An open and interactive pipeline for variant analysis and downstream exome sequencing analysis

A Dissertation Presented

by

Astrinaki Maria

This Thesis submitted to the University of Crete in candidature for the degree of Master

University of Crete
Medical school

An open and interactive pipeline for variant analysis and downstream exome sequencing analysis

A Dissertation Presented

by

Maria Astrinaki

Dedicated to my family & to the memory of my father

ACKNOWLEDGMENTS

The completion of the final project represents one of those moments in life where it's worth to look back and take stock of all the past. Not only means the fruit of several months of development, but also symbolizes a new start in my carrier. For this new start I am extremely grateful to my supervisors, **Dr. Dimitris Kafetzopoulos and Dr. George Potamias**. Their support and guidance was very important for me.

I am thankful to the Postdoctoral Researcher of Institute of Molecular Biology and Biotechnology **Dr. Emmanouela Linardaki**, the fellow researcher of Laboratory of post-genomic applications & ancient DNA lab, Institute of Molecular Biology and Biotechnology **Dr. Despoina Vassou** and the Postdoctoral Researcher of Institute of Computer Science **Dr. Helen Latsoudis** for their support as concerns the biological part of my thesis.

Special thanks to the Postdoctoral Fellow of the Institute of Computer Science **Dr. Alexandros Kanterakis**. Without his assistance and dedicated involvement in every step throughout the process, this thesis would have never been accomplished.

I am also grateful to those outside of work who gave me help, support and encouragement including **Dr. Pantelis Topalis, Dr. Kostas Theodorakis, Georgia Soursou, Katerina Syrigonaki and Alexia Akalestou-Clocher .**

Finally, my deep and sincere gratitude to my family and especially to my daughter Chrysa for their patient, continuous and unparalleled love, help and support.

ABSTRACT

Next Generation Sequencing (NGS) is a technique for the profiling of the DNA sequence of an organism that has been constantly developed and optimized over the last 20 years. Today, the quality, automation and cost-effectiveness of this technique has reached a level that can be directly applied in diagnostic and therapeutic protocols of modern health-care systems. A typical NGS pipeline starts with sample preparation and concludes with a report that is delivered to the clinician. This report contains all findings, usually genetic mutations, with important implications and clinical relevance to the condition of the patient. Two of the most important intermediate steps of this procedure is the annotation and prioritization of the identified mutations. Annotation is the process of complementing a list of mutations with information available in existing and open genetic databases. Prioritization is the process of applying various criteria to the variants in order to rank them according to their pathogenicity. Usually the variants reported to the clinicians are the ones that have the highest rankings. Although all steps of the NGS pipeline have been thoroughly standardized for a wide set of sequencing platforms and diagnostic scenaria, prioritization remains on of the most subjective and under-standardized step. As of today, the most known systematization process for variant prioritization is the set of guidelines introduced by the American College of Medical Genetics and Genomics (ACMG). However, these guidelines are far from complete and contain many vague directions that are open to interpretation by clinical geneticists. The subjective interpretation of these guidelines combined with the plethora of annotation information for tens of thousands of mutations, creates a bewildered terrain for the researcher. To alleviate this, we create an environment, call Zazz which serves two purposes. The first is to store in a relational database the variants and the annotation information of a set of samples. The second is to offer a query and exploration environment where users can apply and fine-tune the prioritization guidelines. Zazz offers a simple User Interface through which the user can apply complex queries and also offers an interactive exploration environment that visualizes variants through modern javascript graph toolkits. Zazz is also integrated with a graphic genome browser. We demonstrate the efficiency of Zazz with exome sequencing data annotated with the IonReporter suite and the open annotation tools VEP and ANNOVAR. We also show how Zazz can accommodate custom annotation fields by importing data from a custom parser of the ClinVar database which extracts fields that other annotation platforms omit. Overall Zazz is a platform that helps clinical geneticists to browse a huge amount of information with the purpose of creating concise reports, in a timely manner.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS **5**

ABSTRACT **6**

List of tables In Thesis **9**

List of tables In Appendix **9**

LIST OF FIGURES **10**

CHAPTER 1: introduction **11**

1.1 Introduction on Basic Concepts of Biology and genetics 11

1.2 Define the Whole exome sequencing and whole genome sequencing 12

1.3 Genomic variation 13

1.3.1 Genotype to Phenotype 14

1.4 Sequencing technologies 16

1.4.1 Sanger sequencing as “first generation sequencing” 16

1.4.1.2 Next Generation Sequencing 17

1.4.1.3 Applications of NGS 19

1.5 GWAS 20

1.6 Specific Aims of this Thesis 22

Chapter 2 : Whole Exome Sequencing **23**

2.1 whole-exome sequencing (WES) 23

2.1.3 The Promise of Whole - Exome Sequencing 23

2.1.4 Variant detection in whole exome sequencing data 24

2.1.4.1 Quality assessment 26

2.1.4.1.1 Tools for Quality Assessment 26

2.1.4.2 Alignment 27

2.1.4.2.1 Tools for Alignment 27

2.1.4.3 Variant Calling 30

2.1.4.3.1 Tools for Variant Calling 31

2.1.4.4 Variant Prioritization 31

2.1.4.4.1 Tools for variant annotation 32

2.1.4.4.2 Filtering methods of annotated VCF 36

2.1.4.4.3 Repositories for Variant storage and annotation 37

2.1.4.5 Tools for Variant Prioritization 39
2.1.4.6 Visualizing Human Genome Variation 40

Chapter 3: Precision Medicine 41
3.1 Next generation sequencing in precision medicine 41
3.1.2 Application of Next-Generation Sequencing in the Era of Precision Medicine 41
3.1.3 Genomic Data Processing in Clinically Actionable Variants 42
3.1.3.1 Tools for automated curation 45

CHAPTER 4: Description of Zazz platform 46
4.1 Client/Server architecture 46
4.2 Information Visualization 47
4.2.1 Information Visualization via JavaScript 47
4.3 Asynchronous JavaScript and XML (Ajax) search engine 48
4.4 The web framework: Django 49
4.5 Architecture of Zazz Platform 49
4.5.1 Define the data 51
4.5.1.1 Multiple fields 52
4.5.2 Genome Browser 54
4.5.3 Importing Data 54
4.5.3.1 Description of the existing IonTorrent annotated file 55
4.5.3.2 Description of the existing annotated vcf files from VEP and Annovar 57
4.5.3.3 ClinVar 59
4.5.3.3.1 Tools and Environments for ClinVar annotation and filtering 61
4.5.3.3.2 Releases of ClinVar 61
4.5.3.3.3 Parsing ClinVar XML 62
4.5.3.3.4 Annotating an exome with custom ClinVar data 64
4.5.3.3.4 Importing custom ClinVar data to Zazz 64
4.5.4 The User Interface 65

CHAPTER 5: Conclusion 67

BIBLIOGRAPHY 68

List of tables In Thesis

Table 1: Summary of the best backend frameworks for web implementations
Table 2: A typical record returned from an annotation tool.
Table 3: A typical record returned from an annotation tool.
Table 4: An example of the parameters used in Zazz platform in order to import annotated fields from Ion Torrent output file.
Table 5: Example of annotated fields from Annovar in Zazz platform.
Table 6: Example of annotated fields from VEP in Zazz platform.

List of tables In Appendix

Table 1: An analytic description of how Zazz manages the fields of an annotated file from Ion Reporter.
Table 2: An analytic description how Zazz manages the annotated fields from a file which is output of Annovar. The description of each column is the same as in Table 1.
Table 3: An analytic description how Zazz manages the annotated fields from a file which is output of VEP. The description of each column is the same as in Table 1.

LIST OF FIGURES

Figure 1: Electropherogram from Sanger sequencing of a nucleotide change from C to T (mutation noted with a Y) compared to sequencing of normal control samples. This mutation is a heterozygous mutation as both alleles harbor a different nucleotide.
Figure 2: Next-Generation Sequencing pH Based Sequencing —Ion Torrent NGS.
Figure 3: The breadth of information that can be generated with high-throughput sequencing and the variety of sample sources is illustrated [36].
Figure 4:Findings of all Genome-Wide Association studies published up to December 2013 gathered across the chromosomes. Colour of the dots reflects the type of trait or disease investigated.
Figure 5: WES and impact of its genetic consequences on human public health [43].
Figure 6: Basic workflow for whole-exome and whole-genome sequencing projects. The wet lab implements the library, then, the samples are sequenced on a certain platform. Next steps are quality assessment and read alignment against a reference genome, followed by variant identification. The detected variants that answer the biological question can further be prioritized and filtered, followed by validation of the generated results in the lab [44].
Figure 7: Mappers time line. DNA mappers are plotted in blue, RNA mappers in red, miRNA mappers in green, and bisulfite mappers in purple. Gray dotted lines connect related mappers (extensions or new major versions). The time line only includes mappers with peer-reviewed publications and the date corresponds to the earliest date of publication .
Figure 8: A CRAM file aligned to a reference genomic region as visualised in Ensembl. Differences are highlighted in red in the reads, and will be called as variants.
Figure 9: A typical annotated VCF file exported by Illumina.
Figure 10: A schematic of the workflow of clinical genomic sequencing.
Figure 11: A schematic of the workflow of Whole Exome Sequencing Analysis .
Figure 12: Different Client-Server models.
Figure 13: A schematic architectonic of the Zazz platform.
Figure 14: The database schema which allows multiple-value annotation for a single variant. By applying a “CROSS JOIN” SQL function on “Transcript_multi” and “GO_multi” tables, we can generate efficiently the contents of Table 3.
Figure 15: The D3GB genome browser integrated in Zazz. Red stripes indicate variants that pass the filters during exploration.
Figure 16: The upper part of this figure shows the expanded database records. Compared to Table 1, we have added another record (rs1234567). The lower part shows the data structure that stores the connections between the primary keys (pk) of the “main” table and the “multi” table. Notice that this is a compact representation of the complete table.
Figure 17: Names of the genetic databases and their versions, used for IonReporter v. 5.6.
Figure 18 : A single variant contains a primary key to a table describing a VCV-RCV-SCV triplet.
Figure 19: The main components of the interface of Zazz. Upper left (A) shows the complete list of fields in a configuration that contains all annotations from Ion Reporter Software, VEP and ANNOVAR (total 158 fields). Upper right (B) shows filters applied in the “client-server” part and the lower (C) subfigure shows the interactive components for data exploration through the dc.js library.

CHAPTER 1: INTRODUCTION

1.1 Introduction on Basic Concepts of Biology and genetics

This thesis requires mentioning some basic biological meanings to help the reader to understand the main idea of this work. Around 1854, a monk known as Gregor Mendel started to study the inheritance pattern among the parents and their offspring, using as a model the pea plants. After years of investigation in his monastery garden, he showed that the inheritance of certain traits in his model follows particular patterns. Although his model was a plant, scientists observed that the principles of inheritance that he discovered apply to some human traits and diseases [1]. Years later, Watson JD and Francis Crick discovered the molecule, known as DNA, that exists in every living organism and has "the recipe" to build and support an organism. Specifically, they found that the components of the DNA are only four deoxyribonucleic Acid (DNA) bases {A, T, C, G}[2]. The DNA is a complex molecule consists of two big regions, the coding and non coding regions. Coding (exome) we call the regions that through the translation they produce protein for the organism. The rest of ~ 8.5% of the human genome known "junk DNA or non coding region" include other important regions such as promoters, enhancers, etc. [3]. Since ancient times, humans try to understand how and why life began; Recently Biology has succeeded through the method of sequencing to partially understand and decrypt the code to all biological life on earth as well as to understand and treat genetic diseases [4]. As DNA sequencing we mean the method that tries to determine the order of nucleotides in DNA to extract information that concerns the traits for the individual. There are two main sequencing methods; The Whole Genome Sequencing (WGS) which tries to determine all the regions in a DNA sequence of an organism's genome at a single time and the Whole Exome Sequencing that tries to determine only the coding regions [4]. Nowadays, these two methods are often used to discover rare diseases that may an individual carries. There are two main categories of diseases; In the first group are the Mendelian (or monogenic) disorders which are diseases that have caused by changes in one gene [2]. In the second group belong the multifactorial disorders; which are changes in multiple genes combine with lifestyles and environmental factors. If we want to search if an individual has a disease we compare his genome base by base with a genome which we call *"reference genome"*. The reference genome is created by scientists assembled a digital nucleic acid sequence as a representative example of the human genome. To assembly the reference genome scientists use a number of individuals, so it is like a haploid mosaic of different DNA sequences from each individual[1]. If there is a different base (variant) in our sample we should examine if it has associated with a rare disease. But how can we understand that a variant leads to a rare disease? American College of Medical Genetics and Genomics (ACMG) has published some guidelines and recommendations that the scientists should follow

[1] https://www.ncbi.nlm.nih.gov/genome/guide/human/

to classify variants in one of the five categories (pathogenic, likely pathogenic, uncertain significance, likely benign, and benign). These categories have used to describe variants identified in genes that cause Mendelian disorders [5].

