPERFAMILY/howto_use_database.html*.

The SUPERFAMILY database [10] server is located at -
http://supfam2.cs.bris.ac.uk/SUPERFAMILY/

Because of the limited space and resources, the database has been accessed via SSH tunneling on the remote server. The details on how the remote database server has been accessed are given further in this section.

SSH is an encrypted network protocol generally used for remote shell service, secure data communication, command execution etc. In SSH tunneling, a payload protocol is wrapped around the SSH protocol (the delivery protocol) and sent over the network.

The schema of the Superfamily database is given below in Figure 16.

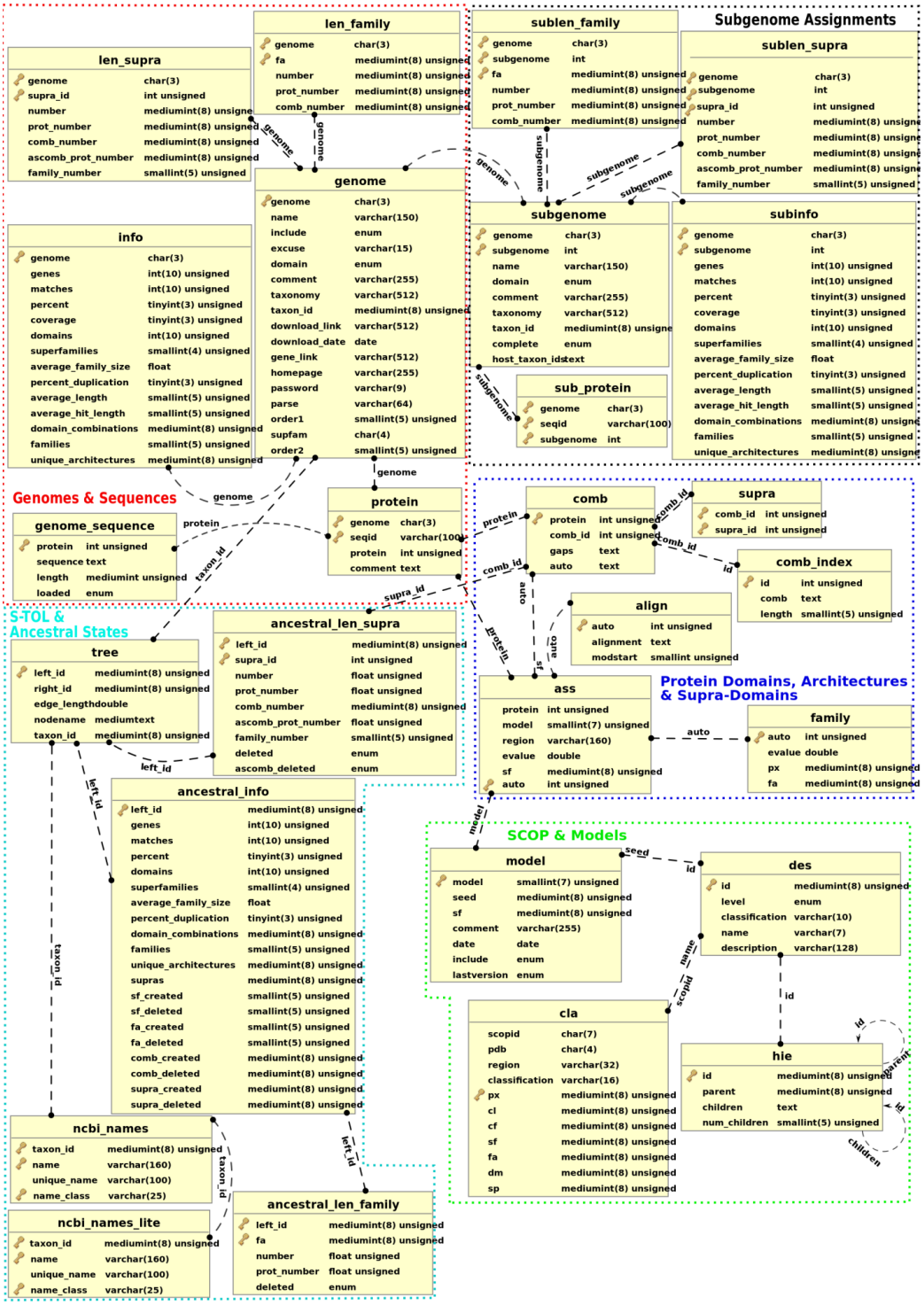


Figure 16 : *Schema of the Superfamily database* [10]

6.4.1 Accessibility [10]

The Superfamily server, is accessible through the Computer Science department at the University of Bristol. An access to the database has been made available for viewing and querying this database with an allotted username and password for this research. The server is available at supfam2@cs.bris.ac.uk The link to the server is *http://supfam2.cs.bris.ac.uk/SUPERFAMILY/* [10] For this research project, the remote database server was accessed via the local machine, through SSH tunneling (NX Client).

By querying this database, for each protein, the domain organizations, sequence alignments and protein sequence details can be viewed and retrieved. The database contains a total of 78 tables, (the description of which is given in Table 2), out of which the main 16 tables can be divided into 4 groups:

- Genomes and sequences - genome, protein, genome_sequence
- Domain statistics - info, len, len_comb, len_family
- Protein domains - ass, family, align, comb
- SCOP and models - des, cla, hie, model, pdb_sequence

TABLES	PROPERTY	DESCRIPTION
align	protein domain	Table containing alignments between the sequences and hidden Markov models.
ass	protein domain	The domain assignments table containing the superfamily level classification for each domain.
cla	SCOP models	Table containing the SCOP domain definitions and classifications.
comb	protein domain	The protein is represented as a sequence of SCOP domains rather than amino acids.
des	SCOP and models	Table containing the description for each SCOP domain.
family	protein domain	The family level classification for each domain, which is contingent upon the superfamily level classification in the 'ass' table.
genome	genome & sequences	General information on each genome; including genome name, taxonomy and download details.
genome_sequence	genome & sequences	A table containing the sequences for all genomes.
hie	SCOP and models	The SCOP database uses a hierarchical classification scheme. This table contains the parent and children nodes of each SCOP entry.

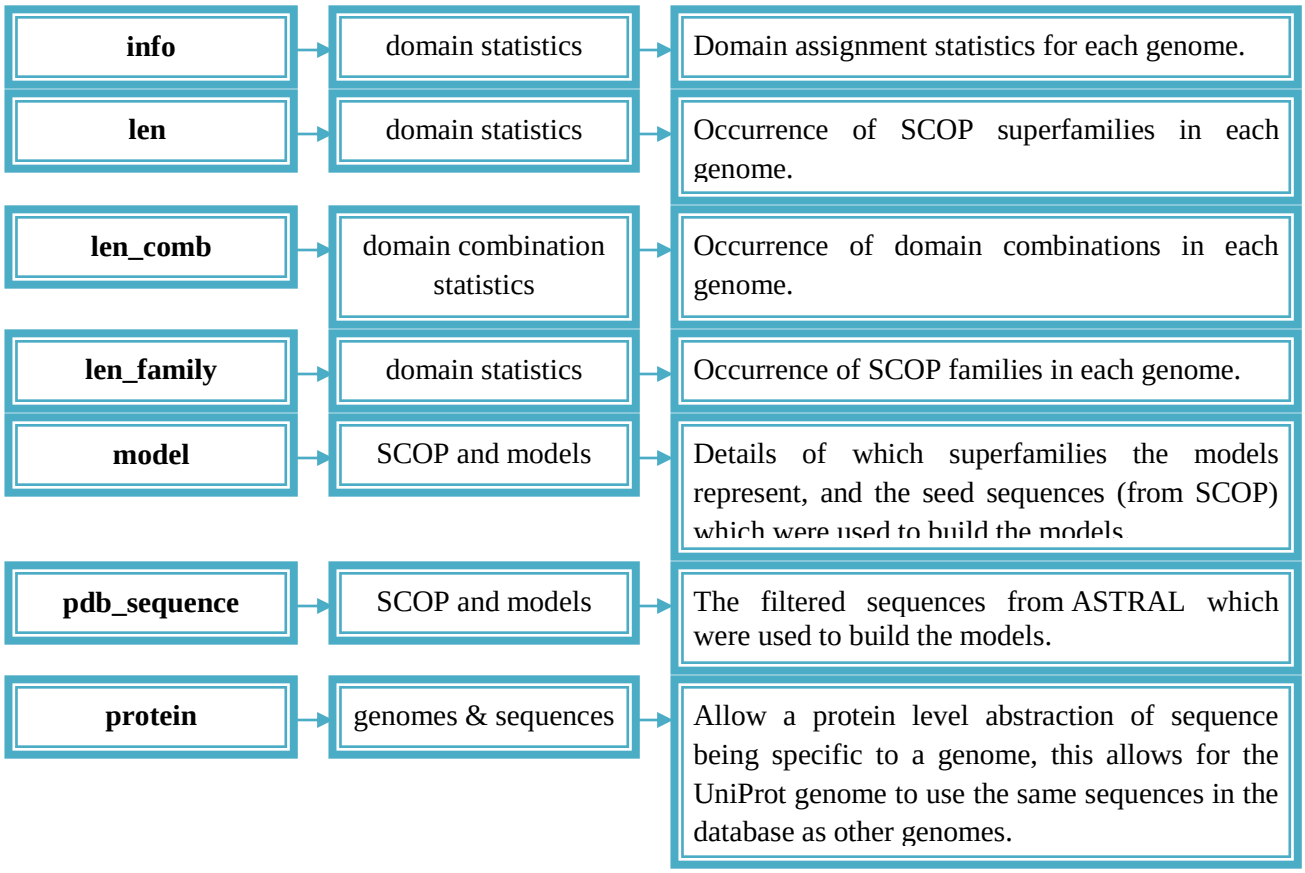


Table 2 : *Description of the individual tables in the Superfamily database*

6.5 Mapping variations from protein to domain level

The relational database (Superfamily) details and specifications of attribute in each table is studied to process and extract the information relevant to this research. The mutations from the exonic regions of the genome are already derived , mapped with their respective protein ENSP identifiers.

Now to further map these mutations on the protein domains, the **Superfamily Database** [10] is used. The database annotates structural protein domains at the SCOP (Structural Classification of Proteins) superfamily level (Refer Section 4.2.3). This information has to be retrieved to get the domain level mappings.

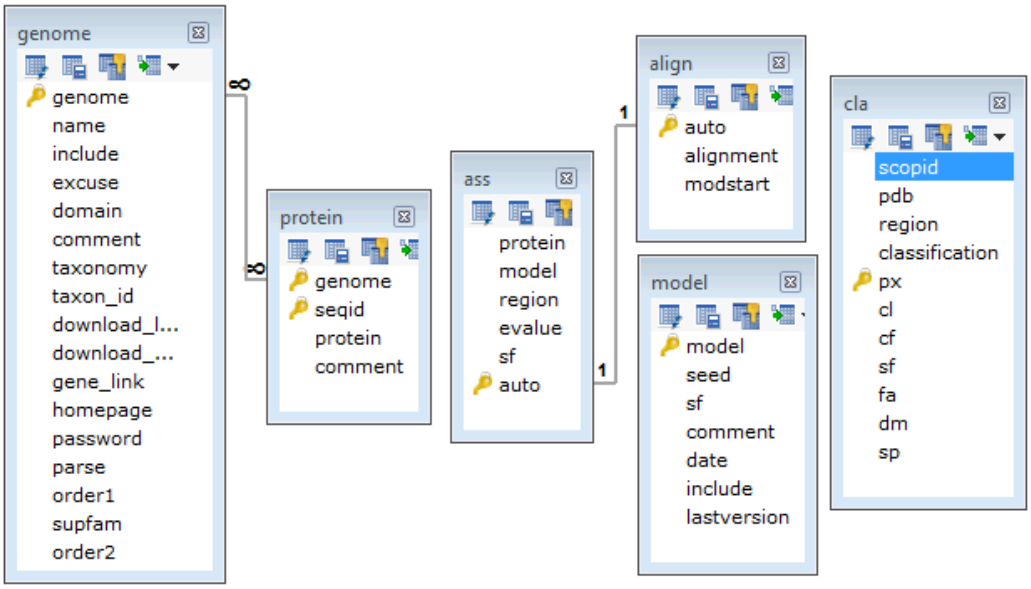


Figure 17 :*Model schema of the database tables from which data is extracted (part of the Superfamily Database)*

6.5.1 Input query

The data collected from VEP has the variant information, and for each variant mutation, the protein identifier (ENSP) and the position of the mutation in the respective protein sequences. These 2 attributes are taken as the fundamental keys to search the information and extract related attributes about the each protein in the Superfamily database. Following are the mappings between the interrelated tables used for data extraction:

The attributes have been mentioned in italic font and the table names in bold letters.

A. From the **protein** table, the *genome* is selected as the human genome to specify the organism of study - termed as 'hs' in the database.

B. The ENSP identifiers specific to the dataset are matched with the sequence ID - "*seqid*" of the **protein** table in the Superfamily database.

C. The *protein* attribute in the **protein** table is the same as the *protein* attribute in the **ass** table.

D. Now, we have the protein ENSP mappings on the domain assignments table - **ass**.

a) The *protein* column is the protein id used in the **protein** table. This gives the access to the genome project specific *seqid* column.

b) The **ass** table includes the attribute *evalue* which has hits with E-values between the values of 0 and 1. E-value is a statistical calculation based on the quality of alignment (the score) and the size of the database. The value of the attribute *evaule* in the **ass** table has been adjusted to a value less than or equal to 0.0001 for efficiently querying the results being extracted out of the database.

c) The *model* from the **ass** table is extracted for further analysis. *Model* represents the model responsible for this assignment.

d) The column *region* in the **ass** table is a comma-separated list of regions. The two numbers separated with a hyphen are the begin and end positions in the sequence of the assignment region. The list of regions together make up the single domain represented by this entry, e.g. 65-84. This column is extracted while querying.

e) The column *auto* in the **ass** table relates to the *auto* column of the **align** table. This *auto* column is for indexing purposes. It defines each protein domain uniquely. "All" the data (columns) in the **align** table is extracted to process the data further for this research.

E. The *model* in the **ass** table matches the *model* values of the **model** table. The *model* column is the identifier for the hidden Markov model, and corresponds to the filename of each model, e.g. 0034782.

F. We now have the mappings till the data in the **model** table. The *seed* column is the SCOP identifier for the protein seed sequence. It is observed that, the *seed* column relates to *px* in the **cla** table.

G. The query has reached the **cla** table, from which we extract the *scopid* by linking the *seed* of the **model** to the *px* in the **cla** table.
a) The *px* represents the SCOP domain entry.
b) The *scopid* contains the SCOP style identifier, e.g. d1ffve2 .

Thus, in a crux, from the Figure of the schema of tables concerned, (Figure 17), the following relationships were built in order to query and extract information.

- Missense.ENSP = Protein.seqid
- cla.px = model.seed
- protein.protein = ass.protein
- ass.auto = align.auto
- ass.model = model.model
- ass.evalue < = 0.0001
- genome = 'hs'

6.5.2 Extracted information

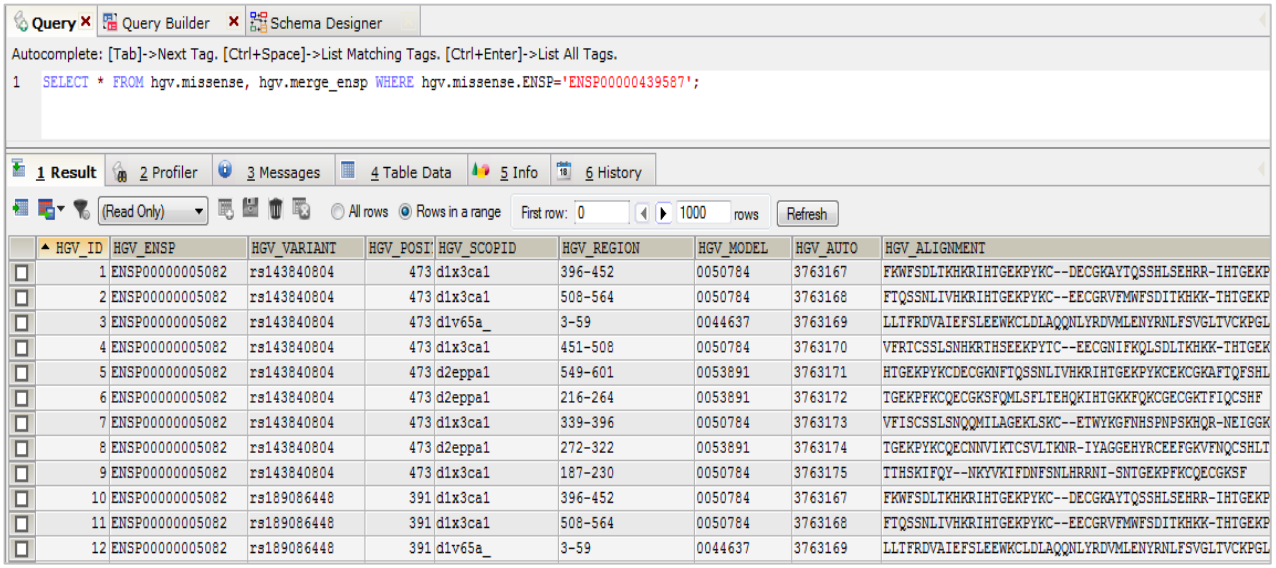
Through the process of selection of attributes, the following information is extracted:

For **each ENSP** (from our initial dataset) :
a. The *model* it corresponds to.

b. The *region* of the domain denoted by numbered positions, separated by commas if the domain is spread at more than one regions in the sequence.

c. The *scopid* of the protein. This gives us the domain names as ID's from the SCOP database.

d. The *alignment* sequence of the concerned domains, the model start number and the auto number are also extracted from the align table. The importance of these attributes and how essential their contents are, will be studied later in this report.