1.2 Define the Whole exome sequencing and whole genome sequencing

The whole human genome which contains ~3 × 10 9 bp includes coding and non-coding regions. Although, scientists can sequence all the genome by Whole Genome Sequencing method many reasons have contributed to focus the sequencing only in coding regions (exomes) with the most important to support in the findings that 85% of the disease-causing mutations located in coding and functional regions of the genome. Moreover, other secondary reasons like limitations in technology, or lack of knowledge about the specific locations of regulatory elements in DNA have contributed to focus only on exomes. We use Whole exome when we want to find rare variants, mostly monogenic, genetic disorders or variants that have been associated with common diseases and cancers [6].

1.3 Genomic variation

Although there are billions of people living on earth, each human differs from any other humans about 0.1%. This small amount of difference between us have been written in our DNA sequences and we call it "*genomic variation"*. Due to the Human Genome Project [7] [8] researchers have understood deeper the variations that occur in human genetic code. Genetic Variation arises with every generation through events like DNA mutations, deletions or recombination events. More analytically in DNA occurs the below events :

Single nucleotide polymorphism (SNP)

Single Nucleotide Polymorphism, we call the variations where a nucleotide in a sequence differs from the normal in at least one percent of the population. Scientists have estimated that the most changes (~90% of all human variation) in a DNA sequence are SNPs [9]. Moreover, we often meet more SNPs transitions (A ↔ G or C ↔ T) instead of SNPs transversions (A ↔ C or T; and G ↔ C or T) [10].

Copy number variation (CNV)

Copy number variations are large-scale changes that occur in many locations throughout the human genome. These changes which maybe are insertions, deletions, inversions or duplications have as result changes in the physical structure of genes on chromosomes. A CNV maybe is a DNA segment of one kilobase (kb) or larger that is present at a variable copy number in comparison with a reference genome [11]. Although, not all the CNVs events influences the phenotype, it was found by researchers that exist CNVs are linked with disease. CNVs' analysis in the human and chimpanzee genomes have shown that CNVs have greater role than the SNPs in human species evolution [12].

Variant number of tandem repeats (VNTR)

VNTR are sequences that repeat their self without other nucleotides in between them. There are two VNTRs categories according to their length, the minisatellite and microsatellite. In the first category, the size of the repeat sequence is generally ten to one hundred base pairs and the number of times the sequence repeats varies from about five to fifty times, while in the second category the repeat sequence is generally 1 to 6 nucleotides [13].

Epigenetics

The term "Epigenetics" was used for the first time by Conrad Waddingthon in the early 1940s [14] He defined the term "Epigenetics" as ''t*he branch of biology which studies the causal interactions between genes and their products which bring the phenotype into being"* [15]. Nowadays, with the term of epigenetic we mean ''*the study of changes in gene function that are mitotically and/or meiotically heritable and that do not entail a change in DNA sequence*.'' [16]. The epigenetic changes are DNA either methylation or histone modifications. We know that DNA methylation is heritable and may also last for multiple generations, while heritability of histone modifications is unknown [17].

Several international efforts have been made to catalogue human variation. Among them the 1000 Genomes Project [18] and International HapMap Project [19] worked upon the common patterns of DNA sequence variation in the human genome and the extent of variation between populations . On one hand the International HapMap Project described the patterns of common SNPs within the human DNA sequence, while, 1000 Genomes Project provided a map of both common and rare SNPs. Genome wide association studies (GWAS) used the results of these studies which were public available to find variants that associate with a disease. Due to the need for a general catalog of genome variation the National Center for Biotechnology Information (NCBI) has established the Single Polymorphism Database (dbSNP) [20] that is used as a central, public repository for genetic variation, where each variation has a unique reference identifier (rsID)

1.3.1 Genotype to Phenotype

'We sometimes seem to have forgotten that the original question in genetics was not what makes a protein but rather 'what makes a dog a dog, a man a man.' (Noble 2006)

Wilhelm Johannsen in 1911 was the first who understood that the results of genetic variation is the variability in phenotype and he tried to define what genotype and phenotype is by working the answer to the following question "How genotypes map over phenotypes" self-fertile common bean [21]. However, in 1865 Gregor Mendel proposed the principles of how traits may be inherited from one generation to another using as model the common pea plan. Although the principles of Mendel did not have as model the humans his research helped clinicians in human disease research;such as, Archibald Garrod applied Mendel's principles to his study of alkaptonuria. Today, whether we are talking about pea plants or human beings, genetic traits that follow the rules of inheritance that Mendel proposed called Mendelian [22].
The principles are :

Law of Segregation:
The Law of Segregation supports that every offspring has two alleles for each trait. One allele from his mother and the other allele from his father. The trait that the offspring finally has, depends on the combination of dominant or recessive alleles.

Law of Independent Assortment:
The allele a gamete receives for one gene does not influence the allele received for another gene. As I mentioned earlier, genomic variation not only influence physical traits (i.e eye-color, hair color etc) or ancestry but, also, is, partly, responsible for our susceptibility to various diseases.
such as, diseases like Huntington's disease, cystic fibrosis or sickle-cell follow the Mendelian disorder. These diseases are highly heritable. Due to high penetrance of risk alleles, most of those genetic disorders are rare and have been prevented by natural selection.
OMIM, Online Mendelian Inheritance in Man, is the database that focus on inherited genetic diseases in humans. Until 2008, OMIM reported 387 human genes of known sequence with a known phenotype, and 2,310 human phenotypes with a known molecular basis. However, OMIM also reported 1,621 confirmed Mendelian phenotypes for which the molecular basis is not known. Furthermore, OMIM reported 2,084 phenotypes for which a Mendelian basis is suspected but has not been fully established, or that may exhibit overlap with other characterized phenotypes. [23] Unfortunately, many of the common diseases have a more complex inheritance pattern and are associated with mutations in multiple genes this makes the discovery of their

genetic motif a rather challenging task. As a result, research efforts have begun to focus on polygenic disease, which can involve complex interactions between genes and the environment.

1.4 Sequencing technologies

1.4.1 Sanger sequencing as "first generation sequencing"

"... [A] knowledge of sequences could contribute much to our understanding of living matter."

Frederic Sanger

In 1977 two teams developed the first sequencing technologies. The first team was Sanger's team [24] The second team was Maxam's [25]. Their discovery was not only very important for the study of the genetic code of living beings but also, very inspiring for faster and efficient sequencing technology development [26].

Fred Sanger believed that the knowledge of the chemical structure of biological molecules was important to understand the primary sequence of the organisms. Sanger determined the first protein sequence, of insulin. Ten years later he analysed RNA sequencing with the following procedure : an RNA species was first fragmented by RNases, next the pieces were separated by chromatography and electrophoresis, then fragments were deciphered by sequential exonuclease digestion, and then sequence was extracted from the overlaps [27]. Frederick Sanger developed a new method for DNA sequencing based on the chain termination method, now known as the Sanger sequencing method (SSM). Sanger sequencing method works as follows the nucleotides in a single-stranded DNA molecules are determined by complementary synthesis of polynucleotide chains, based on the selective incorporation of chain-terminating dideoxynucleotides (ddNTP) driven by the DNA polymerase enzyme [28]. Nowadays using the power of computer processing we can automate the Sanger Sequencing results. Each ddNTP (ddATP, ddGTP, ddCTP, ddTTP) includes a fluorescent marker. When a ddNTP is attached to the elongating sequence, the base will fluoresce based on the associated nucleotide. Scientists have agreed that , A is connected by green fluorescence, T by red, G by black, and C by blue. A laser within the automated machine detects a fluorescent intensity that is translated into a "peak.".When a heterozygous variant occurs within a sequence, loci will be captured by two fluorescent dyes of equal intensity. When a homozygous variant is present, the expected fluorescent color is replaced completely by the new base pair's color (figure 1) [29].

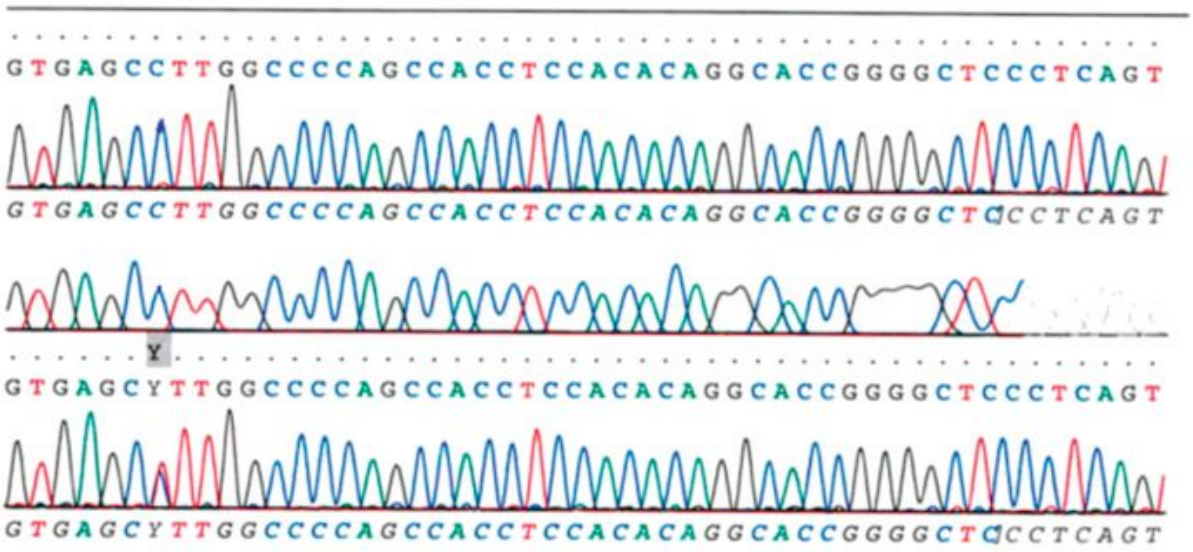


Figure 1: Electropherogram from Sanger sequencing of a nucleotide change from C to T (mutation noted with a Y) compared to sequencing of normal control samples. This mutation is a heterozygous mutation as both alleles harbor a different nucleotide. [2]

1.4.1.2 Next Generation Sequencing

Scientists used for years Sanger and Maxam-Gilbert techniques as sequencing methods but due to the emerging need for larger scale sequencing new technologies had to be introduced. In 2005 was the first appearance of these new technologies, known as Next Generation Sequencing technologies, by introducing Roche 454 platform in the market [30]. From 2005 to 2007 different companies like Illumina/Solexa and LifeTechnologies/ABI proposed different platforms. In our lab we use two sequencers the Ilumina and Torent sequencers. Because the sample that we used as input in our platform was sequenced in Ion Proton sequencer, in the paragraphs below we will describe the method that Ion Torrent sequencer[3] uses.

1. **Library preparation**

We fragment the input DNA/RNA(converted to DNA) into a size from 200 to 400b and then the scientists add common sequencing adapters.

2. **Template Prep/Amplification**

[2] Source https://www.sciencedirect.com/topics/neuroscience/sanger-sequencing

[3] https://allseq.com/knowledge-bank/sequencing-platforms/ion-torrent/

The fragments from the library preparation are attached to beads and amplified using emulsion PCR (emPCR). The purpose of this step is to cluster all the original molecules and their copies in the same position.

3. Sequencing

Ion Torrent's systems sequencing technology is based on pyrosequencing chemistry, applying a form of 'sequencing by synthesis'. In this method, individual bases are introduced one at a time and incorporated by DNA polymerase.

4. Data analysis

The outputs of Ion torrent are simple like FASTQ. For data analysis Ion Torrent offers tools like the 'Torrent Browser' software and the cloud-based solution called 'Ion Reporter' which serves as a front-end for a variety of open source analysis solutions.

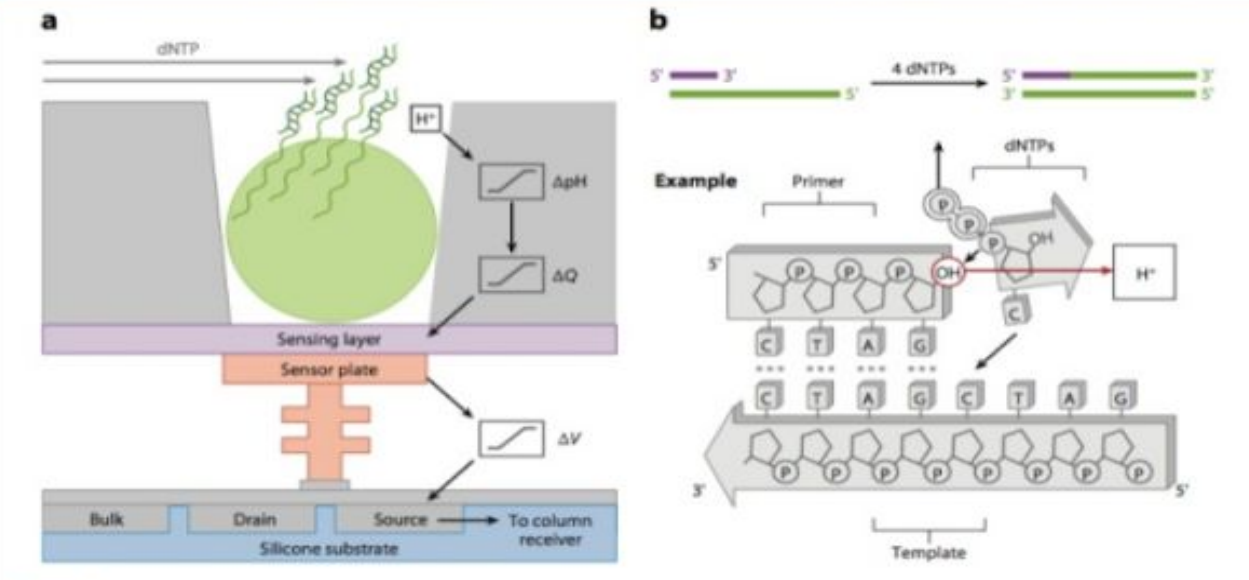


Figure 2: Next-Generation Sequencing pH Based Sequencing —Ion Torrent NGS.