Query | Query Builder | Schema Designer

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->List All Tags.

```
1  SELECT * FROM hgv.missense, hgv.merge_ensp WHERE hgv.missense.ENSP='ENSP00000439587';
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info | 6 History

(Read Only) | All rows | Rows in a range | First row: 0 | 1000 rows | Refresh

HGV_ID	HGV_ENSP	HGV_VARIANT	HGV_POSI	HGV_SCOPID	HGV_REGION	HGV_MODEL	HGV_AUTO	HGV_ALIGNMENT
1	ENSP00000005082	rs143840804	473	d1x3ca1	396-452	0050784	3763167	FKWFSDLTKHKRIHTGEKPYKC--DECGKAYTQSSHLSEHRR-IHTGEKP
2	ENSP00000005082	rs143840804	473	d1x3ca1	508-564	0050784	3763168	FTQSSNLIVHKRIHTGEKPYKC--EECGRVFMWFSDITKHKK-IHTGEKP
3	ENSP00000005082	rs143840804	473	d1v65a_	3-59	0044637	3763169	LLTFRDVAIEFSLEEWKCLDLAQQNLYRDVMLENYRNLFSVGLTVCKPGL
4	ENSP00000005082	rs143840804	473	d1x3ca1	451-508	0050784	3763170	VFRTCSSLSNHKRTHSEEKPYTC--EECGNIFKQLSDLTKHKK-THTGEK
5	ENSP00000005082	rs143840804	473	d2eppa1	549-601	0053891	3763171	HTGEKPYKCDECGKNFTQSSNLIVHKRIHTGEKPYKCEKCGKAFTQFSHL
6	ENSP00000005082	rs143840804	473	d2eppa1	216-264	0053891	3763172	TGEKPFKCQECGKSFQMLSFLTEHQKIHTGKKFQKCGECGKTFIQCSHF
7	ENSP00000005082	rs143840804	473	d1x3ca1	339-396	0050784	3763173	VFISCSSLSNQQMILAGEKLSKC--ETWYKGFNHSPNPSKHQR-NEIGGK
8	ENSP00000005082	rs143840804	473	d2eppa1	272-322	0053891	3763174	TGEKPYKCQECNNVIKTCSVLTKNR-IYAGGEHYRCEEFGKVFNQCSHLT
9	ENSP00000005082	rs143840804	473	d1x3ca1	187-230	0050784	3763175	TTHSKIFQY--NKYVKIFDNFSNLHRRNI-SNTGEKPFKCQECGKSF
10	ENSP00000005082	rs189086448	391	d1x3ca1	396-452	0050784	3763167	FKWFSDLTKHKRIHTGEKPYKC--DECGKAYTQSSHLSEHRR-IHTGEKP
11	ENSP00000005082	rs189086448	391	d1x3ca1	508-564	0050784	3763168	FTQSSNLIVHKRIHTGEKPYKC--EECGRVFMWFSDITKHKK-THTGEKP
12	ENSP00000005082	rs189086448	391	d1v65a_	3-59	0044637	3763169	LLTFRDVAIEFSLEEWKCLDLAQQNLYRDVMLENYRNLFSVGLTVCKPGL

Figure 18: *Information of domain, region, model and alignment present for each protein in database.*

6.5.3 Consequence

The information of all 'domains' which exist in a 'protein' has been retrieved, and the places on the sequence- 'regions', at which they are found to occur in the sequence ('alignment'). (Section 6.5.2)

As mentioned, the Superfamily database is available on a remote server which has been accessed through SSH tunneling. Since the rights of altering the database or its tables is not given to any individual other than the developers, we fetch the information required for formulating the study results by storing the information into the local database "**hgv**".

The full code on how the data has been fetched, the JDBC connection established and the output stored in a file has been given in **Appendix C - C.1.**

The extracted result set is stored in a tab delimited file. The contents of this file are then loaded into a new table in the ***hgv*** database with the table name as "*supfam_ensp_rs*". The structure of the table is shown **as below in figure:**

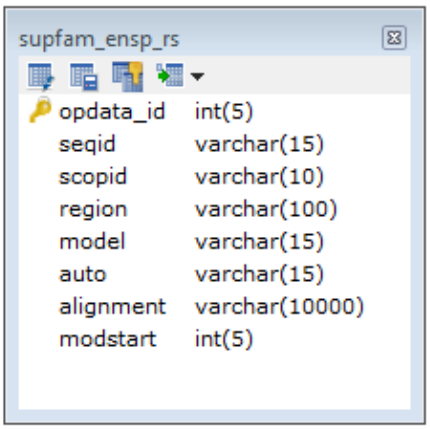


Figure 19 : *Table supfam_ensp_rs and its attributes*

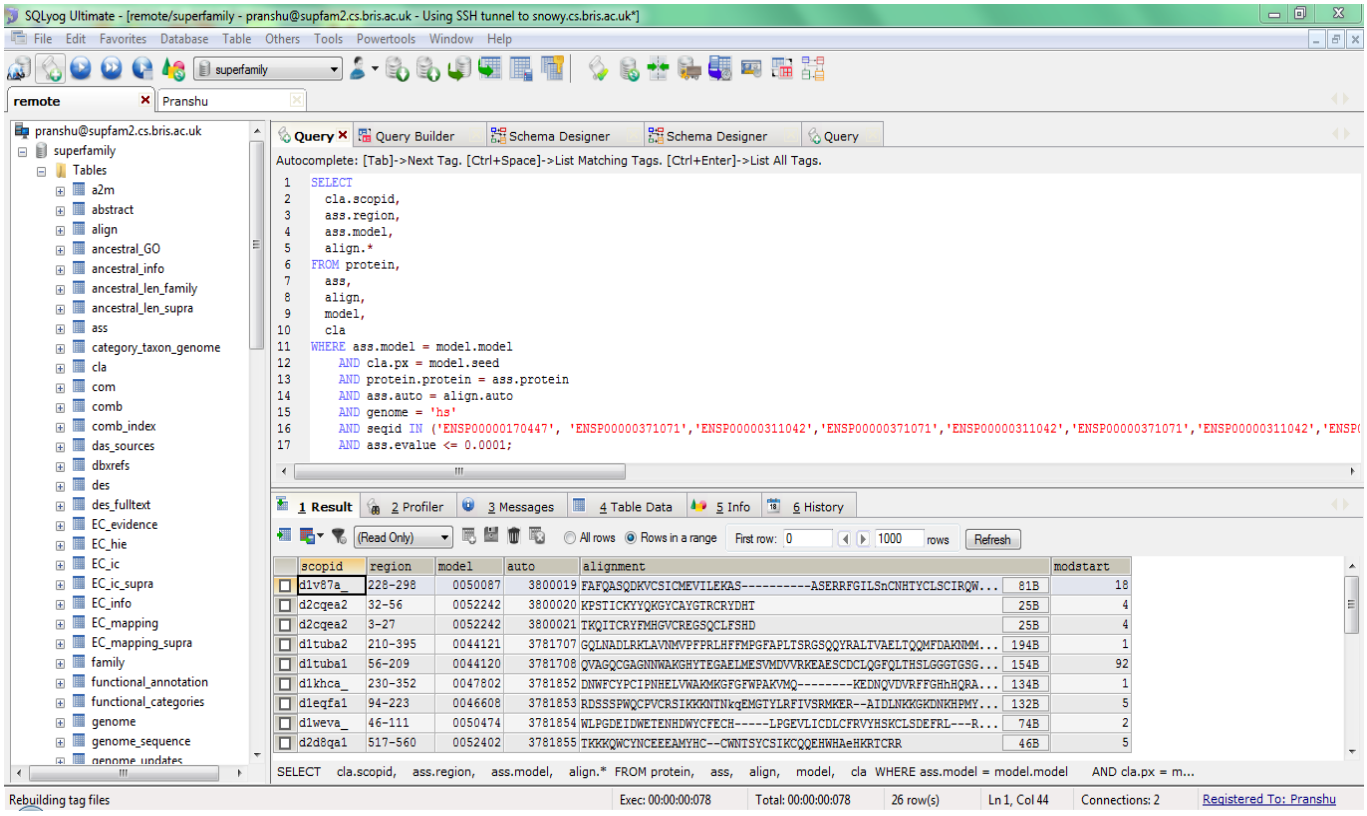


Figure 20 : *Screenshot of query run on Superfamily database (By providing set of ENSP numbers)*

6.5.4 Assigning domains to the variants

Next, the data retrieved from the Superfamily database and the missense data is merged together to map the variations on their respective positions on the domains. For this, the *missense* table and the *supfam_ensp_rs* table are merged together to form a new table, with all the data collated together, called *merge_ensp*. The structure of the table can be viewed in Figure 20.

- The new table is created on the basis of same ENSP numbers i.e. **missense.ENSP** equates to **supfam_ensp_rs.seqid** for data acquisition.
- Unique ID's are allotted to each tuple in the ***merge_ensp*** table, as the data is highly dependent on each other and a single column cannot be considered unique to identify each tuple in the table.

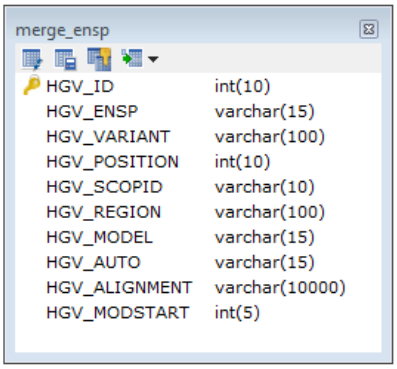


Figure 21: *Table merge_ensp and its attributes*

To extract information from the database, we pick out the attributes we need for further investigation in the later stages of the research and store these attributes in a separate table in the MYSQL database on the local machine. The following are the table built in the local database out of which **missense_fulldata** has the initial variant information extracted through VEP, **supfam_ensp_rs** contains result set retrieved from Superfamily database and **merge_ensp** has relevant data from these two tables merged together on the basis of protein identification numbers.

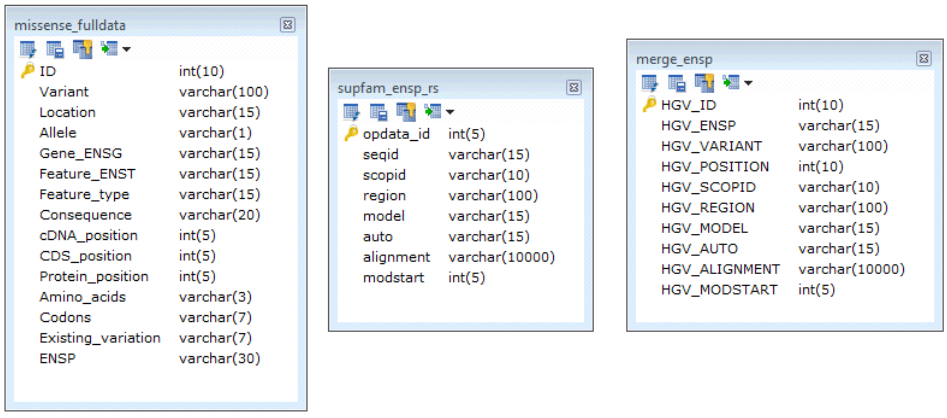


Figure 22: *Tables in local 'hgv' database.*

Data properties and complexity

It is noted that :
i) A protein can contain more than one domain
ii) A domain can exist in more than one region (spitted into parts) and not as one single localized entity
iii) A single mutation can be mapped to more than one domains due to domain overlapping.
iv) The domains may share some parts of protein sequence

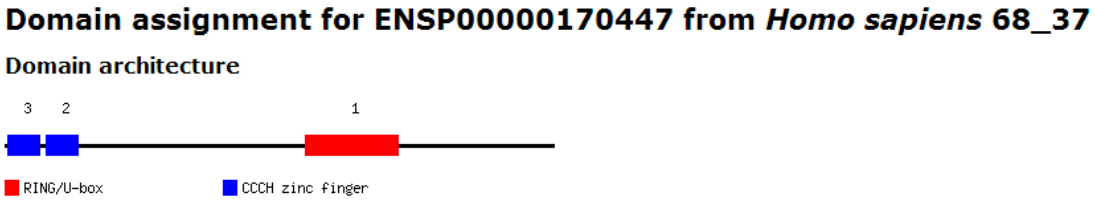


Figure 23: *Screenshot of view of domain assignment from Superfamily database*

6.5.6 Summary

A protein is a combination of domains. It is important to know the domain structure for the functional analysis of the proteins and study areas like structure prediction. , structural genomics.

This section studies how information about the proteins i.e. their domains, and protein sequence (alignment) and the positions of the residues are mapped onto them. The next Section 6.6 deals with further challenges in the mappings and how the research resolves the data issues related with conceptual knowledge.

6.6 Mapping variations on models and extracting coordinates from ASTRAL

Section 6.5, discussed the successful mappings of mutations from proteins to their domains by extracting domain relevant data for each protein. But, is the information retrieved accurate? This section studies the steps taken to process the data generated, by :

- ✓ finding the valid entries of data wherein the position actually matches to a region in the domain
- ✓ locating the position of the mutant residues from the sequence, to the HMM models by studying each sequence (alignment) and its properties.
- ✓ For the processed data, extracting the position specific coordinates from the ASTRAL database using their SCOP Id's (domain name information).

The code for these processing steps is available in **Appendix C : C2.**

6.6.1 Processing valid positions within region

A part of the data extracted from the *merge_ensp* table has been depicted in Table 3. The sample view contains the attributes HGV_ID, HGV_ENSP, HGV_VARIANT, HGV_POSITION, HGV_SCOPID, HGV_REGION, HGV_MODEL. The contents of these attributes have been described in Section 6.5.2.

HGV_ID	HGV_ENSP	HGV_VARIANT	HGV_POSITION	HGV_SCOPID	HGV_REGION	HGV_MODEL
1	ENSP00000005082	rs143840804	473	d1x3ca1	396-452	0050784
2	ENSP00000005082	rs143840804	473	d1x3ca1	508-564	0050784
3	ENSP00000005082	rs143840804	473	d1v65a_	3-59	0044637
4	ENSP00000005082	rs143840804	473	d1x3ca1	451-508	0050784
5	ENSP00000005082	rs143840804	473	d2eppa1	549-601	0053891
6	ENSP00000005082	rs143840804	473	d2eppa1	216-264	0053891
7	ENSP00000005082	rs143840804	473	d1x3ca1	339-396	0050784
8	ENSP00000005082	rs143840804	473	d2eppa1	272-322	0053891
9	ENSP00000005082	rs143840804	473	d1x3ca1	187-230	0050784
10	ENSP00000005082	rs189086448	391	d1x3ca1	396-452	0050784
11	ENSP00000005082	rs189086448	391	d1x3ca1	508-564	0050784
12	ENSP00000005082	rs189086448	391	d1v65a_	3-59	0044637
13	ENSP00000005082	rs189086448	391	d1x3ca1	451-508	0050784
14	ENSP00000005082	rs189086448	391	d2eppa1	549-601	0053891
15	ENSP00000005082	rs189086448	391	d2eppa1	216-264	0053891
16	ENSP00000005082	rs189086448	391	d1x3ca1	339-396	0050784
17	ENSP00000005082	rs189086448	391	d2eppa1	272-322	0053891
18	ENSP00000005082	rs189086448	391	d1x3ca1	187-230	0050784
19	ENSP00000005082	11_3381057_C/G	371	d1x3ca1	396-452	0050784
20	ENSP00000005082	11_3381057_C/G	371	d1x3ca1	508-564	0050784
21	ENSP00000005082	11_3381057_C/G	371	d1v65a_	3-59	0044637
22	ENSP00000005082	11_3381057_C/G	371	d1x3ca1	451-508	0050784

Table 3: *Data view of the* ***merge_ensp*** *table in* ***hgv*** *database and some of its attributes.*

6.6.1.2 Analysis of characteristics of the acquired data

Following are the characteristic points of the acquired data (pointed out in Section 6.5) with reference to the information content in **Table 3**. All the points stated provide a view of the context by visualizing Figure 24.

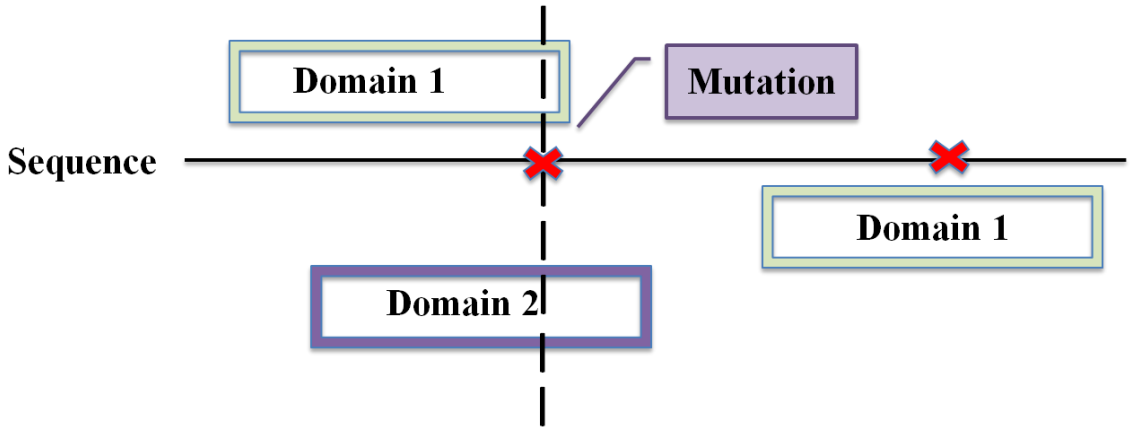


Figure 24: *Complexity of proteins, domains and mutation occurrence*

1. One protein can have more than one entries , each with unique characteristics. Eg: "ENSP00000005082" is present in all 22 rows depicted in the table.

2. Relationship between a proteins and its domains:

a) A protein (ENSP) may consist of more than 1 domains in itself. ENSP00000005082 contains 3 domains - d1x3ca1, d1v65a_, d2eppa1. The concept can be visualized by *Figure 24*.

b) It is also noted, that a single domain can be present at different parts of the protein sequence, i.e. Domain D1 is spread across region 396-452, 508-564, 451-508, 339-396, 187-230. Refer Figure 24, domain 1 is present at two separate regions.

3. Domains are not necessarily allotted specific unique regions of existence. Domains share parts of the sequence depicting linkages and inter-dependence of residues shared between them. *Figure 24* shows overlapping domains - Domain 1 and Domain 2.

4. A single domain may have multiple point mutations. Example: Domain D1, in region 508-564 has occurrences of three point mutations at positions 473 and 391 and 371.

5. Similarly, a mutation (at point 473), can be a part of more than one domain. This usually occurs when the domains are overlapping (explained in point (c))

6. The model for each protein's domain is different. Even if it may seem in the same table depicted above that the model numbers are redundant, it has been found and concluded that the alignment of the sequence is different in each case.

7. The mutation at a position is uniquely named and depicted in the GV_VARIANT column as either the dbSNP "rs ID's" or by the uniquely allotted VEP ID's, allotted by VEP if the mutation is not yet reported into the dbSNP database. (Refer Section 6.2)

Example: In Table 3, the position 473 in the protein ENSP00000005082 has been allotted the Variant ID - "rs143840804" and position 391 in the same protein sequence, has been allotted the Variant ID - "rs189086448".

6.6.1.3 Data filtering on the basis of position and assigned region in a protein domain

A. In Table 3, it is observed that many records contain positions of variants which are not within the start and end limits of the region specified for that very record. For instance, position 473 does not lie in the regions specified for ID's 1,2,3,5,6,7,8 and 9. This position maps only to the region 451-508 present in domain d1x3ca1.

Introducing this parameter and filtering the result set on the basis of the valid region areas where the position of the variation lies, reduced the data size and has given us the valid position mappings which are present in the sequence.

Another aspect taken into consideration while filtering out the relevant data for use in further research, is the multiple region values. The following table, Table 4, depicts some sample records from the **merge_ensp** table.

HGV_ID	HGV_ENSP	HGV_VARIANT	HGV_POSITION	HGV_SCOPID	HGV_REGION	HGV_MODEL
10410	ENSP00000278833	11_62380920_C/T	56	d1g8qa_	149-178,209-262	0037997
20120	ENSP00000309577	rs146720224	2038	d1p3ja1	2093-2128,2206-2322	0048689
20121	ENSP00000309577	rs149468678	2063	d1p3ja1	2093-2128,2206-2322	0048689
20283	ENSP00000310227	11_46766091_G/A	1854	d1u6gc_	602-691,721-788,856-1064,1186-1423	0049802

Table 4 : *Specific records from merge_ensp table for finding valid positions within regions.*

The norms relating to deciphering the multiple region data are explained below:

B. The region "**149 -176, 209-262**" in the domain "d1g8qa_" implies that the domain is spread across regions of the sequence present at and in between residues 149 to 176 and also, between residues 209 to 262. However, while extracting domain information and calculating valid positions to be taken for further analysis, we decipher this region as -

The region starts at residue 149 and end ends at residue 262 of the domain "d1g8qa" and protein "ENSP00000278833 ". The reason behind this theory of domain regions being split into specific parts of the sequence and not in a continuous chain of residues is an interesting biological phenomenon. A domain is formed due to the interaction of certain residues which join together to form an independent functioning unit in the protein.

As depicted in above 3D protein structure diagrams in this report, it is not the continuous sequence amino acids which group together to form parts of the structure, instead, the 3D structure is formed with the folding of the protein chain and collective residues residing close to each other form a subpart. These parameters have been defined while processing the result set for extraction of the residue information from the protein sequence. The code is given in **Appendix C : C2** Introducing this parameter, reduces the data size by a substantial difference.

6.6.2 Calculation of residue position in the model

Once we have the filtered data with the valid results where each position does map to a region within the domain, we now map this position to the position of occurrence of mutation on the Hidden Markov Model. The models have been assigned unique numbers in the database and the information of this attribute is retrieved in the **merge_ensp** table (merge_ensp.*HGV_MODEL).*

Till the present stage, the calculations have been carried out on the sequence level basis, and mappings were done onto the sequence (*alignment* column) to integrate, merge, filter and thus process the pure result set. The information extracted till the present stage of the research consists of values, out of which 4 attributes are proposed as inputs for calculation of the next mapping step i.e. the calculation of the modstart number to find the position of the residue on the HMM model.

The following attributes are considered for calculation of residue position in the model:

- Alignment *(hgv.merge_ensp.HGV_ALIGNMENT)*
- Position (hgv.*merge_ensp.HGV_POSITION*)
- RegionStart *(hgv.merge_ensp.HGV_REGION)*
- ModStart number (hgv.*merge_ensp.HGV_MODSTART)*

6.6.2.1 The protocol followed

For each protein record (valid entry - filtered in the previous section), we have an alignment sequence, a position of the variation on the sequence, a region start number and a modstart number. The relevance of these attributes is stated below:

Alignment is the sequence alignment for the entry in the ass table. Some residues in the alignment are in lower case letters, some in upper case. Upper case residues are matching (depicting a match position or conserved residue), and lower case residues are not matching (depicts that there might be either an insertion or a deletion occurring at that position), and thus not involved in the assignment. The sequence also consists of hyphens " - " depictig gaps between the sequence alignments.

Position is the position of the variation occuring on the sequence of the protein. This is the initial position value which was extracted out of the VEP script with the ENSP identifiers.

RegionStart is the start position of the region calculated in the previous section . This is only calcualted for the variations, the position of which lies in between the defined region. The starting position of the region is then termed as *RegionStart*. Example: if the region reads [45-52, 78 - 92] and the position is 50, then the *RegionStart* value shall be 45 and *RegionEnd* value 92.

ModStart is the position that the model starts from. This position might not be 1 in most of the cases. The model starts from a pre-defined position in the sequence, and it is counted forward each position in the sequence by adding the *modstart* number prior to going ahead in the sequence.

6.6.2.2 To calculate the position on the model

For each record in the *merge_ensp* table, we consider the *alignment* and note the *position* of occurrence of mutation.

scopid	position	region	alignment	modstart
d2cqea2	8	3-27	TKQITCRYFMHGVCREGSQCLFSHD	4

For the above record: Position = 8, RegionStart = 3, ModStart = 4.
(This is a filtered record wherein the position (8) falls between the region (3-27)).

We need to calculate the output ***ModelPosition***.

We start with the *RegionStart* number and count each element in the alignment which is an Upper Case or a Lower Case letter. The algorithm continues to count only the letters until the *Position* is reached (here 8). In Figure 25 :

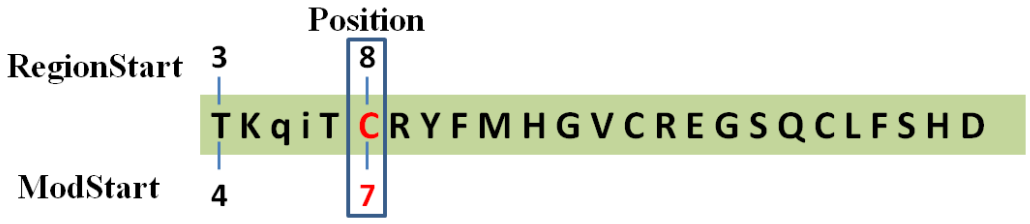


Figure 25: *Simplified view of calculation of* ***ModStart*** *from Alignment*

1) *Positon* number 8 would be counting letters only, starting at *RegionStart* 3. We store this character and the Position.

2) A check is imposed to make sure that the retrieved character at the stated position is an Upper Case letter (depicting a match state).

3) We then count the ***ModelPosition*** by counting only the Upper Case letters and the "- " until we reach the position of the residue where RegionStart = Position (Step 1). This gets us to a position that maps to a position in the model. The position in the model would be 4 plus the counting to the same position (Position of character "C"). i.e. 4 + (3 uppercase letters) = **7**.

Thus, the norms of querying the Superfamily database with the attributes region and ModStart are as follows:
a) The *RegionStart* value is increment only by counting the letters in the alignment (Upper case and Lower). The letters represent residues in the sequence, which is a continuous chain of amino acids.
b) The *ModStart* value is incremented only by counting the Upper Case letters and the "-". The model constitutes either the match states or the gaps (if any).

6.6.2.3 Complexity

While calculating the ModelPosition, there are countable cases noted when, the character found at the site where RegionStart = Variant position, is not an upper case / match state.
To overcome this problem, we introduce a parameter of finding the nearest match state by introducing a window size of 5 on either side of the resultant non-match residue.

As an example, say a variation is recorded to occur at *Position* 473 ; the *ModStart* is reported as 18 and the *RegionStart* has been extracted as 452.
The *alignment* is considered as:

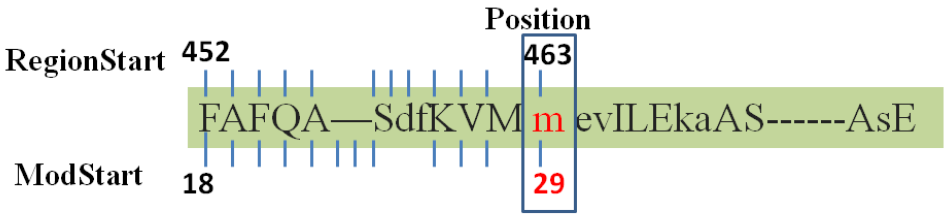


Figure 26: *Advanced calculation of* ***ModStart*** *from Alignment*

1) We start with the *RegionStart* position (position 1 in sequence (residue F in the above case) depicts *RegionStart* value of 452), and count only letters till we reach Position 463 (Residue "m").

2) The ***ModelPosition*** is calculated by starting with *ModStart* as 18 and counting Uppercase letters and '-' till we reach the column with residue "m".

- If the residue read "M" or an upper case letter for the match state, we would extract the ***ModelPosition*** as 29.
- Since it reads a mismatch state, we introduce a window size of 5 characters on each side, and the match state found nearest to the present ***ModelPosition*** is returned and its position extracted.
- Thus, in the above case, the ***ModelPosition*** would be returned **28** with the residue information "M" (found on one left of the target column. The residues on the right still read small case letters for 2 columns).

6.6.2.4 Significance

The calculated position of the residue in model - ***ModelPosition*** is the position from which we extract the 3 dimensional coordinates from the ASTRAL files. In the next section, Section 6.6.3, we would find the X, Y and Z coordinates of this ***ModelPosition*** by running our integration algorithm on the respective domain file (scopID) for each record.

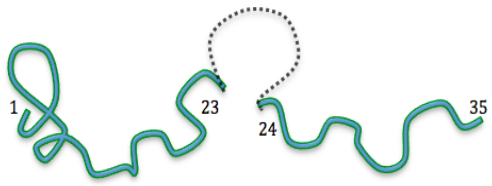


Figure 27: *Deciphering the sequence and model positions*

By examining Figure 27, it is noted that although the sequence continues after residue 23 to form a loop, these loops are not stable and contain no valid information. Thus these portions of the loops are not sequenced. These regions are generally depicted by group of " - - " in the sequences. Thus the next residue match state after 23, in the figure, would be 24 (by neglecting the loop portions). This is why the study aims at calculation of model positions (match / mismatch state positions) from the sequence positions to get a precise position of the residue so as to continue with coordinate extractions.

This step provides us with information of the domains and positions of the variations occurring on the domain models. In the next section, it is explained how we take this as a source data and extract the 3D spatial coordinates of each variant residue.

6.6.3 Spatial coordinates extraction

Mapping the variants on the 3D spatial plane , which is the protein structure level, is the next challenge in this research. A facility which provides the structural view of the biological data is required for accomplishing this task. The Protein Data Bank (PDB) archive is the single worldwide repository of information about the 3D structures of large biological molecules, including proteins and nucleic acids. [16] The (PDB) format provides a standard representation for macromolecular structure data derived from X-ray diffraction and NMR studies. The ASTRAL compendium provides databases and tools useful for analyzing protein structures and their sequences. It is partially derived from, and augments the SCOP: Structural Classification of Proteins database. Most of the resources provided here depend upon the coordinate files maintained and distributed by the Protein Data Bank. [16]

The PDB-style files for SCOP domains are downloaded from the website ***http://astral.berkeley.edu/pdbstyle-1.75.html***. The filtered and processed files are stored in a database file system called the ASTRAL database. The data was downloaded on September 2, 2012. The total size of the downloaded zipped ASTRAL files is approximately 5 GB's.

The files are unzipped and extracted. The extracted files consist of domain ".ent" files addressed with the SCOP domain names of the domain information they contain.
Each file has the PDB file format, as discussed in Section 4.4.2.

6.6.3.1 Reading data from ASTRAL File

A sample ASTRAL file for domain - d1a0ha1 is shown below: (Filename : d1a0ha1.ent)

```
HEADER    SCOP/ASTRAL domain d1a0ha1  [44655]     02-JUN-09   0000
REMARK  99
REMARK  99 ASTRAL ASTRAL-version: 1.75
REMARK  99 ASTRAL SCOP-sid: d1a0ha1
REMARK  99 ASTRAL SCOP-sun: 44655
REMARK  99 ASTRAL SCOP-sccs: g.14.1.1
REMARK  99 ASTRAL Source-PDB: 1a0h
REMARK  99 ASTRAL Source-PDB-REVDAT: 17-JUN-98
REMARK  99 ASTRAL Region: a:164-270
REMARK  99 ASTRAL ASTRAL-SPACI: 0.15
REMARK  99 ASTRAL ASTRAL-AEROSPACI: 0.15
REMARK  99 ASTRAL Data-updated-release: 1.61
ATOM      1  N   SER A 164     105.392  37.839   8.083  1.00100.00           N
ATOM      2  CA  SER A 164     106.457  37.425   7.119  1.00100.00           C
ATOM      3  C   SER A 164     106.480  38.412   5.965  1.00100.00           C
ATOM      4  O   SER A 164     105.439  38.697   5.365  1.00100.00           O
ATOM      5  CB  SER A 164     106.206  36.007   6.578  1.00 92.35           C
ATOM      6  OG  SER A 164     107.214  35.607   5.661  1.00 85.17           O
ATOM      7  N   PRO A 165     107.675  38.917   5.620  1.00 98.96           N
ATOM      8  CA  PRO A 165     107.897  39.884   4.539  1.00 98.08           C
ATOM      9  C   PRO A 165     107.916  39.210   3.178  1.00 97.45           C
ATOM     10  O   PRO A 165     107.212  39.614   2.249  1.00 96.99           O
ATOM     11  CB  PRO A 165     109.289  40.449   4.861  1.00 97.71           C
ATOM     12  CG  PRO A 165     109.569  40.001   6.275  1.00 94.62           C
ATOM     13  CD  PRO A 165     108.931  38.660   6.335  1.00 96.54           C
ATOM     14  N   LEU A 166     108.692  38.135   3.104  1.00100.00           N
ATOM     15  CA  LEU A 166     108.884  37.364   1.878  1.00100.00           C
ATOM     16  C   LEU A 166     107.772  36.321   1.649  1.00100.00           C
ATOM     17  O   LEU A 166     108.040  35.207   1.189  1.00100.00           O
ATOM     18  CB  LEU A 166     110.260  36.671   1.918  1.00100.00           C
```

Figure 28: *Sample view of the ASTRAL file (d1a0ha1.ent)*

Each ASTRAL domain file contains atomic coordinates for standard residues and the occupancy and temperature factor for each atom.

- The file consists of a HEADER and REMARK section which contains the description and metadata about the information contained in the file.
- ATOM records for proteins are listed from amino to carboxyl terminus.
- Nucleic acid residues are listed from the 5' to the 3' terminus.

The core information is stored in the rows with records starting with "ATOM". The information stored in each column is described as:

1) Column 1 - Record name "ATOM" depicts that the row contains information about an atom or a residue.
2) Column 2 - Atom serial number.
3) Column 3 - Atom name , C-alpha / O or N depending on the atom found.
4) Column 4 - The three letter amino acid code for the residue.
5) Column 5 - Chain identifier (an entry can contain more than 1 chains)
6) Column 6 - Residue sequence number
7) Column 7 - Orthogonal **coordinates for X** in Angstroms
8) Column 8 - Orthogonal **coordinates for Y** in Angstroms
9) Column 9 - Orthogonal **coordinates for Z** in Angstroms
10) Column 10 - Temperature factor
11) Column 11 - Element identifier

6.6.3.2 Protocol of extracting coordinate information

The information needed to extract the coordinates has been calculated in the previous section, Section 6.6.2. From the **merge_ensp** table, we extract the *scopId* information and the position returned as ***ModelPosition*** and define a function which, for each record, fetches the coordinates from the file which related to the *scopId* domain.

NOTE: Domains in the file hierarchy are named according to **SCOP** ID, and not **ASTRAL** ID

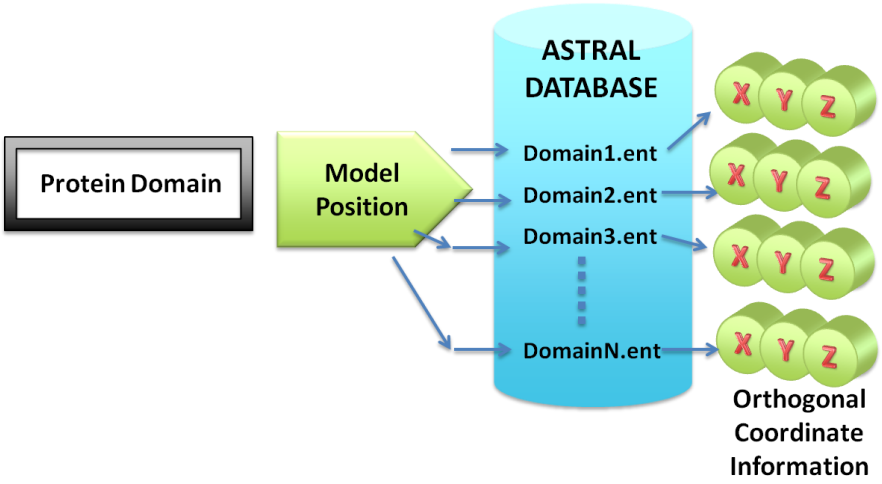


Figure 29 *: Extraction of coordinates from ASTRAL database*

The method of extraction followed is sequentially described as:

I. For each entry in the **merge_ensp** table, the domain name (fetched from scopID attribute) equals to the name of the ASTRAL file with an extension of ".ent". The path till the location of the ASTRAL file is defined and the contents of the file are read.

II. For the rows which contain the "ATOM" element, the residue sequence number (Column-6 in the ASTRAL file) is matched with the *ModelPosition* in the **merge_ensp**

table record.

III. a) There are more than one rows in the file which contain the residue sequence number equal to the *ModelPosition*. This number depicts the amino acid in the sequence. In the Figure 29, if the *ModelPosition* = 164, then the rows narrowed down to corresponding reidue sequence number would be the first 6 ATOM rows (rows 13 to 18 in the whole file).

b) From the list of these rows, we consider the row with the atom "CA". CA represents the C-alpha carbon atom of the residue. This atom is considered as C-alpha atom forms the backbone of all amino acids . (Figure 3 in Section 2.3) Also, considering the C-alpha atom would mean considering the centre of mass of the amino acid residue.

IV. The row confined to the residue sequence number and the C-alpha atom has now been selected (Row 2 for ATOM elements in the sample file). We now extract the X, Y and Z coordinates (Column 7, 8 and 9) from the same row.

V. A check is imposed to observe that the amino acid residue (column 4 in ASTRAL file) is the same as the missense variant information stored in the MySQL *missense* table in **hgv** database.

This process of extraction produces the orthogonal coordinates X, Y and Z for each domain entry and *ModelPosition*. These coordinates can now be plotted on 3-dimensional plane and would serve as points with characteristics of their individual domain and protein information's.

NOTE: In some cases, the residue in sequence being found from the file, is not present in the ASTRAL file. These are cases where the ASTRAL files lack full information of the domain architectures or there may be occurrence of random loop regions at those positions which cannot be taken as static.

PROTEIN ID	X- COORDIANTE	Y- COORDIANTE	Z- COORDIANTE
ENSP00000311042	-61.053	57.511	-28.945
ENSP00000328808	-55.442	43.872	-17.169
ENSP00000311042	-55.442	43.872	-17.169
ENSP00000280886	55.203	137.072	36.558
ENSP00000280886	68.258	163.113	18.418
ENSP00000280886	65.525	166.885	48.145
ENSP00000280886	55.066	172.428	46.415
ENSP00000280886	43.052	150.354	52.695
ENSP00000280886	26.620	135.125	40.348
ENSP00000280886	31.961	156.367	34.821
ENSP00000280886	48.993	155.513	30.174
ENSP00000280886	37.851	156.774	18.592
ENSP00000280886	60.517	139.459	23.748
ENSP00000280886	64.000	147.418	18.770
ENSP00000280886	45.468	178.352	34.043
ENSP00000326128	15.242	25.146	17.486
ENSP00000354040	1.665	32.695	72.208

Table 4 : CoordinateOutput *File with coordinates of mutations in each protein (Part of the table)*

Each row in the result ***CoordinateOutput*** file is a point on the 3D space with property of belonging to the protein in the first column. (Refer Table 4). The result gives us **16,498** coordinates of mutant residues mapped to their respective proteins. Each point depicts the centre of mass of the residue which has been found as a missense variant throughout the course of investigation. The values of the orthogonal coordinate positions on 3D space are given in Armstrongs.

6.7 Summary

This chapter dealt with is the most crucial part of information control and data integration. The flow of information and storage of data together to make sense out of it is a challenge. From deciphering the contents of the initial dataset and paving a way out from the sequence level till the structure level, and finally extracting the coordinates of mutations on the structure, required an efficient method. After many hits and trials at each step, the project protocol was finally chosen as the one explained in this chapter. All the challenges related to the characteristics of biological data have been dealt with and incorporated in the research.

After performing the tasks specific to this chapter - Data integration, the following have been delivered:

- ✓ Mutation mappings on proteins
- ✓ Mutation mappings on domains
- ✓ Mutation mappings on HMM models
- ✓ Position of residues in model calculated
- ✓ Data extracted from ASTRAL database
- ✓ Coordinates retrieved and linked to their specific proteins.

7 Statistical Calculations

With the completion of the data integration phase, many detailed information features are disclosed. From extracting the mutations on the human populations, to mapping them on their genomes, then to individual proteins and then their respective domains, till extracting the mappings of these mutations on the 3-dimensional structure of a protein space has provided us with data which can be used to statistically investigate our initial hypothesis.

With the information about the position of mutations on the 3D space we now calculate the degree to which these mutations occur together. It is perceived that if complementary mutations were to occur, they would occur at positions very near to each other in the structure. This premise has been explain in this study while explaining the nature of complementary mutations.

7.1 Methodology

The data integration phase has provided with the coordinate dataset wherein each coordinate can be associated as a mutation on a domain or a protein level. These coordinate information are used to find the interactions between the residues (here, points on 3D space) by calculating Euclidean distances between the points which belong to the same protein.

The distance between two points is the length of the path connecting them. In Euclidean three-space, the distance between points (x_1, y_1, z_1) and (x_2, y_2, z_2) is given by the following formula :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}\ .$$

Thus, for each protein in the result set, if the protein contains information about more than one mutation, or, in other words, if there are more than one existences of coordinate elements for the same protein, we calculate the Euclidean distances between then using the above formula.

To proceed with this, an algorithm is devised in MATLAB (The detailed code can be referred to from **APPENDIX D**) which finds the statistical relationships between these points.

I. The result file ***CoordinateOutput*** is imported into MATLAB and the contents are read.

II. The first column is separated as 'text data' as it contains information about the proteins and the next 3 columns are separated as a matrix 'data' which is numeric in character as it has the positions of the X, Y and Z orthogonal coordinates measured in Armstrongs.

III. Both the *textdata* and *data* are passed into a function ***Eu_distance*** which would calculate the Euclidean distances and return a histogram plot of the distance versus the frequency of distances found.

a) The unique elements of the *textdata* attribute are collected into a new set called as *values*. The *values* variable now contains the unique protein ENSP numbers from the first column of the result set elements.

b) For each element in the values set, if the occurrence of the element is found more than once in a file, it means that there are at least two points in the ***CoordinateOutput*** file belonging to the same protein.

c) For such values we calculate the Euclidean distances by extracting the respective indexes of the protein elements in the *textdata* matrix , to the same indexes in the *data* matrix for the coordinate values.

d) The distances are calculated using the ***pdist(X)*** function in MATLAB. This function computes the pair wise Euclidean distances between objects in the data matrix, X. This data matrix is updated each time for a new ENSP instance.
Thus, distances are only calculated for the mutation coordinates which belong to the same protein.

distance = pdist(compare values, 'euclidean');

IV. A histogram is plotted for distances versus the frequency of the distances observed while computing distance calculations between points belonging to the same proteins.

a) The **histfit()** function provided by MATLAB is used for constructing the histogram. The number of bins is taken to be 100 to ensure the output is well readable and the distribution can be viewed efficiently.

histfit(data, nbins) plots a histogram using nbins bins (here, 100) and fits a normal density function.

distancesNew = histfit(distance,100);

b) The calculated distances are plotted on the X axis and the frequency or the number of times a distance is calculated (within a bin range) is plotted on the Y axis of the histogram.

7.2 Summary

The information extracted through the data integration phase gave a resultant file with all the coordinates of these mutations on the 3D space. This chapter dealt with formulating the information by constructing visual graphs - histograms which help the researchers analyze the nature of information retrieved and what it collectively represents. This area can be further researched and more statistical analysis can be drawn out of the given data by using other measures of calculations and visualizations. The next chapter - Results discusses the details about the histogram plotted and what we infer from the study.

8 Results and discussion

For this research experiment, data from the 1000 Genomes Project was taken which includes the genome sequences of more than 2000 individuals. The single nucleotide polymorphisms (mainly missense variants) between amino acid residues of the reference human genome and the genomes reported, were studied for all the individuals on the exon level of the genomes. The information about these mutations, their positions on the genome, the proteins they belong to, was generated through the VEP tool provided by ENSEMBL. The data was studied and queried at different stages through the SUPERFAMILY database to find the related mappings of the mutations on positions on the respective protein domains. It is then that the position from the sequence of the domains was mapped to positions on the Hidden Markov Models of the models in the SUPERFAMILY database. These model positions were used to extract the orthogonal X,Y and Z coordinates of the mutations mapped onto the 3D structures of the proteins and their domains.

Statistical testing is important to be carried out as it evaluates if the results are statistically significant in nature or not. in the following paragraphs we present the results of the analysis on the structure level of proteins and also state the findings which would answer our hypothesis.

8.1 Histogram Plot

There are different methods to test statistical results, out of which, this study chooses to build a histogram of the Euclidean distance calculations of mutant residues occurring in the same protein structure.