Summarizing, the main benefit of NGS methods is that do not need a bacterial cloning procedure and prepare libraries for sequencing in a cell free system [31].

1.4.1.3 Applications of NGS

There are many applications for NGS and new methods being developed continuously. In this section, we will describe some of them.

- To build a new genome from unknown organisms, researchers use a tool that is called "assembler." Assemblers put fragmented reads of DNA together like a puzzle by aligning regions with overlap to build a genome sequence [32].
- To compare a genome sequence with a reference genome to find the genetic variation. For these cases we use whole exome or whole genome NGS technologies [33].
- To analyze transcriptome results with sequencing, researchers synthesize complementary DNA from RNA for sequencing. Using this method researchers examine splicing of RNA, gene fusion, mutation, and differential gene expression [34].

NGS technologies are also important for microbial ecology scientists to investigate genetic materials from environmental samples on a tremendous scale [35].

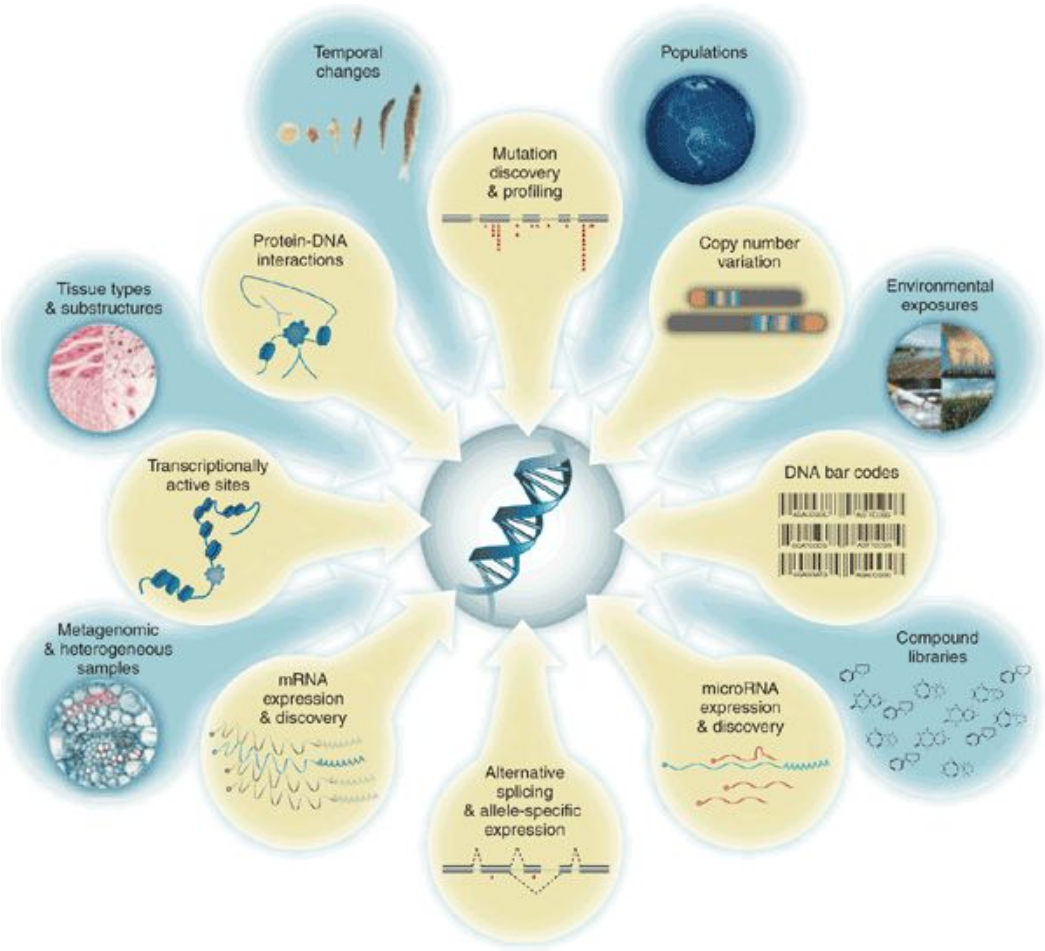


Figure 3: The breadth of information that can be generated with high-throughput sequencing and the variety of sample sources is illustrated [36]

1.5 GWAS

The purpose of Genome-Wide Association Studies (GWAS) is to find alleles that correlate to different diseases and traits [37]. The basic design of a GWAS experiment is simple and includes two different groups, the individuals that are affected by a disease (cases) and individuals without the disease (controls), such as, between individuals with schizophrenia and healthy controls. The next step of a GWAS study is genotyping the samples, each sample is usually genotyped around 500.000 to 1 million SNPs and then the differences in minor allele frequency (MAF) of the SNP are investigated between cases and controls. The statisticians use Chi-squared test to find the p-value of the significance of difference in minor allele frequency and then they use Bonferroni correction [38].

GWAS catalog includes all eligible GWAS studies since the first published GWAS on age-related macular degeneration in 2005 [39]. As of 1st September 2016 it contains 24,218 unique SNP-trait associations from 2,518 publications in 337 different journals (Figure 4).

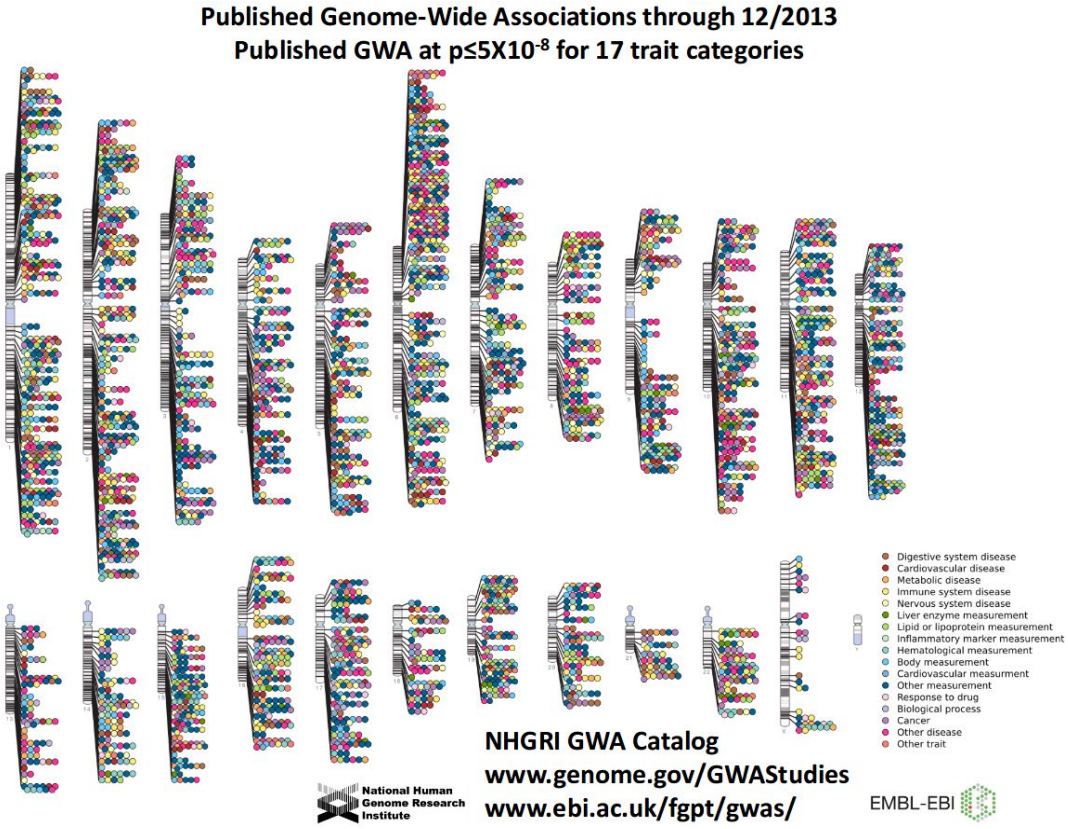


Figure 4: Findings of all Genome-Wide Association studies published up to December 2013 gathered across the chromosomes. Colour of the dots reflects the type of trait or disease investigated. [4]

The statistics show that despite the success of GWAS in detecting new disease associations, we are far from applying the findings into clinical settings. For example, the majority of SNPs almost 90% were not in the coding region of a gene, and only 8% were non-synonymous variants Indeed, 43% were not in a known gene and 23% were not within 20,000 bases of a known gene. We also should take into account the concept of linkage disequilibrium (that blocks of chromosomes are inherited together and can be 'tagged' by a defined, limited set of SNPs within those blocks), although, we believe that the present technology which is used

[4] Source: www.genome.gov/gwastudies .

captures the majority of common variants, it is possible that some genetic variants that may be important susceptibility alleles are not covered by the SNPs that are genotyped [40].

The scientists in GWAS studies hypothesizes that common diseases may be caused, in part, by common genetic variants [41]. Under this sense GWAS designed to detect associations between disease and common SNPs (e.g., minor allele frequency >5%). The SNP arrays measure variants primarily at or above this frequency means that some common variant may not have been detected [42]. However, we should have in mind, that common diseases are undoubtedly also due to rare variants. These variants can act alone or reflect allelic heterogeneity, whereby many different rare alleles within a particular locus each increase risk. The inability of existing GWAS to evaluate rare variants may help explain why they account for little heritability.

Potential explanations for the missing heritability could be based on other factors such as rare variants, copy number variations, epigenetics and environmental exposures influencing the final phenotype [40].

CHAPTER 2 : WHOLE EXOME SEQUENCING

2.1 whole-exome sequencing (WES)

2.1.3 The Promise of Whole - Exome Sequencing

The WES method helped the scientists to find the role of more than 150 genes in various diseases, and these statistics are quickly growing. Identification of variants causing the disease brings the research into clinical practice.WES method tries to provide effective genetic data that could be used from clinicians for best treatment; albeit, accurate, fast and cost-effective diagnosis of the patients. As we observe in figure 5 WES could improve health care in various levels (figure 5) A common example is the drug discovery or finding disease mechanism.

We can classify the detectable variants into two groups. The first group is the disease-causing variants with large pathogenic effect (high penetrance) mostly seen in single gene disorders. These variants are mainly rare (maf<1%). Although some variants are in a small amount of individuals that are associated with rare or uncommon diseases, they categorize as likely disease-causing variants with less certainty of the variants causing the disease due to incomplete penetrance. NGS approach would verify these variants, which is helpful for the individuals carrying such variants. The other group is the variants with higher frequency and lower penetrance in cases than controls based on genome-wide association studies. These variants could be detected with DNA chip genotyping and NGS approaches. WES which is able to find these variants we can use them in clinical management of individuals, for example, in familial hypercholesterolemia dietary management in individuals having the causal variants is lifesaving. High penetrance variants detected by WES are important in diagnosis of the patients and healthy carriers [43].

One main limitation of Whole exome sequencing relies on coverage. Although exomes suppose that cover all the protein-coding regions of the genome, the average coverage in many platforms

tends to be between 85 and 95%. This means that a particular gene of interest may not covered, completely or partially.