A total of 16,498 data points (each point representing a residue position on the 3 dimensional space) have been analyzed and processed to calculate Euclidean distances between the,. For each protein, if a protein has more than one point mutations, we calculate the distance between those points. Example, for a protein A, if there are 4 coordinate sets found (4 data points), there are $^{4}\mathbf{C}_{2}$ combinations of distances calculated between each pair of atoms.

The coordinates for each protein were studied and distances between them on the 3D plane were calculated with the help of MATLAB to present a histogram of the distances versus the frequency of the distances, for a visual analysis via graphical representations. The histfit function is used which gives the histogram of the data with superimposed normal density.

The following was the resultant histogram obtained:

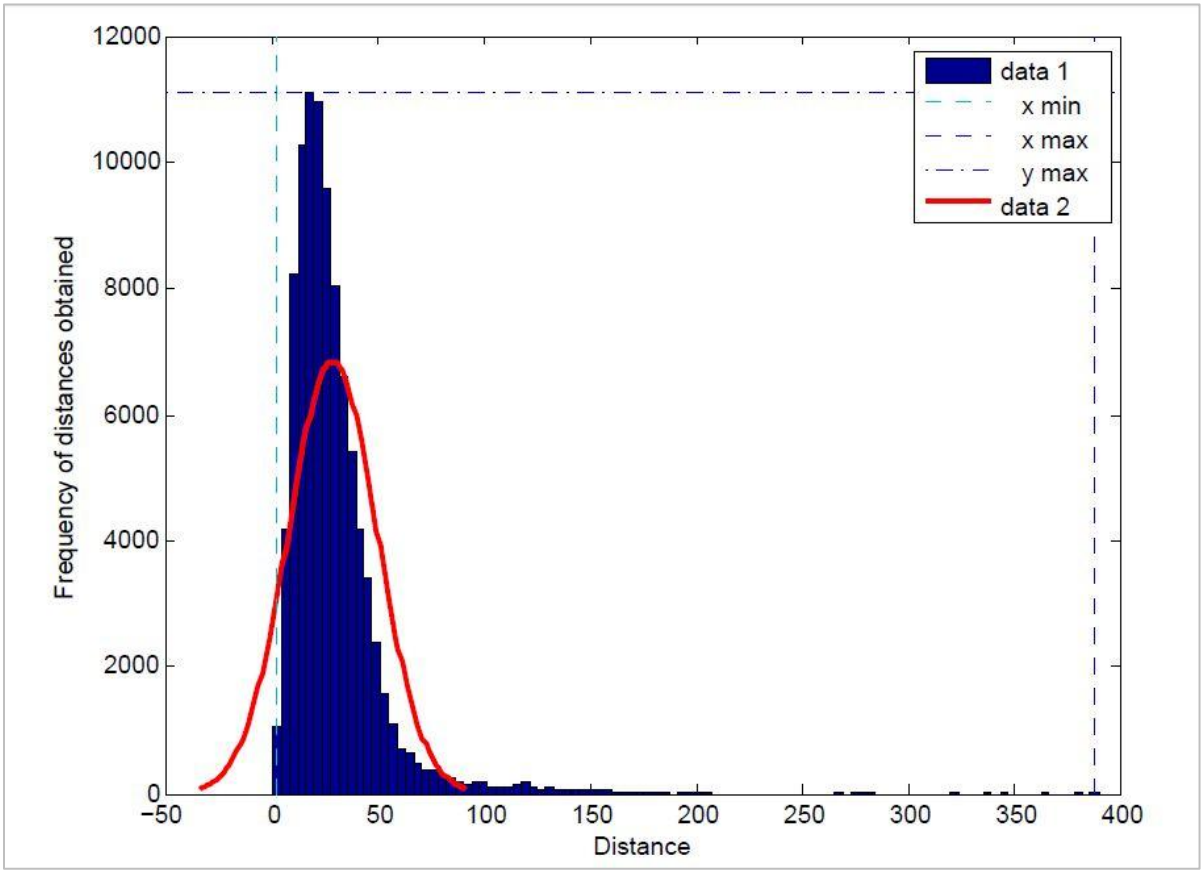


Figure 30: *The final histogram with the data distribution. (data 1 - distance calculations) in blue and the normal distribution (data 2) of the data in red.*

8.1.1 Detailed description

Statistical Value	X axis	Y axis
minimum	2.931	0
maximum	388.3	1.11e + 04
mean	195.6	931
median	195.6	4
mode	2.931	0
std deviation	112.9	2495
range	385.3	1.11e + 04

Table 6 : *Statistical Results of observed distribution*

I. The mode of the above data is 2.931 depicting that most of the mutations fall between a distance range of approximately 2.9 Armstrongs. Also, the minimum value and the mode value for the distances are the same i.e. 2.931 Armstrong.

II. Values within the distance of approximately 2 to 25 Armstrongs are found to be significantly higher in number as compared to the ones calculated between residues present at a farther

distance.

III. The histogram predicts that most of the complementary mutations occur on the 3 dimensional structure when the residues are present very close to each other.

IV. Distances between 20 -30 Armstrong are found to be the most frequent (calculated to occur 8000-11000 times) between the mutant residues.

V. The resultant distance calculations tend to follow an exponential distribution with their frequencies. The nature of the graph tends to exhibit a normal continuous distribution.

8.1.1.1 Observations

The resultant graph acquired from the study has a normal continuous distribution implying that the values are correlated and the histogram shows significant sense of mutations occurring very close to each other a majority of the times.

The above results suggest that it is more likely that in a selected protein structure, the mutations occur very close to each other. This can prove to be a significant statistical measure to test our hypothesis which argues on the existence of complementary mutations.

8.1.2 Validation of results obtained

The histogram above represents the actual results of distance calculations between points of mutation in the sequence. There points were derived by pointing out specific positions on the model sequence which has been calculated as ***ModStart*** in earlier chapters. this ***ModStart*** gives us the exact position of the residue in the model which is the mutant residue and thus the coordinates of that very residue from the respective ASTRAL files.

To proceed with the validation of the results, we produce more histograms wherein the position of the mutation has not been specified prior to extracting the coordinates. This means that we now extract any random residue from a sequence and then its respective coordinates from the ASTRAL files to observe the random data distribution.

We generate a background random distribution which gives a basis to present if the results we have achieved from the study are random, or significantly different and more valuable than random results.

- A random position code was generated which selects any arbitrary residue at any position in the sequence and then gives the result set of these coordinates.
- There were two random plots generated to give a weighty basis of comparison between random distributions and significant results.

8.1.2.1 Random Run I :

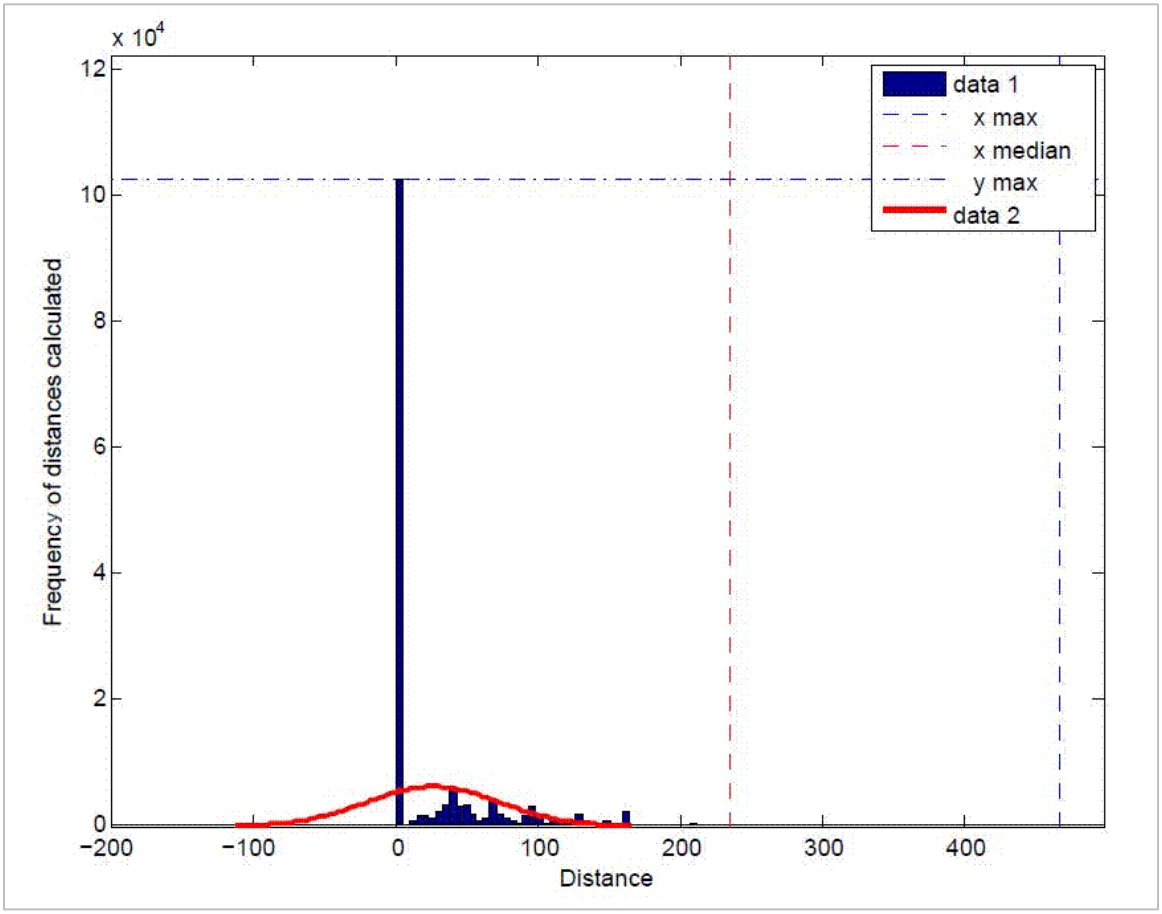


Figure 31: *Histograms of distance calculation of random position coordinates I calculated for 23380 residues.*

The coordinate data of 23380 atoms with their protein information is generated, wherein the positions on the sequences were not specified.120,000 combinations of distances lie at zero, which means the distance between the points is zero if they are the same point. Thus we ignore the zero distances because they would refer to comparing distance between the same residue.
The histogram shows random output frequencies for the relevant data (the distances between two different points).

Statistical Value	X axis	Y axis
minimum	2.351	0
maximum	467.9	1.024e + 05
mean	235.1	1566
median	235.1	8
mode	2.351	0
std deviation	136.4	1.024e + 05
range	465.5	1.024e + 05

Table 7: *Statistical Results of random distribution I*

8.1.2.2 Random Run II

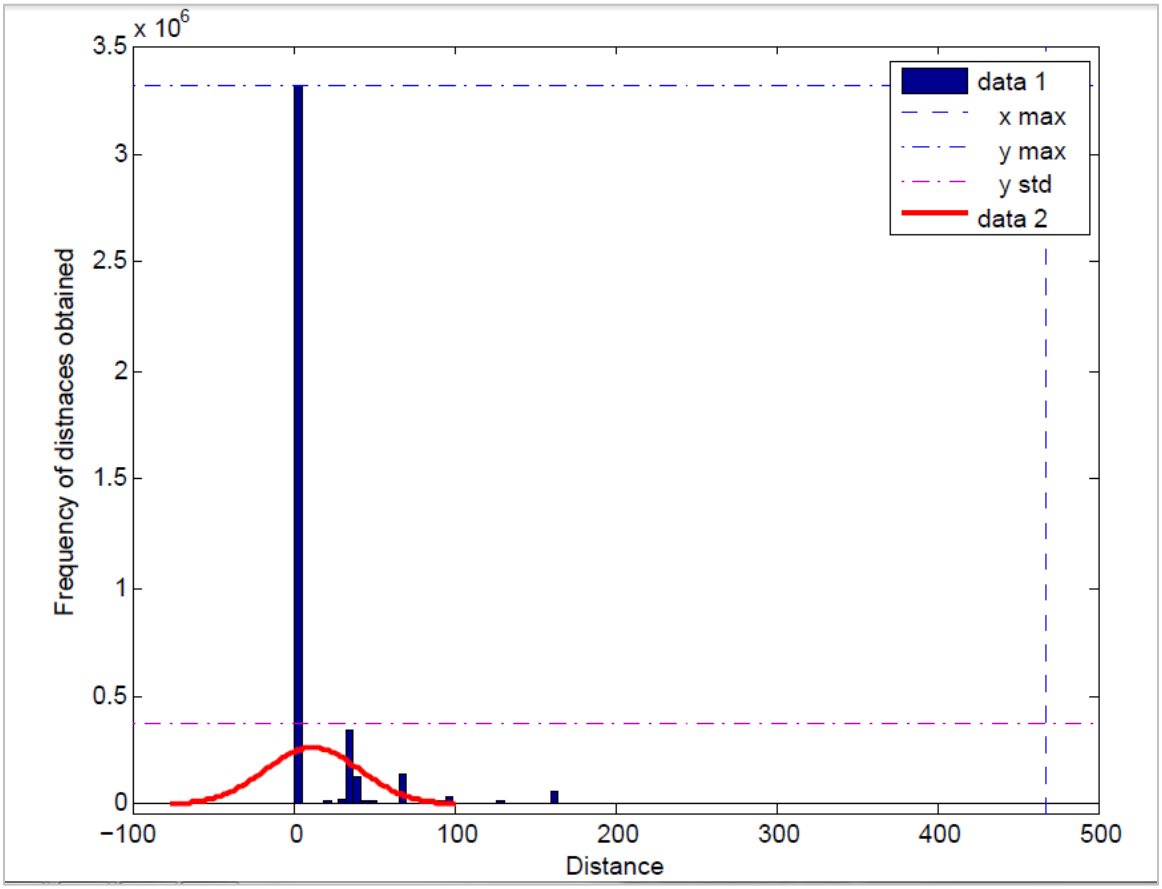


Figure 32 :Histograms of distance calculation of random position coordinates I I

In Random Run II, the data size has been increased to produce more coordinate set of atoms to efficiently analyze the effect of random positions and study their distribution.

In this run, 38,609 random coordinates were plotted on the histogram, to study the distance versus frequency. The graph is highly similar to the random run I. We would neglect the zero based positions as these are the distances being calculated from an atom the very same atom.

Statistical Value	X axis	Y axis
minimum	2.351	0
maximum	467.9	3.32e + 06
mean	235.1	4.183e + 04
median	235.1	8
mode	2.7791	0
std deviation	136.4	3.334e + 05
range	465.5	3.32e + 06

Table 8: *Statistical Results of random distribution II*

8.2 Inference

The two Random Runs I and II, are highly similar in distribution. In the first run the data size has been taken as nominal, whereas in the second run we increase the data size to observe to what extent the random distribution histogram can vary.

The random graphs have been compared to the resultant histogram obtained from the research. It is clearly noted that :

- ✓ The result histogram is evenly distributed, giving a significant amount of information on the data.
- ✓ The result histogram is also correlate, unlike the random outputs produced by the two random histograms.
- ✓ The research results are significantly different from random.
- ✓

The random distributions show random noisy data at different values of the distance frequencies. It is also observed that even if the data size has been increased, the mean value remains the same. This implies that for any random elements, the distribution would almost be the same. This un-correlation represents that the random distributions are very different from the observed distribution.

Thus we infer that the observed histogram has significant results. Also, that most of the mutations lie between the range of 2 to 50 Å. A maximum of pair of mutated atoms lie at a distance between approximately 10 to 30 Å. On the basis of these information, we can finally test the hypothesis.

8.3 Hypothesis Testing

In this research, statistical testing is essential to argue about the correlation of mutations and how they tend to follow similar or different trends when compared on the structural level.

The calculations have been carefully proceeded with, while considering the elements of each protein at a time, so as to find the correlated distances between its own elements of point mutations.

The histogram plot representation depicts a normal distribution of distances wherein, it can be predicted that mutations occurring close to each other are much more probable and more likely correlated than those occurring at distant points on the structure of a folded protein.

But, statistical inference using a normal distribution is not robust to the presence of outliers. It is not necessary that all the pairwise mutations found to exist close to each other must be complementary in nature. Instead, in this research, we conclude that it is "more likely" that complementary mutations exist and provide statistical significance on the basis of pair wise distance calculations between the mutations on each protein.

We present the hypothesis to be true to certain extent, at which this study has been able to draw a conclusion through data integration and usage of multiple database and information resource.

9 Conclusion

The research has successfully presented with large scale analysis of data related to human genetic variation on the chromosome to DNA, to protein sequence, to domain, to protein structure levels. A knowledge based approach has been presented wherein the characteristics of the biological data and the immense possibilities of combinations of processes and their amalgamation, within a cell and so a genome, have all been taken into account through each phase in the progression of the research.

This study on human genetic variation has integrated data from various bioinformatics data resources. The findings can be studied much further to extract knowledge relating to evolution. It is concluded that the information and results obtained are precise and accurate at each phase of the study. Validation testing is carried out to confirm that the results retrieved are highly significant.

Complementary mutations are important because this is a fundamental question of '**how**' evolution takes place at the molecular level. Scientists have raised the question about the existence of complementary mutations in yester years, but there is no consensus in the scientific community about it. This research shall prove to be a pioneering discovery in respect of the same and also in an attempt to study together the genome and the proteins. Furthermore, this high - end bioinformatics research shall help scientists re-think about the existence of complementary mutations and how this research could be taken forward. Not to mention that understanding evolution at the molecular level is a challenge to the human race of equal intellectual importance to understanding the origins of the universe, neuroscience etc. This research answers on a new level, what is life?

9.1 Future work

The study can be taken further to different levels of expertise. The genome level discoveries can be studied by observing these mutations on an individual level. This means, there can be population diversity discoveries among inter and intra populations, to infer the level at which mutations are similar or different among individuals. Furthermore, integrating common and rare genetic variations in diverse human populations can be an area of research.

Since 3D structural analysis and associating data on these structures is not yet an area of much research due to its novelty, the 3D structural analysis can be taken to a more higher and complex level. Also, the existence of complementary mutations can be inspected in more detail even at the domain level.

Disease detection and examining the rates of mutations, is another area where the results of this research can be applied to. Most of the diseases are mainly caused by the mutations occurring at unexpected or unidentified regions. With this work, the mutations can be mapped to their origin, to study how diseases actually progress. Overall this research paves a modern approach to view evolution, and study human variation and the intricacies of biological entities.

Bibliography

[1] Lesk, A.M., Chothia, C. **How different amino acid sequences determine similar protein structures: The structure and evolutionary dynamics of the globins.** J. Mol. Biol. **136**:225-270, 1980.

[2] B. D. Singh (2007). **Fundamentals of Genetics.** Ludhiana: Kalyani Publishers

[3] The International HapMap Consortium. (December 2003). **The International HapMap Project**. Nature **426** , 789-796.

[4] Robert Krulwich (2001-04-17). **Cracking the Code of Life** (Television Show). PBS. ISBN 1-5375-16-9.

[5] Gobel et al., (1994), **Correlated mutations and residue contacts in proteins**, Proteins: Structure, Function, and Bioinformatics, **18** , pp. 309–317. (doi: 10.1002/prot.340180402)

[6] The 1000 Genomes Project Consortium. (2010). **A map of human genome variation from population-scale sequencing**. Nature 467, 1061-1073. (doi:10.1038/nature09534)

[7] Flicek P., Amode M. R., Barrell D., Beal K., Brent S., Chen Y., Clapham P., Coates G., Fairley S., Fitzgerald S. et al. (2011). **Ensembl 2011**. Nucleic Acids Res. **39** Suppl 1, D800–D806. doi:10.1093/nar/gkq1064

[8] McLaren W, Pritchard B, Rios D, Chen Y, Flicek P, Cunningham F. **Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor.** *BMC Bioinformatics* 26(16): 2069-70(2010 doi:10.1093/bioinformatics/btq330

[9] Gough, J., Karplus, K., Hughey, R. and Chothia, C. (2001). "**Assignment of Homology to Genome Sequences using a Library of Hidden Markov Models that Represent all Proteins of Known Structure**." J. Mol. Biol., 313(4), 903-919.

[10] Derek Wilson, Ralph Pethica, Yiduo Zhou, Charles Talbot, Christine Vogel, Martin Madera, Cyrus Chothia and Julian Gough. (2009). **SUPERFAMILY—sophisticated comparative genomics, data mining, visualization and phylogeny**. Nucleic Acids Research, Database issue. Vol. **37**, D380–D386. (doi:10.1093/nar/gkn762)

[11] Andreeva,A., Howorth,D., Chandonia,J.-M., Brenner,S.E., Hubbard,T.J.P., Chothia,C. and Murzin,A.G. (2008) **Data growth and its impact on the SCOP database: new developments**. Nucleic Acid Res., **36**, D419–D425.

[12] Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) **Basic local alignment search tool**. J. Mol. Biol., **215**, 403–410.

[13] Chandonia,J.M., Hon,G., Walker,N.S., Lo Conte,L., Koehl,P., Levitt,M. and Brenner,S.E. (2004) **The ASTRAL compendium in 2004**. Nucleic Acids Res., **32**, D189–D192.

[14] Karplus,K., Barrett,C. and Hughey,R. (1998) **Hidden Markov models for detecting remote protein homologies**. Bioinformatics, 14, 846–856.

[15] Murzin A. G., Brenner S. E., Hubbard T., Chothia C. (1995). **SCOP: a structural classification of proteins database for the investigation of sequences and structures.** J. Mol. Biol. 247, 536-540.

[16] Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) **The Protein Data Bank**. Nucleic Acids Res., **28**, 235–242.

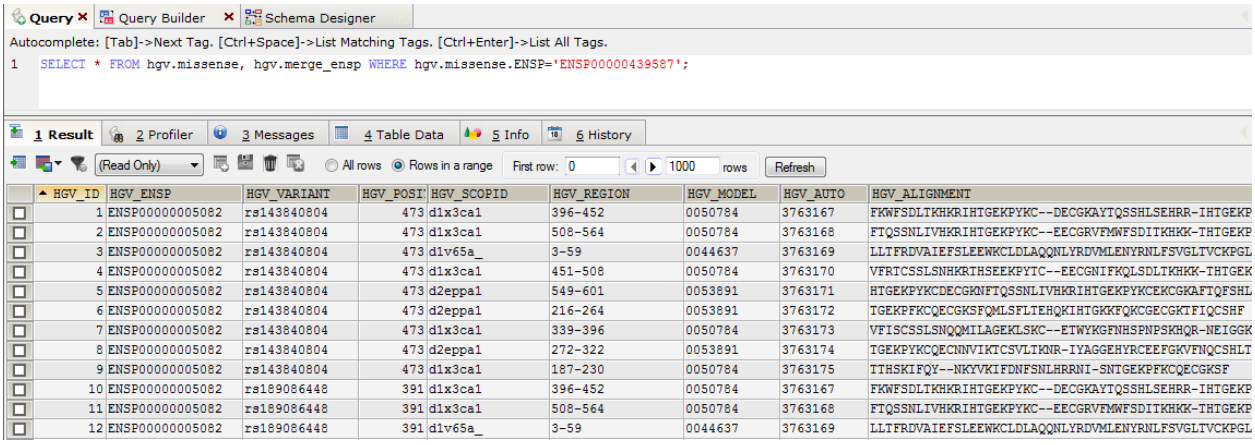
[17] **MATLAB** [Internet] c.2011. MathWorks: [cited 2010 August 29]. Available from: http://www.mathworks.co.uk/products/matlab/

[18] Kitts A, Sherry S. The Single Nucleotide Polymorphism Database (dbSNP) of Nucleotide Sequence Variation. 2002 Oct 9 [Updated 2011 Feb 2]. In: McEntyre J, Ostell J, editors. The NCBI Handbook [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2002-. Chapter 5. Available from: http://www.ncbi.nlm.nih.gov/books/NBK21088/

[19] **SQLyog** [Internet]. c2010. Webyog: [cited 2010 August 2]. Available from: http://www.webyog.com/en/

APPENDIX A TABLES

MISSENSE FILE INITIAL



Query | Query Builder | Schema Designer

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->List All Tags.

```
1  SELECT * FROM hgv.missense, hgv.merge_ensp WHERE hgv.missense.ENSP='ENSP00000439587';
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info | 6 History

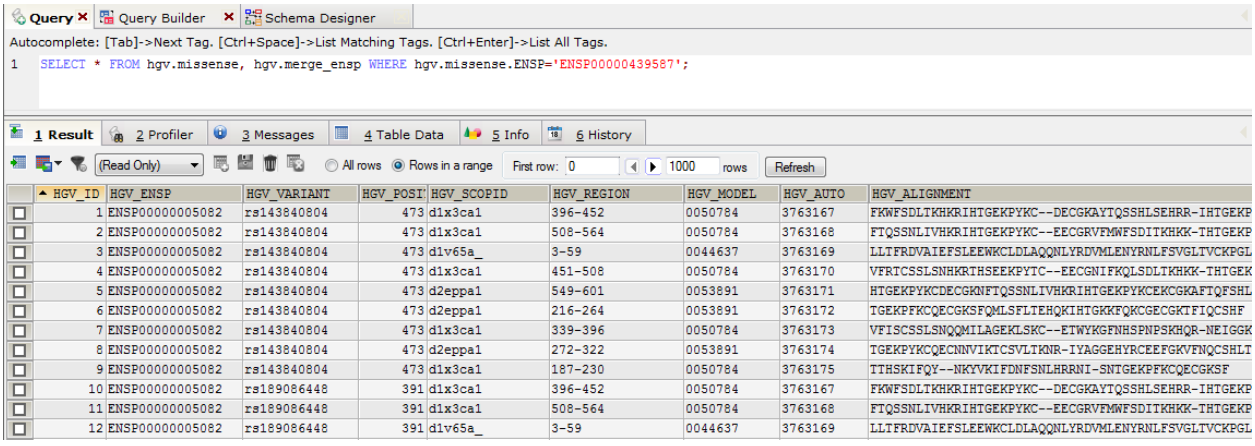
(Read Only) | All rows | Rows in a range | First row: 0 | 1000 rows | Refresh