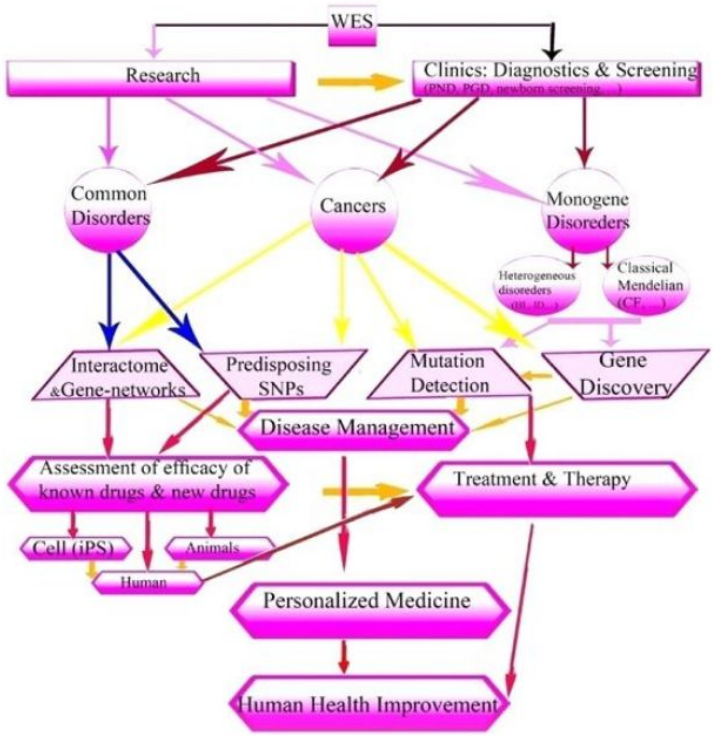


Figure 5: WES and impact of its genetic consequences on human public health [43].

2.1.4 Variant detection in whole exome sequencing data

The main steps to analyze data from Whole exome or whole genome sequencing are briefly described in figure 6. In the following chapters there is analytical description for each procedure.

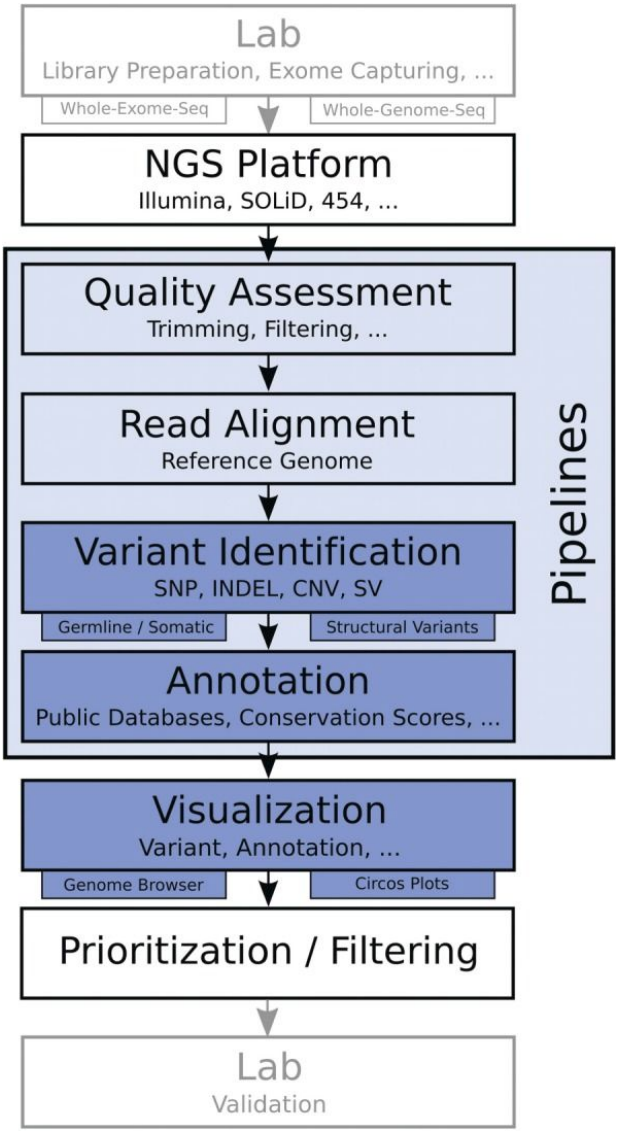


Figure 6: Basic workflow for whole-exome and whole-genome sequencing projects. The wet lab implements the library, then, the samples are sequenced on a certain platform. Next steps are quality assessment and read alignment against a reference genome, followed by variant identification. The detected variants that answer the biological question can further be prioritized and filtered, followed by validation of the generated results in the lab [44].

There are four different parts to detect Single Nucleotide Variants (SNVs) and small insertions and deletions (indels) from raw NGS data:

1. **Quality assessment -** Reads are processed to have a certain quality standard

2. **Alignment** - The NGS reads must be mapped to a reference genome.

3. **Variant Calling-identification** - Variants that differs from the reference genome are identified

4. **Variant Annotation** - Variants are filtered to remove low quality variants and annotated (they described) with additional information from biological databases .

Basic concepts of these four tasks are introduced in the following paragraphs.

2.1.4.1 Quality assessment

After completing the sequencing run, follows the quality evaluation of raw reads. Raw data which is generated by any sequencing platforms includes sequence artifacts such as base calling errors or poor quality reads. That kind of errors are closely associated with the chemistry that the platform uses or with instrument failures.

Three major approaches are commonly used for the pre-processing of the reads: quality trimming, that is the polishing of the reads based on descriptive statistics calculated on their quality scores; PCR de-duplication, for elimination of identical reads or read pairs that might derive from PCR amplification of the same DNA fragment; merging of overlapping pairs, that finds pairs of reads originating from DNA fragments shorter than the combined length of the mates, into a longer, non-redundant sequence.

2.1.4.1.1 Tools for Quality Assessment

Only a few standalone tools for quality assessment of NGS data are publicly available; other than commercial software supplied with the sequencing machines, which are not sufficiently optimal. NGSQC Toolkit [64] is consisted of various easy-to-use standalone tools for quality check and filtering, trimming, generating statistics and conversion between different file formats/variants of NGS data from Illumina and Roche 454 platforms. The toolkit is based on user-friendly options to provide an automatic and fast parallel process of large amounts of sequence data. PRINSEQ [65]is another open-source application which implemented in Perl and we can use it as a standalone version or accessed online through a user-friendly web interface. The above tools have as outputs summary graphs and tables to quickly assess the data

quality. SolexaQA [66], is a another free user-friendly software package written in Perl which accept as input data one (or multiple) sequence read files in Solexa- or Illumina-style FASTQ format. This package contains software to trim sequences dynamically using the quality scores of bases within individual reads. SolexaQA has as output statistics data using package R and graphics and heatmaps using the heatmap visualizer matrix2png. Many scientists use FastQC tool [67] is free and user-friendly tool written in java language. BAM, SAM or FastQ files are the input data. The main export is a report on HTML format, graphs and tables to quickly assess user's data. Offers offline operation to allow automated generation of reports without running the interactive application. The main feature is that provide a quick overview to inform the user where the problems occurs. The FASTX-Toolkit is suitable for Short-Reads FASTA/FASTQ files preprocessing and uses a set of command line tools. These tools can be used in two forms: Web-based through Galaxy or running the tools from command line (or as part of a script)[68]. TagCleaner [69] is a publicly available web application that is able to automatically detect and efficiently remove tag sequences from metagenomic dataset. TagCleaner has design to filter the trimmed reads for duplicates, short reads, and reads with high rates of ambiguous sequences.

2.1.4.2 Alignment

After the variant discovery step, typically, the results have stored in one or more text files in FASTQ format file. This file, contains millions of short reads together with quality values for each base. Generally to analyze the output of next-generation sequencing, the reads need to be either assembled (de novo assembly) or aligned to a reference genome.
As concerns the human samples there are two main sources for the reference genome assembly: the University of Santa Cruz (UCSC), which is also hosting the central repository for ENCODE data [62] and the Genome Reference Consortium (GRC), which focuses on creating reference assemblies [63]. Both resources provide several versions of the human genome. UCSC offers versions hg18 and hg19 while GRC provides GRCh36 and GRCh37. Together these are the most widely used reference genomes. UCSC (hg) and GRC (GRCh) human assemblies only differ to their nomenclature (e.g. UCSC uses a 'chr' prefix) [44].

2.1.4.2.1 Tools for Alignment

Although, since 2007 there is a rapid increase in alignment tools (figure 7), the choice of the right alignment tool from the researcher is not an easy case; he needs to answer questions like

- What kind of sequences/experiment do I have?
- What sequencing platform do I have ? Do I deal with Illumina,Ion Torrent for example ? Each machine has its own characteristic read length and error types
- What kind of further analysis should I do with my alignments?

- What kind of computing support doI have? Can I run my computations on a server or a cluster or does it need to run on my desktop computer?

According to the above concerns we can understand there is no ideal alignment tool for all the cases not even the most updated [45].

In 1970 created the first automated sequencing method by Saul Needleman and Christian Wunsch. It is a dynamic programming algorithm for sequence alignment. This algorithm solves the original problem by dividing the problem into smaller independent sub problems. It has the following steps:

- Initialization of score matrix.
- Calculation of scores and filling traceback matrix.
- Determining alignments from traceback matrix.

The algorithm tries to explain global sequence alignment for aligning nucleotide or protein sequences which are similar in length as well as similar across the length [46]. In nearly 2002 implemented the MUMmer algorithm to align DNA sequence; it is based on Suffix trees concepts. Generally, we use the suffix tree concept for internal representation of Genome sequence. MCAlign tool developed in year 2004 and we mention it, as an example of failure for dynamic programming concepts. This tool used the algorithm called stochastic hill-climbing for finding the alignment between the sequences. It uses the model of frequency distribution of insertion/deletion events. MCAlign tool failed for long insertions and data containing errors [46]. In 2008 Li R et al. Developed a short oligonucleotide alignment program (SOAP). We can use SOAP for gapped and ungapped alignment of short oligonucleotides over reference sequences. This alignment tool is a command-driven program, which supports multi-threaded parallel computing. In this method reference genome sequence loads into an array or hash table for building index [47]. SeqMap tool ,also, implemented in 2008 and uses the same concept of hash table indexing. It is a tool for mapping large amount of oligonucleotide to the genome. With carefully designed index-filtering algorithm and delicate implementation, SeqMap can efficiently map millions of short sequences to a genome of several billions of nucleotides. While doing the mapping, several mutations and insertions/deletions of the nucleotide bases in the sequences be tolerated and furthermore detected [48]. In 2009 invented by Langmead the Bowtie algorithm. Bowtie have characterized by their creators as "*an ultrafast, memory-efficient alignment program for aligning short DNA sequence reads to large genomes*". Thanks to the use of Burrows-Wheeler transformation Bowtie has great speed [49]. BWT resorts a given input string, such that equal letters tend to occur in groups, which allows efficient compression. In 2009 implemented also the Burrows-Wheeler Alignment tool (BWA). It's a new read alignment package based on backward search with Burrows–Wheeler Transform (BWT). BWA efficiently align short sequencing reads against a large reference sequence such as the human genome,

allowing mismatches and gaps [50]. BWA uses BWT to store a compressed prefix trie of the reference genome in memory. A prefix trie is a data structure that stores every prefix of a string such that every exactly repeated substring is only stored once. This allows to search for a string, e.g. a part of a NGS read, in a prefix trie in linear time. Comparing the speed between Bowtie and BWA the second one is faster. CloudBurst developed in 2009 is a new highly sensitive parallel seed-and-extend read-mapping algorithm optimized for mapping single-end next generation sequence data to reference genomes. CUSHAW belongs to the group of next-generation sequencing read alignment software package. This tool based on multi-core and many-core computing. CUSHAW's aligner designed based on the Burrows-Wheeler transform (BWT) and programmed using CUDA C++ parallel programming language. The execution time of CUSHAW tool is less than Bowtie, BWA and SOAP2 [51].

.

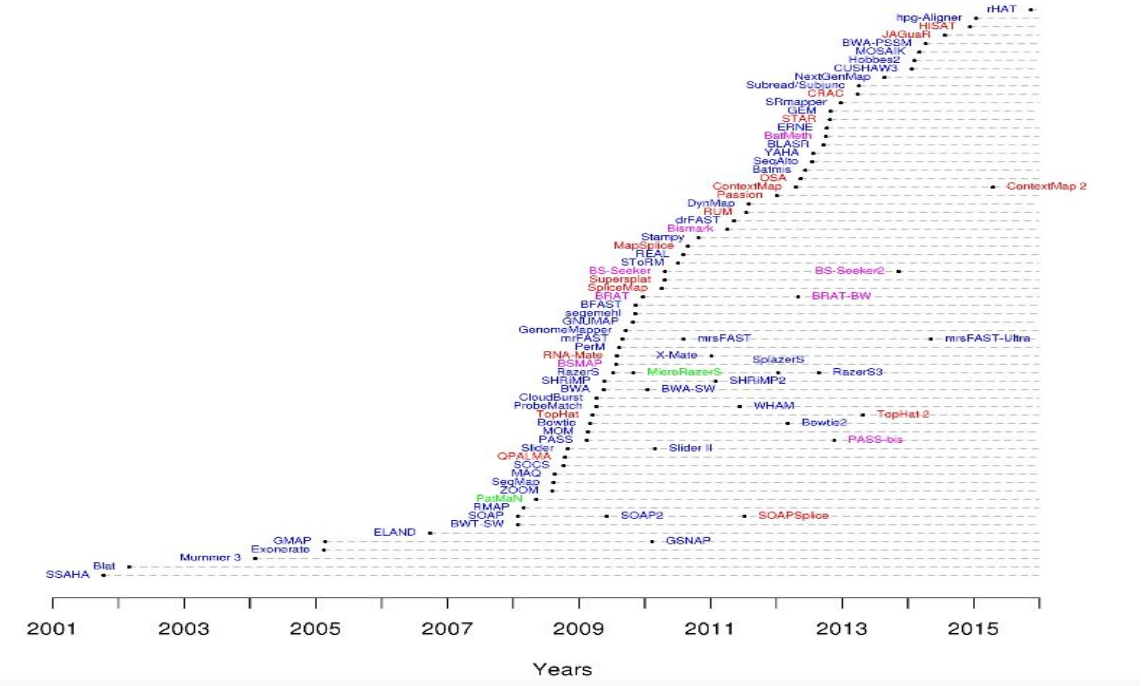


Figure 7: Mappers time line. DNA mappers are plotted in blue, RNA mappers in red, miRNA mappers in green, and bisulfite mappers in purple. Gray dotted lines connect related mappers (extensions or new major versions). The time line only includes mappers with peer-reviewed publications and the date corresponds to the earliest date of publication [5].

[5] https://www.ebi.ac.uk/~nf/hts_mappers/

Summarizing, we have two groups of alignment algorithms, the first group includes the algorithms that use hash table indexing, like SeqMap or MOSAIK and the second group includes the algorithms that use some sort of compressed tree indexing based on the Burrows–Wheeler transform like Bowtie or BWA. Most alignment algorithms follow the seed-and-extend paradigm like SOAP. [44]

2.1.4.3 Variant Calling

Variant calling is the process of identifying the genetic differences between an examined genome and the reference one, such as single nucleotide variants, copy number variations, structural variants, like indels inversions, and fusion genes. In this project, we only look at single nucleotide variants and indels (figure 8).

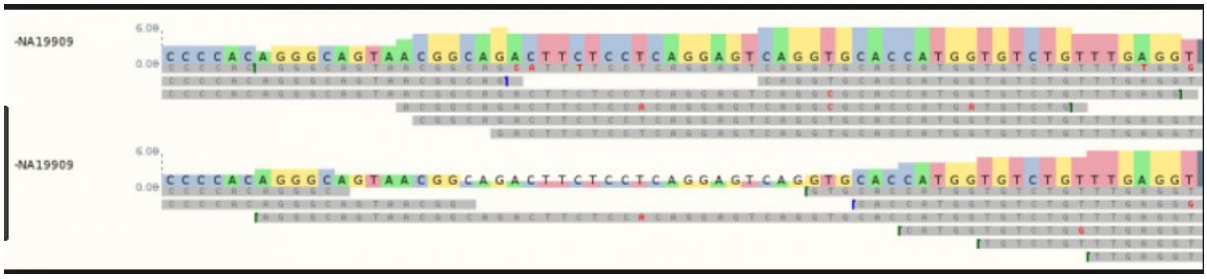


Figure 8: A CRAM file aligned to a reference genomic region as visualised in Ensembl. Differences are highlighted in red in the reads, and will be called as variants. [6]

We can use special tools to visualize the variants, identifying loci where reads have aligned with high confidence, but differ from the reference genome. By further inspecting the aligned reads at these positions, genotypes can inferred. For example, if 50 out of 100 reads have different base at a certain locus, this might indicate that the individual in question is heterozygous at this locus. If all reads have different base, the individual might be homozygous for the allele in question.

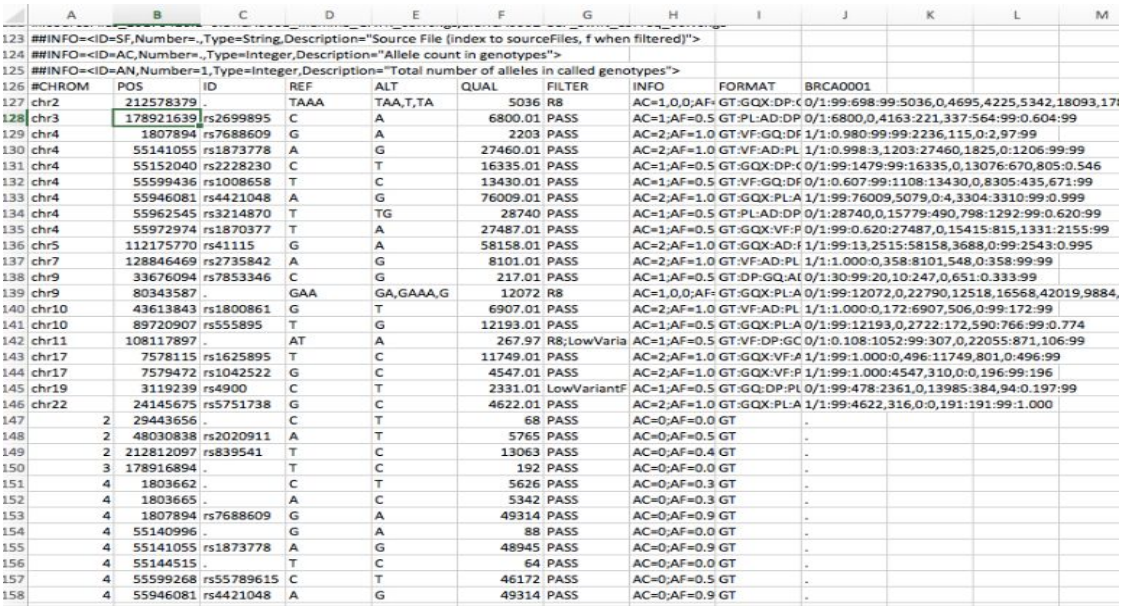
[6] https://www.ebi.ac.uk/training/online/course/human-genetic-variation-i-introduction/variant-identification-and-analysis/what-variant

2.1.4.3.1 Tools for Variant Calling

In the past eight years more than 40 open-source tools for variant calling have developed. All these tools have one common aim: to call variants in NGS data with high sensitivity and precision. The difference among all these tools is the algorithm that they use to apply variant calling. There the tools that use bayesian approaches like, GATK, Platypus, FreeBayes and SAMtools. Moreover, there are tools like VarScan that uses a heuristic/statistical method to identify variants. SNVer relies on a frequentist approach, while LoFreq uses a Poisson-binomial distribution. There is a group of tools like GATK, VarDict or Strelka that apply local realignment to improve indel calling. Usually the tools provide a set of parameters characterizing the reported variants and recommendations for filtrering [52]. But what considerations should we follow to choose the appropriate algorithm for the variant calling? The choice of variant caller largely depends on the kind of variants that we are interested in. For example, while the most variant callers report SNVs, some offer indel and/or SV detection.
The desired variant allele frequency (VAF) is another important parameter. Variant callers based on joint genotype SomaticSniper [53], FaSD-somatic [54], JointSNVMix2 [55], Virmid [56], SNVSniffer [57] and Seurat [58] use the assumption that diploidy in both tumor and normal and evaluates the likelihood of the joint genotypes. These tools are designed for WGS, WES, or targeted sequencing and they are not sensitive enough to detect low-frequency variants, To call low-frequency variants, especially with high-coverage targeted sequencing data, one should choose variant callers that model allele frequencies directly (Strelka [59], MuTect, LoFreq [60], EBCall [61]).

2.1.4.4 Variant Prioritization

Although Variant calling in WES data result in hundreds of variants, most of them may not contribute to any phenotypic condition. As a result, a significant challenge that the scientists need to address is to understand the functional content within the data and therefore perform prioritization analysis on all variants for functional follow-up on selected variants. The first step in variant prioritization is the annotation, which is the process of adding from various genomic databases functional information to DNA variants to describe the variants.
There are many different types of information that could be associated with variants, with the most common being, the functional impact of the variant on the gene/protein product, its frequency in the population, disease association and variant-drug association



	A	B	C	D	E	F	G	H	I	J	K	L	M
123	##INFO=<ID=SF,Number=.,Type=String,Description="Source File (index to sourceFiles, f when filtered)">												
124	##INFO=<ID=AC,Number=.,Type=Integer,Description="Allele count in genotypes">												
125	##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">												
126	#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	BRCA0001			
127	chr2	212578379	.	TAAA	TAA,T,TA	5036	R8	AC=1,0,0;AF=	GT:GQX:DP:(	0/1:99:698:99:5036,0,4695,4225,5342,18093,17			
128	chr3	178921639	rs2699895	C	A	6800.01	PASS	AC=1;AF=0.5	GT:PL:AD:DP	0/1:6800,0,4163:221,337:564:99:0.604:99			
129	chr4	1807894	rs7688609	G	A	2203	PASS	AC=2;AF=1.0	GT:VF:GQ:DF	1/1:0.980:99:99:2236,115,0:2,97:99			
130	chr4	55141055	rs1873778	A	G	27460.01	PASS	AC=2;AF=1.0	GT:VF:AD:PL	1/1:0.998:3,1203:27460,1825,0:1206:99:99			
131	chr4	55152040	rs2228230	C	T	16335.01	PASS	AC=1;AF=0.5	GT:GQX:DP:(	0/1:99:1479:99:16335,0,13076:670,805:0.546			
132	chr4	55599436	rs1008658	T	C	13430.01	PASS	AC=1;AF=0.5	GT:VF:GQ:DF	0/1:0.607:99:1108:13430,0,8305:435,671:99			
133	chr4	55946081	rs4421048	A	G	76009.01	PASS	AC=2;AF=1.0	GT:GQX:PL:A	1/1:99:76009,5079,0:4,3304:3310:99:0.999			
134	chr4	55962545	rs3214870	T	TG	28740	PASS	AC=1;AF=0.5	GT:PL:AD:DP	0/1:28740,0,15779:490,798:1292:99:0.620:99			
135	chr4	55972974	rs1870377	T	A	27487.01	PASS	AC=1;AF=0.5	GT:GQX:VF:P	0/1:99:0.620:27487,0,15415:815,1331:2155:99			
136	chr5	112175770	rs41115	G	A	58158.01	PASS	AC=2;AF=1.0	GT:GQX:AD:F	1/1:99:13,2515:58158,3688,0:99:2543:0.995			
137	chr7	128846469	rs2735842	A	G	8101.01	PASS	AC=2;AF=1.0	GT:VF:AD:PL	1/1:1.000:0,358:8101,548,0:358:99:99			
138	chr9	33676094	rs7853346	C	G	217.01	PASS	AC=1;AF=0.5	GT:DP:GQ:AI	0/1:30:99:20,10:247,0,651:0.333:99			
139	chr9	80343587	.	GAA	GA,GAAA,G	12072	R8	AC=1,0,0;AF=	GT:GQX:PL:A	0/1:99:12072,0,22790,12518,16568,42019,9884,			
140	chr10	43613843	rs1800861	G	T	6907.01	PASS	AC=2;AF=1.0	GT:VF:AD:PL	1/1:1.000:0,172:6907,506,0:99:172:99			
141	chr10	89720907	rs555895	T	G	12193.01	PASS	AC=1;AF=0.5	GT:GQX:PL:A	0/1:99:12193,0,2722:172,590:766:99:0.774			
142	chr11	108117897	.	AT	A	267.97	R8;LowVaria	AC=1;AF=0.5	GT:VF:DP:GC	0/1:0.108:1052:99:307,0,22055:871,106:99			
143	chr17	7578115	rs1625895	T	C	11749.01	PASS	AC=2;AF=1.0	GT:GQX:VF:A	1/1:99:1.000:0,496:11749,801,0:496:99			
144	chr17	7579472	rs1042522	G	C	4547.01	PASS	AC=2;AF=1.0	GT:GQX:VF:P	1/1:99:1.000:4547,310,0:0,196:99:196			
145	chr19	3119239	rs4900	C	T	2331.01	LowVariantF	AC=1;AF=0.5	GT:GQ:DP:PL	0/1:99:478:2361,0,13985:384,94:0.197:99			
146	chr22	24145675	rs5751738	G	C	4622.01	PASS	AC=2;AF=1.0	GT:GQX:PL:A	1/1:99:4622,316,0:0,191:191:99:1.000			
147	2	29443656	.	C	T	68	PASS	AC=0;AF=0.0	GT	.			
148	2	48030838	rs2020911	A	T	5765	PASS	AC=0;AF=0.5	GT	.			
149	2	212812097	rs839541	T	C	13063	PASS	AC=0;AF=0.4	GT	.			
150	3	178916894	.	T	C	192	PASS	AC=0;AF=0.0	GT	.			
151	4	1803662	.	C	T	5626	PASS	AC=0;AF=0.3	GT	.			
152	4	1803665	.	A	C	5342	PASS	AC=0;AF=0.3	GT	.			
153	4	1807894	rs7688609	G	A	49314	PASS	AC=0;AF=0.9	GT	.			
154	4	55140996	.	G	A	88	PASS	AC=0;AF=0.0	GT	.			
155	4	55141055	rs1873778	A	G	48945	PASS	AC=0;AF=0.9	GT	.			
156	4	55144515	.	T	C	64	PASS	AC=0;AF=0.0	GT	.			
157	4	55599268	rs55789615	C	T	46172	PASS	AC=0;AF=0.5	GT	.			
158	4	55946081	rs4421048	A	G	49314	PASS	AC=0;AF=0.9	GT	.			