HGV_ID	HGV_ENSP	HGV_VARIANT	HGV_POSI	HGV_SCOPID	HGV_REGION	HGV_MODEL	HGV_AUTO	HGV_ALIGNMENT
1	ENSP00000005082	rs143840804	473	d1x3ca1	396-452	0050784	3763167	FKWFSDLTKHKRIHTGEKPYKC--DECGKAYTQSSHLSEHRR-IHTGEKP
2	ENSP00000005082	rs143840804	473	d1x3ca1	508-564	0050784	3763168	FTQSSNLIVHKRIHTGEKPYKC--EECGRVFMWFSDITKHKK-THTGEKP
3	ENSP00000005082	rs143840804	473	d1v65a_	3-59	0044637	3763169	LLTFRDVAIEFSLEEWKCLDLAQQNLYRDVMLENYRNLFSVGLTVCKPGL
4	ENSP00000005082	rs143840804	473	d1x3ca1	451-508	0050784	3763170	VFRTCSSLSNHKRTHSEEKPYTC--EECGNIFKQLSDLTKHKK-THTGEK
5	ENSP00000005082	rs143840804	473	d2eppa1	549-601	0053891	3763171	HTGEKPYKCDECGKNFTQSSNLIVHKRIHTGEKPYKCEKCGKAFTQFSHL
6	ENSP00000005082	rs143840804	473	d2eppa1	216-264	0053891	3763172	TGEKPFKCQECGKSFQMLSFLTEHQKIHTGKKFQKCGECGKTFIQCSHF
7	ENSP00000005082	rs143840804	473	d1x3ca1	339-396	0050784	3763173	VFISCSSLSNQQMILAGEKLSKC--ETWYKGFNHSPNPSKHQR-NEIGGK
8	ENSP00000005082	rs143840804	473	d2eppa1	272-322	0053891	3763174	TGEKPYKCQECNNVIKTCSVLTKNR-IYAGGEHYRCEEFGKVFNQCSHLT
9	ENSP00000005082	rs143840804	473	d1x3ca1	187-230	0050784	3763175	TTHSKIFQY--NKYVKIFDNFSNLHRRNI-SNTGEKPFKCQECGKSF
10	ENSP00000005082	rs189086448	391	d1x3ca1	396-452	0050784	3763167	FKWFSDLTKHKRIHTGEKPYKC--DECGKAYTQSSHLSEHRR-IHTGEKP
11	ENSP00000005082	rs189086448	391	d1x3ca1	508-564	0050784	3763168	FTQSSNLIVHKRIHTGEKPYKC--EECGRVFMWFSDITKHKK-THTGEKP
12	ENSP00000005082	rs189086448	391	d1v65a_	3-59	0044637	3763169	LLTFRDVAIEFSLEEWKCLDLAQQNLYRDVMLENYRNLFSVGLTVCKPGL

RESULT AFTER SUPREFAMILY DETAILS RETRIEVED

scopid	region	model	auto	alignment		modstart
d2d8qa1	463-506	0052402	36856582	TKKKQWCYNCEEEAMYHC--CWNTSYCSIKCQQEHWHAeHKRTCRR	46B	5
d1eqfa1	71-169	0046608	36856581	IVSRMKER--AIDLNKKGKDNKHPMYRRLVHSAVDVPTIQEKVNEGKYRSYEEFKAD...	101B	34
d1khca_	176-298	0047802	36856580	DNWFCYPCIPNHELVWAKMKGFGFWPAKVMQ--------KEDNQVDVRFFGHhHQRA...	134B	1
d1tuba1	1-171	0044120	36838789	MDSVRSGPFGQVFRPDNFIFGQCGAGNNWAKGHYTEGAELMESVMDVVRKEAESCDC...	171B	75
d1tuba2	172-357	0044121	36838788	GQLNADLRKLAVNMVPFPRLHFFMPGFAPLTSRGSQQYRALTVAELTQQMFDAKNMM...	194B	1
d2cqea2	3-27	0052242	3800021	TKQITCRYFMHGVCREGSQCLFSHD	25B	4
d2cqea2	32-56	0052242	3800020	KPSTICKYYQKGYCAYGTRCRYDHT	25B	4
d1v87a_	228-298	0050087	3800019	FAFQASQDKVCSICMEVILEKAS----------ASERRFGILSnCNHTYCLSCIRQW...	81B	18
d2d8qa1	517-560	0052402	3781855	TKKKQWCYNCEEEAMYHC--CWNTSYCSIKCQQEHWHAeHKRTCRR	46B	5
d1weva_	46-111	0050474	3781854	WLPGDEIDWETENHDWYCFECH-----LPGEVLICDLCFRVYHSKCLSDEFRL---R...	74B	2
d1eqfa1	94-223	0046608	3781853	RDSSSPWQCPVCRSIKKKNTNkqEMGTYLRFIVSRMKER--AIDLNKKGKDNKHPMY...	132B	5
d1khca_	230-352	0047802	3781852	DNWFCYPCIPNHELVWAKMKGFGFWPAKVMQ--------KEDNQVDVRFFGHhHQRA...	134B	1
d1weva_	106-166	0050474	3781847	NKDWETENHDWYCFECH-----LPGEVLICDLCFRVYHSKCLSDEFRL---RDSSSP...	69B	7