Figure 9: A typical annotated VCF file exported by Illumina [7].

2.1.4.4.1 Tools for variant annotation

Although, there are plenty of annotator tools, in this section, we will focus on the most popular tools that the scientific community uses for the moment.

We can use MuTect which is part of the GATK pipeline to detect somatic Single Nucleotide Polymorphisms in whole exome sequencing data. MuTect takes as input sequence data from matched tumor and normal DNA after alignment of the reads to a reference genome and standard preprocessing steps. This tool has a main advantage in terms of its tradeoff between specificity and sensitivity. The MuTect sensitivity derives from the variant detection statistical test, which includes estimation of the allele fraction of the event, and the working point chosen along the ROC curve. MuTect due to the carefully tuning of the filters is able to reject true false positive calls without sacrificing sensitivity [62]. ANNOVAR, one of the most popular tools, [63] developed in 2010 with the aim to rapidly annotate millions of variants with ease; is a command line tool that could be executed on a variety of operating system if we have install a Perl interpreter. It can take many input formats, including the most commonly used VCF format, and it outputs an annotated variant file in several different formats (such as annotated VCF file, tab-delimited text file or comma-delimited text file), which contains annotations for each variant in the input file. The described tool provides a wide variety of different annotation techniques, organized in the categories gene-based, region-based and filter-based annotation. The tool uses several databases, which need to download individually. This means that the user

[7] Source http://homer.ucsd.edu/homer/ngs/annotation.html

should download the most up to date database version. Also, he avoids to download large unnecessary data sets [44]. For the researchers that prefer a graphical user interface environment they developed a web server called wANNOVAR [64]. SnpEff, is another popular open source variant annotation tool which has integrated within Galaxy and GATK. It is faster than the ANNOVAR and permits annotation of more genome versions. In addition, SnpEff is characterized by flexibility because supports the ability to add custom genomes and annotations, also, perform non-coding annotations. When SnpEff integrated into the GATK, it replaced the ANNOVAR program for variant analyses [65]. The Variant Annotation Tool (VAT) can be used as a command line interface or as a web application. VAT annotator tool it can annotate variants from multiple personal genomes at the transcript level providing statistical information across genes and individuals. Moreover, this tool provides to the user a significant set of choices, for example, visualizes the effects of different variants or integrates allele frequencies and genotype data from the underlying individuals. This set of choices make the VAT tool suitable for comparative analysis between different groups of individuals. The Variant Annotation Tool (VAT) provide a unique advantage to their users, they can use VAT as a virtual machine (VM) that can be run within a cloud-computing environment (including that operated by Amazon) to take advantage of the scalability and unlimited storage capacity offered by this framework [66]. In 2016 implemented a powerful tool set known as Ensembl Variant Effect Predictor which is open source and free to use. VEP applies annotation and prioritization in coding and non coding region; It can manage variants that came from large-scale sequencing projects or smaller analysis studies. The main different of the other tools is that provides full reproducibility of results.

Scientists compared VEP's runtime performance with Annovar and SnpEff which are the most widespread tools for variant annotation. SnpEff loads its entire annotation database into memory at start-up, unlike VEP, which loads the relevant genomic segments on demand; this accounts for VEP performing better than SnpEff on small datasets. Annovar, does not provide the same depth of annotation as VEP and so runs faster. [67] Unfortunately, with these tools is difficult to perform simply and flexible annotation with many different data sets that include the information for the annotation. Vcfanno [68] annotation tool, created to solve the above problem providing flexibility as concerns the extraction and the summarization of attributes from multiple annotation files. Then, merges the annotations within the INFO column of the original VCF file. Although, Vcfanno is a powerful annotation tool the users need to be familiar with the computer science because this annotation tool needs to construct a vcfanno configuration file to query in the files with the information for the annotation. Although there is a significant number of annotation tools, Annovar, SnpEff and Variant Effect Predictor (VEP) are widely used.

Below we will try to achieve a deeper comparison between these three tools. It is no surprise that these tools do not always agree since the way the rules have defined differ slightly between each application. To start with, each tool outputs its annotations in a different format.

Their mayor different is in the format of the output file. Annovar's output is a tab separated file, while SnpEff and VEP produce VCF files which use the "INFO" field to encode their annotations. In addition, the format of Annovar's rows changes depending on context while SnpEff and VEP represent data in a consistent format. These differences in the format of the context into the files create significant problems in the parsing of these files. For example, as concern the gene field, Annovar uses this field to provide distance information for all intergenic variants but for variants which labeled as splicing, it overloads the gene field with HGVS notion. Nowadays, with the use of the NGS is common to annotate thousands of variants at one time, so, it is really important to be able to parse the annotation file fast and reliable [69]. Except for the conventional annotation tools scientists have designed various tools such as automated variant annotation pipelines, web services for variant analysis or tools that use algorithms to predict possible impact of an amino acid substitution on the structure and function of a human protein. FamAnn [70], is a known pipeline which have used for targeting discovery for family-based sequencing studies. This tool is able to apply a different inheritance pattern or a de novo mutations discovery model to each family then select single nucleotide variants and small insertions and deletions segregating in each family or shared by multiple families. FamAnn has many advantages.

Firstly, is very easy to use because it only needs a Perl command line to generate the output Excel file with all the annotated files. Also, provides to the users the choice to apply various thresholds such as allele frequency cutoffs, directly on the output to prioritize variants. Moreover, faces the problem with the variants that hit multiple transcripts and hence may have different types of functional effects, FamAnn outputs all the effects for the same variant to avoid missing critical biological information. It provides functionalities offered by different bioinformatics resources, such as ENCODE annotation, frequency checking in public databases, pathogenicity prediction and conservation scores. FamAnn takes as input all types of sequencing data, in VCF format. Last but not least FamAnn pipeline support two annotators ; the snpEff annotator and the Variant Effect Predictor (VEP). Extasy is an other pipeline for ranking non synonymous single nucleotide variants that substantially improves prediction of disease-causing variants in exome sequencing data by integrating variant impact prediction, haploinsufficiency prediction and phenotype-specific gene prioritization. Someone can use Extasy on line or through command line. Unlike, Extasy require HPO/phenotypic terms which is problem in the case of diagnostic analysis of disorders where no such prior knowledge exists. Gemini (GEnome MINIng) is a flexible framework for exploring genome variation. It needs as input a VCF file (and an optional PED file) into the database. The first step is to automate annotation of all the variants by comparing them to several genome annotations from databases

such as OMIM, dbSNP. It uses SQLite database to store all the annotation information for every variant and allows users to explore and interpret both coding and non-coding variation using "off-the-shelf" tools or an enhanced SQL engine. Moreover supports reproducibility of the results. If we compare this tool with current tools such as VEP and snpEff we observe that Gemini provides better user friendly environment than other softwares. Other web services that offer variant annotation are MyGene.info and MyVariant.info which are high-performance web services for querying gene and variant annotation information. These web services besides they are public they have more than three million visitors per month. Also, they offer as service a cloud-based model for organizing and querying biological annotation information. The use of web service solves various problems for example the users do not need to write parsers for the files to process the files or the users do not need to store their data in local databases. Except for the fact that Web services make data management easier the mayor advantage is that the information is up to date. On the other hand, web services do not support big and complex queries in their data-base, due to network limitations [71].

In 2015 Matthias Arnold and his team tried to face the above problem by introducing SNiPA web service. SNiPA, is freely available to the scientific community offers variant-centered genome browsing and interactive visualization tools. SNiPA applies human genome annotation using the genome datasets from the Ensembl database and numerous variant-specific annotations taken from published datasets. Also, SNiPA contains annotations for all bi-allelic variants in phase 3 version 5 of the 1000 genomes project . The primary effect prediction of SNVs have calculated by the use of Ensembl VEP tool. SNiPA, has organize the complex information that provides to the users in a clear, comprehensive and informative structure extending effect categories contained in the Sequence Ontology. But when SNiPA web service is it updated? Because SNiPA takes the information that it needs for annotation by the Ensembl database,applies its update taking into account the Ensembl database updates [72].

2.1.4.4.2 Filtering methods of annotated VCF

Usually, scientists start to sequence the genome of an individual because they need to answer a biological question which associate with one or more hundreds variants. So, in this case filtering method which is a very important step is a complex exercise that bioinformaticians try to face it. Another important issue that scientists need to deal with is the choice of the appropriate filtering recipe. Regardless of the choice of the filtering strategy, the task remains the same: analysts try to decrease the false positive variants for their questions. At present, scientists open the annotated VCF file using a spreadsheet software package such as Microsoft Excel and then they apply one filter at the time. Below we describe the most common types of filters to answer the biological question. The special category of variant identification for clinical studies

demands special handling in the choice of filtering which we will analyze in the next subsection.

Inheritance and statistical filtering:

Scientist may have in mind an inheritance model for filtering. There are two main types of Mendelian inheritance, autosomal dominant and autosomal recessive model. In the first model the candidates are heterozygous variants, while unaffected individuals are homozygous for the reference allele. On the other hand, in the second type of inheritance pattern is usually the following: the affected individuals in a family are heterozygous for two different pathogenic mutations when parents are either non-consanguineous or less often homozygous. Unfortunately, the above filters can apply only in monogenic disorders. In addition, scientists should take into account parameters like incomplete penetrance or variable expressivity of the clinical phenotype or in the case of WES they should examine whether the causal variant is in the non-coding regions of the genome.

Variant class filtering:

Other filters that we can use in combination or not with the above filter are the genomic location and the variant class, which is in the "consequence" field of the VCF file. Variants in exonic missense, nonsense, stop-loss, frameshift and splice site regions probably affect protein function. Although, truncating and splicing mutations are important to cellular environment, additional consideration is necessary when progressing these as candidates for further evaluation. To eliminate the False Positives results scientists propose extracting reads spanning the genomic location of the variant, aligning them to the specific region in the reference genome using softwares like IGV.

Population frequency filtering:

SNPs with rare frequency in the population are expected to have a functional effect on the encoded protein. The annotators include the frequency of each variant by populations databases like ExAC (7 ethnicity groups + average) and 1000 Genomes project (5 ethnicity groups + average)

The scientific community recommends to prefer ExAC database due to its frequencies being calculated using more samples. Summarizing, it is important to note that is not always necessary to use all the above filters in a study. Moreover, each filter should be adjusted to the study requirements. [73].

2.1.4.4.3 Repositories for Variant storage and annotation

The information that we use to apply the thresholds to filter the variants have stored in specific databases. Below we describe the most important databases that we use their information to filter variants when we need to find an answer in a clinical associate question.

Single Nucleotide Polymorphism database (dbSNP) :

The need for a general catalog of genome variation to address the large-scale sampling designs required by association studies, gene mapping and evolutionary biology has led to the Single Nucleotide Polymorphism database. It was implemented by the National Center for Biotechnology Information (NCBI). DBSNP not only includes information for SNPs but also for microsatellites, and small-scale insertions and deletions along with publication, population frequency, molecular consequence, and genomic and RefSeq mapping information for both common variations and clinical mutations [74]. The main disadvantage is that dbSNP includes more false positive information in comparison with 1000 Genome because the data are not always accurate [75] .

Exome Aggregation Consortium (ExAC):

The Exome Aggregation Consortium (ExAC) is a group of scientists who collect and combine the exomes from global sequencing projects and make information available to the scientific community. Each variant is described by its functional annotation, and its frequency in global populations (African, American, Non-Finnish Europeans, Finnish Europeans, East Asians, South Asians). The user only needs to know for exploration to the web the ID, if he is searching for information for a specific variant, gene or region, or the chromosome position. This moment clinicians and scientists strongly recommend the scientific community to use only ExAC database which is the most well informed database (includes information from over 60,000 individuals) [76] .

Online Mendelian Inheritance in Man (OMIM) :

The founder of the Online Mendelian Inheritance in Man (OMIM) was Dr Victor A. McKusick and his vision was to create a repository for Mendelian Inheritance in Man. Nowadays, the National Center for Biotechnology Information supports OMIM database electronically. OMIM which is the largest repository of known pathogenic variants reported for humans is freely available and updated daily. This database focuses on the relationship between phenotype and

genotype containing information for all the known mendelian disorders and over 15.000 genes [77].

ClinVar Database:

ClinVar is one of the largest genomic databases in clinical and pathological analyses related to mutations. One of the main information that ClinVar provides is the clinical significance of the variants; the five categories that ClinVar support are recommended by the ACMG (benign, likely benign, uncertain significance, likely pathogenic and pathogenic). ClinVar, also, contains classifications of each variant as reported by clinical laboratories clinicians, expert groups, patients, researchers, and other databases maintained by the National Institutes of Health[78].

COSMIC Database:

COSMIC, the Catalogue Of Somatic Mutations In Cancer has designed to unite the scientific community with the world's information on somatic mutations in human cancer using a single system and make it easily explorable. One main benefit of COSMIC database is the manually curation of the genes using knowledge of both via publication and consortium data portals, delivers to deep mutation profiles on known cancer genes selected from the Cancer Gene Census [79].

PharmGKB Database:

The study of a human genome is not restricted only in variants that associate with disease, but also the analysts search for variant-drug association that enables the scientists to interpret efficacy of the drug in the presence of a particular variant. PharmGKB is the public database that provides information to the clinicians or researchers about the impact of genetic variation on drug response. In PharmGKB database the user can find curated knowledge about gene-drug associations and genotype-phenotype relationships [80].

2.1.4.5 Tools for Variant Prioritization

So, WES analysis pipeline requires the use of many different tools for different analysis steps and their complex parameters need to optimize for every run. As a consequence, the users should have advanced computing skills and a complete understanding of the tools that are

available on the web. We can manage the complexity of the exome sequencing analysis using software suites which provide all the tools required for each step in one package.

SIMPLEX is a pipeline that takes as input vcf files from Illumina and ABI SOLiD. It combines several applications to perform the following steps: initial quality control, intelligent data filtering and pre-processing, sequence alignment to a reference genome, SNP detection, functional annotation of variants using different approaches, and detailed report generation during various stages of the workflow. SIMPLEX provides tuning of a wide amount of parameters to "solve" the given biological problem and, at the same time, provides a set of standard parameters for naive users [81]. SeqMule is another pipeline user-friendly environment that uses Linux to manage data from whole exome or whole genome sequencing. It is suitable for both Mendelian disease study and tumor-normal paired somatic mutation analysis. SeqMule uses a variety of aligners (BWA-MEM, BWA-BACKTRACK, Bowtie, Bowtie2, SOAP2, SNAP) and annotators (GATK, SAMtools, VarScan, SOAPsnp, Freebayes). This platform is very flexible as it allows users to modify configuration file to fine tune its parameters [82]. Ingenuity Variant Analysis (IVA) offers the opportunity to users to search pathogenic and disease-causing variants in QIAGEN's Knowledge Base which is a repository of knowledge of millions of literature curated biological findings. Of course the user needs to have an active license for Ingenuity Variant Analysis [83]. An important feature that any platform should have for the genome exploration is the embedded genome browser. Users can interact better with the variants if they picture their genomic characteristics of them. Database.bio is a freely available web application that combines variant annotation, prioritization and visualization of variants. Variant information is presented in HTML pages that contain annotation details with a powerful embedded genome browser, allowing clinicians and researchers to carry out analysis of genomic sequencing data in a highly configurable manner. The Database.bio uses for variant annotation more than twenty-nine public databases. One main characteristic of this platform is that preprocess the data for annotation on a super computer and reduces database space because it uses a unified database representation with compressed fields. This tool has been designed under the guidelines of Health Insurance Portability and Accountability Act (HIPAA),which is a stringent standard for privacy and security in order to protect patient data which are retrieved and managed over the web [84].

DisGeNET is a free platform that collects information with two manners. Firstly, it gathers all the valuable information that associate with genotype-phenotype relationships from expert curated databases (such as UniProt, OMIM and ClinVar) and secondly, gathers information through NLP-based text-mining tools. One the other hand it offers several tools to interact with the data, including a web interface, a Cytoscape Ap or an R package [85].

2.1.4.6 Visualizing Human Genome Variation

"A picture is worth a thousand words."

The last decades there are groups of scientists that develop tools to successfully visualize the genome variation.

In biology there are different tools that visualize different information; for example tools for network biology (Cytoscape, Igraph etc) or tools for pathways (BiNA, KEGG Atla etc.) etc . Generally, in genomics field we meet four categories of visualization which are : assemblies viewers, genome alignment visualization tools, genome browsers and tools to directly compare different genomes with each other for efficient detection of SNPs and genomic variations. In this chapter, we will focus on the genome browsers category. Genome browser [86] provides a graphical interface for users to browse, search, retrieve and analyze genomic sequence and annotation data. JBrowse is a free, fast and full-featured genome browser built with JavaScript and HTML5. It is user friendly tool and users can easily embedded the JBrowse into websites or apps but can also be served as a standalone web page. JBrowse as tool can manage fast complex interactive queries. Analysis functions can readily be added using the plugin framework; JBrowse supports to browse local annotation files offline and to generate high-resolution figures for publication [87]. ChromoMap which developed using R programming language and it is not a web-based genome browser tool, but someone can install it as package in R [88]. The outputs of this genome are interactive and concern the visualization of entire chromosomes or chromosomal regions of any living organism can be used in publications. This tool offers some extra services like visualize polyploidy or create chromosome heatmaps.

One of the main challenges that the genome browsers have to face are the graphical challenges of displaying whole chromosomes within the available canvas space suitable for publications; these challenges arise due to the humongous genome size of certain species, including our own one. But how the chromoMap solves the main challenge of visualization? It renders chromosomes as a continuous composition of loci [89].

D3GB which is an interactive Web genome browser it can be found as an R package, a Python module or a WordPress plugin and users can integrate it in their pipeline. It provides web-oriented genome browser. It generates images in a format ready for publication. Moreover, It provides different forms to represent genes, sequence features, VEP annotated variants etc [90]. As we observe they are plenty of genome browsers and they have similarities with each other, but each one also has its unique uniqueness that not all of them share.

So, what is missing right now? It misses a global genome browser for all the scientific community that the scientists can use and build plugins that can be shared with other scientists, without the complexity of changing the core of the tool [91].

CHAPTER 3: PRECISION MEDICINE

3.1 Next generation sequencing in precision medicine

With the use of NGS methods (for example whole exome sequencing or multigene panels) the model of traditional medicine upgraded in a model which is called precision medicine. Precision medicine leads to a more exact diagnosis of human diseases and to the use of specific drugs for individual treatment. The new model of personalized medicine promises to reduce the healthcare cost on an overburdened and overwhelmed system.

3.1.2 Application of Next-Generation Sequencing in the Era of Precision Medicine

In this chapter we will mention some current applications of NGS in precision medicine. Firstly, doctors use the diagnostic sequencing to find rare variants associated with Mendelian disorders. Moreover, they use WES in cases where patients have failed to receive a diagnosis using multigene testing panel and workup. Recent studies [92] [93] have shown that WES provides a diagnostic rate ranging from 25% to 40%, which is two to three times higher than traditional genetic testing methods. Other publications [94] [95] have also proved that if we use WES as first test we will reduce not only the time for diagnosis but also the cost (in some cases WES costs half to one-quarter the cost of traditional testing) [96].

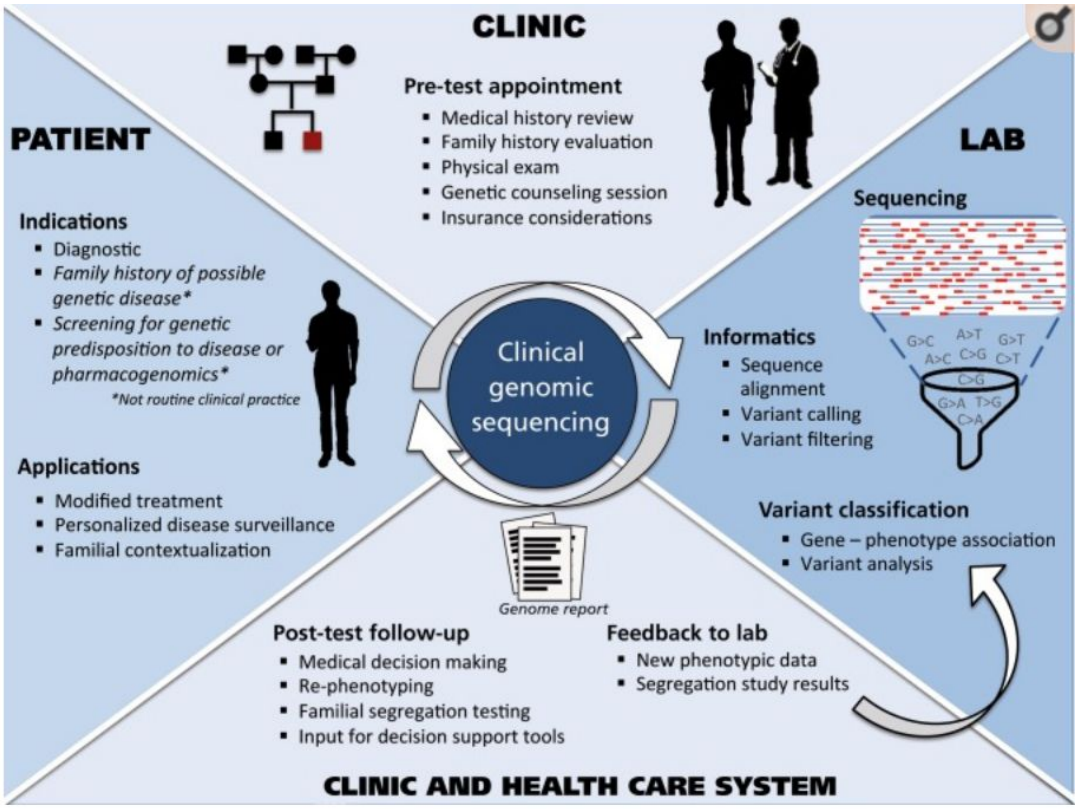


Figure 10: A schematic of the workflow of clinical genomic sequencing[8].

In precision medicine model despite of the individual genome sequencing important information from the family medical history will derive. Moreover, the doctor should apply systematically evaluation of the patient phenotype in databases for possible overlap with known syndromes. Also, doctors should inform the individuals that may learn about disease risks that may also affect their relatives. The whole pipeline of clinical sequencing is briefly described in figure 10 [97].

3.1.3 Genomic Data Processing in Clinically Actionable Variants

Having the scientist the raw data of the sample uses the known bioinformatic pipeline to create a processed vcf file. This pipeline consists of three steps: (1) Alignment, (2) variant calling, and (3) variant prioritization, these steps could vary based on the sequencing systems and the type of the capturing kits. Alignment step allows a number of quality control measures to determine with the most important being the average coverage depth of a genome; which is calculated as

[8] Source https://www.researchgate.net/publication/309631541_Genomic_sequencing_in_clinical_practice_Applications_challenges_and_opportunities

the number of bases of all short reads that match a genome divided by the length of this genome.

For exome sequencing in homozygous SNVs should be 100x (3x local depth) while for heterozygous SNVs 100x (13x local depth) [98], the percentage of all reads that map to a reference sequence and the percentage of reads that unmapped to a reference sequence. In the annotate VCF file we have ~ 60,000–100,000 genetic variants that can classified as pathogenic, benign or with uncertain significance (VUS) [99]. False positives variants may occur in this file due to mistakes in the sample preparation steps such as enrichment, instrument use. False positive variants are removed in the filtering process. In clinical studies, it is important to keep the variants which classified as Pathogenic Likely Pathogenic and Unknown Significance. If the biological question has relation with drug response of the individual then we keep also the variant with drug response information. Generally, the next step is to focus only on the variants that change an amin-acid or a splice site (non-synonymous variants). The third step is to apply the population frequency filter less than 1% to remove the polymorphisms of our samples, so our remaining list contains very rare variants that are either heterozygous or homozygous. Depending on the question, the analyst could apply more disease specific filters that may relate to the inheritance, gene expression and function in the remaining variants. Before the scientists proceed to the curation step, the analyst should check the remaining variants that seems to answer the biological question through the IGV tool for strand bias. With the term strand bias we mean that the genotype inferred from the positive strand and negative strand are significantly different, with one homozygous and the other heterozygous [100]. After the filtering step follows the variants curation based on the guidelines which are published in 2015 by the American College of Medical Genetics and Genomics and the Association for Molecular Pathology (ACMG/AMP). The role of ACMG/AMP guidelines is to provide in genetics community guidelines to classify variants as “pathogenic”, “likely pathogenic”, “uncertain significance”, “likely benign” or “benign” according to a series of criteria with levels of evidence defined as very strong, strong, moderate or supporting [101]. The classification system is based on eight parameters which are: population frequency data, genomic annotation and computational predictive data, functional data, segregation data, de novo data, allelic/genotypic data, public databases and literature and other data [102]. Scientists apply variant curation using the above described databases such as OMIM or COSMIC. As concerns the clinical care, one of the most important recommendations of ACMG for the laboratories that perform whole- exome sequencing or whole genome sequencing is to analyze the sequence of 56 specific genes. These genes were selected based on clinical evidence that pathogenic variants result in a high probability of severe disease that someone can prevent before symptoms occur. [103]. It is important to notice that while manual curation is the gold standard method for curation of variants, it is time consuming

In figure 11 briefly describes the common workflow that is used for exome sequencing analysis as well the quality metrics that in most cases are used to filter variants for each analysis step.