PrOTEIN AND DOMAIN INFORMATION WITH VARIANTS



Query | Query Builder | Schema Designer

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->List All Tags.

```
1  SELECT * FROM hgv.missense, hgv.merge_ensp WHERE hgv.missense.ENSP='ENSP00000439587';
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info | 6 History

(Read Only) | All rows | Rows in a range | First row: 0 | 1000 rows | Refresh

HGV_ID	HGV_ENSP	HGV_VARIANT	HGV_POSI	HGV_SCOPID	HGV_REGION	HGV_MODEL	HGV_AUTO	HGV_ALIGNMENT
1	ENSP00000005082	rs143840804	473	d1x3ca1	396-452	0050784	3763167	FKWFSDLTKHKRIHTGEKPYKC--DECGKAYTQSSHLSEHRR-IHTGEKP
2	ENSP00000005082	rs143840804	473	d1x3ca1	508-564	0050784	3763168	FTQSSNLIVHKRIHTGEKPYKC--EECGRVFMWFSDITKHKK-THTGEKP
3	ENSP00000005082	rs143840804	473	d1v65a_	3-59	0044637	3763169	LLTFRDVAIEFSLEEWKCLDLAQQNLYRDVMLENYRNLFSVGLTVCKPGL
4	ENSP00000005082	rs143840804	473	d1x3ca1	451-508	0050784	3763170	VFRTCSSLSNHKRTHSEEKPYTC--EECGNIFKQLSDLTKHKK-THTGEK
5	ENSP00000005082	rs143840804	473	d2eppa1	549-601	0053891	3763171	HTGEKPYKCDECGKNFTQSSNLIVHKRIHTGEKPYKCEKCGKAFTQFSHL
6	ENSP00000005082	rs143840804	473	d2eppa1	216-264	0053891	3763172	TGEKPFKCQECGKSFQMLSFLTEHQKIHTGKKFQKCGECGKTFIQCSHF
7	ENSP00000005082	rs143840804	473	d1x3ca1	339-396	0050784	3763173	VFISCSSLSNQQMILAGEKLSKC--ETWYKGFNHSPNPSKHQR-NEIGGK
8	ENSP00000005082	rs143840804	473	d2eppa1	272-322	0053891	3763174	TGEKPYKCQECNNVIKTCSVLTKNR-IYAGGEHYRCEEFGKVFNQCSHLT
9	ENSP00000005082	rs143840804	473	d1x3ca1	187-230	0050784	3763175	TTHSKIFQY--NKYVKIFDNFSNLHRRNI-SNTGEKPFKCQECGKSF
10	ENSP00000005082	rs189086448	391	d1x3ca1	396-452	0050784	3763167	FKWFSDLTKHKRIHTGEKPYKC--DECGKAYTQSSHLSEHRR-IHTGEKP
11	ENSP00000005082	rs189086448	391	d1x3ca1	508-564	0050784	3763168	FTQSSNLIVHKRIHTGEKPYKC--EECGRVFMWFSDITKHKK-THTGEKP
12	ENSP00000005082	rs189086448	391	d1v65a_	3-59	0044637	3763169	LLTFRDVAIEFSLEEWKCLDLAQQNLYRDVMLENYRNLFSVGLTVCKPGL

APPENDIX B : MYSQL QUERIES

1. Loging into mysql
   execute the following on terminal : mysql -u root -p

2. **hgv Database creation**
   create database hgv;

3. **Missense table creation**
   create table missense (
   ID Integer(10) Primary Key,
   Variant VARCHAR(100),
   Position INTEGER(10),
   ENSP VARCHAR(15));

4. **Loading the acquired variant data into missense data**
   LOAD DATA INFILE 'C:/Users/Pranshu/Desktop/DB/missense.csv' INTO TABLE missense COLUMNS TERMINATED BY ',' LINES TERMINATED BY '\n' ;

```
create table hgv.supfam_ensp_rs (
opdata_id Integer(5) primary key,
seqid Varchar(15),
scopid Varchar(10),
region Varchar(100),
model Varchar(15),
auto Varchar(15),
alignment Varchar(10000),
modstart Integer(5)  );

mysql> LOAD DATA INFILE 'C:/Users/Pranshu/Desktop/op_data_ID.txt' INTO TABLE sup
fam_ensp_rs  COLUMNS TERMINATED BY '\t' LINES TERMINATED BY '\n' ;
```

```
select
   cla.scopid, ass.region, ass.model, align.*
from
   missense,
   protein,
   ass,
   align,
   model,
   cla
where
   missense.ENSP_Variant = protein.seqid
      and ass.model = model.model
      and cla.px = model.seed
      and protein.protein = ass.protein
      and ass.auto = align.auto
      and genome = 'hs'.      and ass.evalue <= 0.0001;
```

APPENDIX C : JDBC connections and JAVA codes

C.1. Establishing connection with Superfamily database server to extract related information about the queried proteins and the positions of mutations on them.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class jdbcCode {

	private static String loadColumn(String filename) throws IOException{
		File file = new File(filename);
		String text = "";
		if(file.exists()){
			int len = (int)file.length();
			byte[] data = new byte[len];
			FileInputStream fis = new FileInputStream(file);
			fis.read(data);
			text = new String(data);
			text = text.trim().replace('\n', ',').replace("", '\");
		}
		//System.out.println(text);
		return text;
	}
	public static void main(String args[]) throws IOException {
		String dbtime;
		String dbUrl = "jdbc:mysql://supfam2.cs.bris.ac.uk/superfamily";
		String dbClass = "com.mysql.jdbc.Driver";
		String query = "SELECT  seqid, cla.scopid, ass.region, ass.model, align.* FROM protein, ass, align,  model,
cla WHERE ass.model = model.model    AND cla.px = model.seed    AND protein.protein = ass.protein    AND ass.auto =
align.auto    AND genome = 'hs'    AND seqid in ("+loadColumn("ensp.csv")+")  AND ass.evalue <= 0.0001";

		try {

			Class.forName("com.mysql.jdbc.Driver");
			Connection con = DriverManager.getConnection(dbUrl, "pranshu",
					"pranshu");
			Statement stmt = con.createStatement();
			ResultSet rs = stmt.executeQuery(query);
			StringBuilder sb = new StringBuilder();
			// Get result set meta data
			ResultSetMetaData rsmd = rs.getMetaData();
			int numColumns = rsmd.getColumnCount();
			List<String> cNamaes = new ArrayList<String>();
			// Get the column names; column indices start from 1
			for (int i = 1; i < numColumns + 1; i++)
				cNamaes.add(rsmd.getColumnName(i));
			sb.append(cNamaes.toString().replaceAll("\\[|\\]", ""));
```

```
                        sb.append('\n');
                        while (rs.next()){
                                for(String name: cNamaes)
                                        sb.append(rs.getString(name)).append('\t');
                                sb.deleteCharAt(sb.length()-1).append('\n');
                        }
                        // end while
                        con.close();
                        File file = new File("op.txt");
                        file.createNewFile();
                        FileOutputStream fw = new FileOutputStream(file);
                        fw.write(sb.toString().getBytes());
                        fw.close();
                } // end try

                catch (ClassNotFoundException e) {
                        e.printStackTrace();
                }

                catch (SQLException e) {
                        e.printStackTrace();
                }

        } // end main

} // end class
```

C.2. The program returns the coordinates of mutated residues from the corresponding ASTRAL files

- The information is gathered, the positions are checked to be within valid ranges,
- The valid entries are processed to find the position of residue on the model.
- The ASTRAL files of respective domains are consulted and coordinates of positions extracted.
- The results are formulated and written into an output file which has the information of coordinates of each residue in the 3D space along with its protein information.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

	public class residue_position {

	private static final int INDEX_ID = 1;
	private static final int INDEX_POSITION = 2;
	private static final int INDEX_REGION = 3;
	private static final int INDEX_SCOP_ID = 4;
	private static final int INDEX_MOD_START = 5;
	private static final int INDEX_AMINO_ACID = 6;
	private static final int INDEX_ALIGNMENT = 7;
	private static final int INDEX_ENSP = 8;

	private static String alignChar;
	public static String aminoAcid;

	public static void main(String args[]) throws IOException {
		File op = new File("Output.txt");
		PrintWriter pr = new PrintWriter(op);

		String dbUrl = "jdbc:mysql://localhost:3306/hgv";
		String query = "SELECT hgv.merge_ensp.HGV_ID, hgv.merge_ensp.HGV_POSITION,
hgv.merge_ensp.HGV_REGION,  hgv.merge_ensp.HGV_SCOPID,  "
				+ "hgv.merge_ensp.HGV_MODSTART, hgv.missense_fulldata.Amino_acids,
hgv.merge_ensp.HGV_ALIGNMENT, hgv.merge_ensp.HGV_ENSP "
				+ "FROM hgv.merge_ensp, hgv.missense_fulldata "
				+ "where hgv.merge_ensp.HGV_ID=hgv.missense_fulldata.ID  ";

		try {

			Class.forName("com.mysql.jdbc.Driver");
			Connection con = DriverManager.getConnection(dbUrl, "root", "root");
			Statement stmt = con.createStatement();
			ResultSet rs = stmt.executeQuery(query);

			while (rs.next()) {

				int id = (Integer) rs.getObject(INDEX_ID);
```

```
                                        int position = (Integer) rs.getObject(INDEX_POSITION);
                                        String region = (String) rs.getObject(INDEX_REGION);
                                        String scopId = (String) rs.getObject(INDEX_SCOP_ID);
                                        int modStart = (Integer) rs.getObject(INDEX_MOD_START);
                                        String aminoAcid = (String) rs.getObject(INDEX_AMINO_ACID);
                                        String alignment = (String) rs.getObject(INDEX_ALIGNMENT);
                                        String ENSP = (String) rs.getObject(INDEX_ENSP);

                                        int calculatedModStart;
                                        int leftValue = checkPositionWithinRange(rs);
                                        if (leftValue > 0) {
                                                calculatedModStart = findModStart(alignment, leftValue,
                                                                modStart, position, scopId, aminoAcid);

                                                System.out.println("ModStart: " + calculatedModStart);

File fd = new File("C:/Users/Pranshu/Desktop/PDB files or ASTRAL!!/ASTRAL/pranshu  ASTRAL/tmp/pranshu/ scopId +
".ent");

                                                BufferedReader br = new BufferedReader(new FileReader(fd));
                                                String str = null;
                                                String result;

                                                while ((str = br.readLine()) != null) {

                                                        String[] tuple = str.split("[ |\t]+");

                                                        if (tuple[0].equals("ATOM")) {
                                                                if (!isInteger(tuple[5])) {
                                        System.out.println("CORRUPTED   " + scopId + "\t" + calculatedModStart+ "CORRUPTED");
                                                                        continue;
                                                                }
                                                                if (Integer.parseInt(tuple[5]) == calculatedModStart) {

                                                                        if (tuple[2].equals("CA")) {

                                                        result = ENSP + ',' + tuple[6] + "," + tuple[7] + "," + tuple[8];

                                                                                pr.write(result + "\n");
                                                                                pr.flush();
                                                                                break;
                                                                        }

                                                                }
                                                        }
                                                } // end inner while
                                        }
                                }
                                con.close();
                        } // end try

                        catch (ClassNotFoundException e) {
                                e.printStackTrace();
                        }

                        catch (SQLException e) {
                                e.printStackTrace();
                        }

                } // end main
```

```
public static boolean isInteger(String input) {
		try {
			Integer.parseInt(input);
			return true;
		} catch (Exception e) {
			return false;
		}
	}
```

C.2.1. To check if residue is found within the domain range

```
public static int checkPositionWithinRange(ResultSet rs)
			throws SQLException {

		int id = (Integer) rs.getObject(INDEX_ID);
		int position = (Integer) rs.getObject(INDEX_POSITION);
		String region = (String) rs.getObject(INDEX_REGION);
		String regionArray[] = region.split(",");

		String leftString = regionArray[0].trim();
		String rightString = regionArray[regionArray.length - 1].trim();

		String leftStringArray[] = leftString.split("-");
		String rightStringArray[] = rightString.split("-");

		int leftInt = Integer.parseInt(leftStringArray[0]);
		int rightInt = Integer.parseInt(rightStringArray[1]);

		if (position >= leftInt && position <= rightInt) {

			return leftInt;
		}

		return -1;
	}
```

C.2.2. Finding ModStart position in Model

```
public static int findModStart(String alignment, int regionStart,
			int modStart, int position, String scopId, String aminoAcid) {

		regionStart -= 1;
		modStart -= 1;

		int positionCount = 1;

		for (int i = 0; i < alignment.length(); i++) {

			if (Character.isUpperCase(alignment.charAt(i))
					|| Character.isLowerCase(alignment.charAt(i))) {
				regionStart = regionStart + 1;
```

```
                                        if (regionStart == position) {

                                                if (!Character.isUpperCase(alignment.charAt(i))) {
                                                        int leftCap = 1;
                                                        int rightCap = 1;

                                                        for (int x = i + 1; x < i + 6; x++) {

                                                                if (Character.isUpperCase(alignment.charAt(x))) {

                                                                        break;
                                                                }
                                                                rightCap += 1;
                                                        }

                                                        for (int x = i - 1; x < i - 6; x--) {

                                                                if (Character.isUpperCase(alignment.charAt(x))) {

                                        System.out              .println("Character shifted to left by "+ leftCap);
                                                                        break;
                                                                }
                                                                leftCap -= 1;
                                                        }

                                                        if (leftCap < rightCap) {
                                                                positionCount -= leftCap;
                                                        } else {
                                                                positionCount += rightCap;
                                                        }
                                                }

                                                if (Character.isUpperCase(alignment.charAt(i))) {
                                                                }
                                                break;
                                        }
                                }positionCount++;
                        }

                        for (int i = 0; i < positionCount; i++) {
                                if (Character.isUpperCase(alignment.charAt(i))
                                                || alignment.charAt(i) == '-') {
                                        modStart = modStart + 1;
                                }
                        }

                        if (positionCount > alignment.length() - 1) {
                                alignChar = "Out of Bound";
                        } else {
                                alignChar = "" + alignment.charAt(positionCount);
                        }

                        System.out.println(" " + aminoAcid + " " + alignChar);

                        return modStart;
                }

} // end class
```

APPENDIX D :MATLAB Code

```
function   distances2 = Eu_distance2( textdata, trainData  )

%
%This function returns a matix with the Euclidean distance between all of
%the data points of the protein and its coordinates. Every row in these metrices
%is a be a different data point.

[numRow numCol ] = size(trainData);
insNum = numRow;

values = unique(textdata);
[numRowtext numCOltexxt] = size(values);
distances2 = [];

if(numRowtext>1)


for i =[1:numRowtext]


   index = strmatch(values(i), textdata);
   compare = trainData(index,:,:);
   distances =  pdist(compare, 'euclidean');
   distances2 = [distances2,distances];

end

end

histfit(distances2,100);
end
```