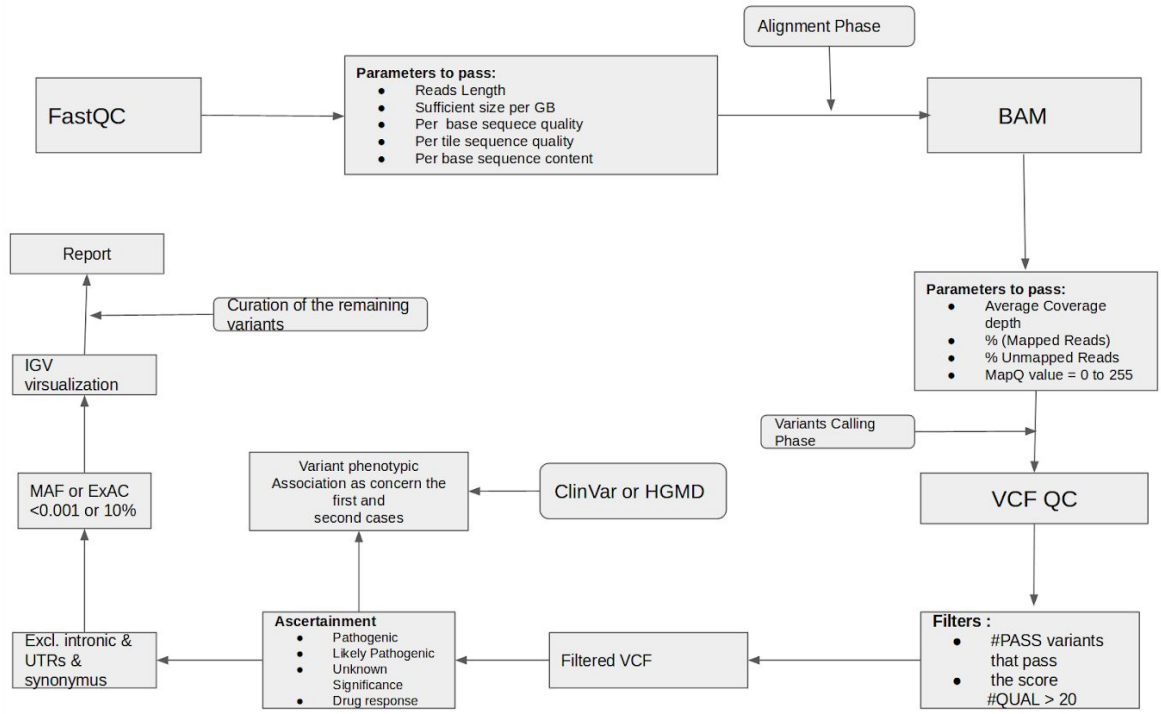


Figure 11: A schematic of the workflow of Whole Exome Sequencing Analysis .

The team that analyses a Whole Exome output after the priorization phase should contain various scientists like physicians, geneticists, bioinformaticians and other health professionals in order to connect clinical and laboratory findings with phenotype, family history and symptoms [104]. If more than one candidate variant is detected, the team must perform further evaluation(s) like sanger to determine which of the variant is causing the phenotype. Finally, in the case that no candidate variants are found should be mentioned in the report.

3.1.3.1 Tools for automated curation

MEMA [105] and MuteXt [106] are tools that use a dictionary search based on word distance metric to find the right protein or gene -mutation pairs extracted based on sentence co-occurrence. The algorithm calculates the distance as follows: A variant in a paper is closer to the names of its related genes/proteins rather than names of other proteins/genes that are also

present in the paper [103]. The main difference of MEMA and MuteXt tools is that the first uses regular expressions to extract mutations and mutation-gene pair while the other searches for mutation data using a pattern matching approach [153] Mutation GraB (Graph Bigram), is another tool that identifies, extracts, and verifies point mutations from biomedical bibliography. Mutation GraB uses a graph-based bigram traversal to identify these relevant associations and exploits the Swiss-Prot protein database to verify this information [105].

Tools that use text mining for automated curation are the below:

MutationMiner: MutationMiner system is the first open source solution that includes detection of mutation impacts, connecting them to their respective mutations and recognizing the affected protein properties, in particular kinetic and stability properties together with physical quantities [107]. MarkerInfoFinder: MarkerInfoFinder belongs to omicX tools, is an open-source, rule-based system for extracting point mutation mentions from text. MarkerInfoFinder is publicly available and there are versions of this tool in Python, perl and Java language [108]. In 2019 published Pathogenicity of Mutation Analyzer (PathoMAN) tool which is web American College of Medical Genetics and Genomics (ACMG) automated. It allows users to query single variants or upload a file with variants. A different approach of automated curation is the Machine learning method in curation research studies which have used to overcome manual curation processes. In 2014 Almeida et al. published a method for automated curation with the name mycoSORT [109]. This method uses support vector machine (SVM), Naïve Bayes and Logistic Model Trees to find relevant publications for curation for the mycoCLAP database. This method has some limitations, for example, it requires several text preprocessing steps and an extensive feature extraction process which are data/domain dependent. Kyubum Lee et al. proposed a machine learning method to assist Almerida's algorithm. He used the training dataset UniProtKB/Swiss-Prot and the NHGRI-EBI GWAS Catalog and then he used for training deep learning models based on convolutional neural networks. Then he used the trained models to classify and rank new publications for curation. In the end, he compared his method with the method proposed by Almeida et al. and conclude that the deep learning-based classifier performs better than traditional machine learning classifiers, even without feature engineering [110].

CHAPTER 4: DESCRIPTION OF ZAZZ PLATFORM

4.1 Client/Server architecture

If someone wants to think the Client/ Server model out of the box he can compare it as a restaurant where the waiter takes your order for a pizza, goes to the kitchen and comes back with the materials. You get to cook the pizza at your table and add your favorite components. Although, someone has a lot of work to do the advantages of this procedure are important. firstly, the service is much faster, the food is cooked exactly to your preferences, and the huge, expensive stove in the kitchen can be replaced by lots of cheap little grills. Now, under scientific terms, clients can send data requests to the server as queries (*query-shipping* systems) or as requests for specific data items (*data-shipping*) (see figure 12) [111].

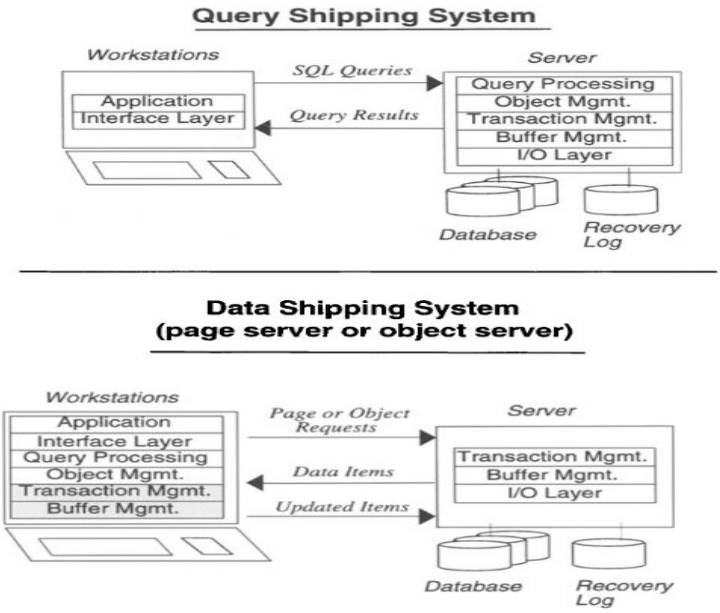


Figure 12: Different Client-Server models.

4.2 Information Visualization

Nowadays, we live in the era of the big data explosion, due to the advent of high-throughput genomics, biologists join this big-data club. Imagine that until 2012 the European Bioinformatics Institute (EBI) in Hinxton, UK had stored 20 petabytes (1 petabyte is 10^{15} bytes) of data and back-ups about biology information. Except for the big laboratories, even the "small" biology laboratories can produce a large amount of data due to the sequencers that exist in the market. In this point, raises an important question : How can we manage all this information quickly and efficiently?
The answer is hidden in the information visualization. But why the information visualization is so important? The main advantage is that help people to understand the hidden information.

4.2.1 Information Visualization via JavaScript

Nowadays, computer programmers use the web browser to store and visualize information. They use web browser due to the increased demand for real-time interactivity. People need to navigate, by clicking, scrolling etc fast and under a user friendly environment in their data. Because JavaScript is the language of web browser, it is used while visualizing the data.
There are plenty of JavaScript libraries that support a dynamic user friendly environment such as :

D3js

D3.js is the most popular JavaScript data visualization library. D3 uses HTML, SVG, and CSS for data management [112].

Raw graphs

Someone that needs to visualize spreadsheets he should use Stars Raw library. The format file that supports is tabular for example spreadsheets and it is based on the SVG format [112]

Metabase

Users can use Metabase library through a quick and simple way in order to create data dashboards without knowing SQL. Someone can create canonical segments and metrics, send data to Slack and more. [112]

DC.js

DC.js implemented by Nick Zhu; it is an open source JavaScript library based on Crossfilter and D3.js. We often use DC.js for multi-dimensional data visualization. D3 is used to render charts in CSS-friendly SVG format. When the filter or brush changes, all other charts are dynamically updated [113]. In this project we choose to work with DC.js library because it is considered a great library to make a dynamic dashboard. If someone uses once the crossfilter.js it is really easy to build lots of different graphs. It does not have to interact all the graphs together. In addition, doesn't need to handle the complexity of the data with every filter. So the amount of time saved is huge. This library also has the huge advantage of being really quick. For a dataset of 1MB, with 50 distinct dimensions and groups, changing simultaneously the filter of all those dimensions take less than 0.1 seconds.
Of Course it has some disadvantages but in this project we are able to face them. first of all, to use the library is important to learn how to deal with crossfilter.js. Yet, they have not implemented all the graphs and it is really difficult to add a graph not featured in dc.js. Moreover, it is not easy to mix dc.js with another graph library. Finally, each graph has a lot of different options and it is hard to find them because the documentation is not crystal clear [114].

4.3 Asynchronous JavaScript and XML (Ajax) search engine

Ajax is not a programming language, but a method of building interactive applications for the Web. It combines several programming tools including JavaScript, dynamic HTML (DHTML), Extensible Markup Language (XML), cascading style sheets (CSS), the Document Object Model (DOM), and the Microsoft object, XMLHttpRequest. It is worth mentioning that Ajax applications work with a Web browser and do not need installation. It is able to access the server both synchronously and asynchronously. With the term of synchronous we mean that the script stops and waits for the server to send back a reply before continuing. On the other hand, with the asynchronous method, we mean that the script allows the page to continue to be processed and handles the reply if and when it arrives. Processing your request asynchronously has as main advantage to avoid the delay while the retrieval from the server takes place because your visitor can continue to interact with the web page; the requested information will be processed in the background and the response will update the page as and when it arrives. The second method is very important especially in case of very large data where even if a response is delayed site may not realize it because they are active elsewhere on the page Ajax is part of our life the most known example of application that uses Ajax is the Google Maps. The applications that use Ajax are so widespread that even though a few years ago only Microsoft's

Internet Explorer browser support Ajax applications, currently, the most browsers support Ajax [115].

4.4 The web framework: Django

Another difficult task that we had to face in this thesis is the choice of the best web application framework. There are many of them on the market, and each of them has its peculiarities, strong and weak sides, as well as the best scopes of application. A correct choice of a framework can take off the processes, while a wrong decision can cost in the development of extra time, and budget. Some of the best tools for backend framework implementation are ExpressJs, Ruby on Rails, Spring etc. In table 1 is summarized the advantages and disadvantages of the best Backend frameworks [116].

Table 1: Summary of the best backend frameworks for web implementations

Backend Frameworks	Pros	Cons
Django	1. Scalable and flexible 2. Great for MVPs 3. Secure 4. Great documentation	1. Not the fastest 2. Monolithic
Expressjs	1. Simple 2. flexible 3. Packages for API development	1. Many callbacks 2. Unhelpful error messages 3. Not suitable for heavy apps
Ruby on Rails	1. Many tools and libraries 2. Fast development 3. Good for prototyping 4. Test automation	1. Slow boot time 2. Not the best choice for heavy applications 3. Lack of proper documentation
Spring	1. Great for java apps 2. Easy to cooperate with other programs 3. flexible	1. Difficult to learn 2. Can be unstable
Symfony	1. Fast Development 2. Reusable code 3. Great documentation	1. Comparatively slow

4.5 Architecture of Zazz Platform

The process of variant annotation and prioritization is one of the most diverse in clinical genetics. Although there is a plethora of genomic databases, each has different semantics and different applicability such as population, phenotype and sequencing protocol. Moreover,

existing annotation guidelines are too vague, with limited standardization that act more like a set of ad hoc generic directions than a clearly defined and directly applicable protocol. This has brought forward the need for a platform where researchers can simply define their own annotation schema with limited programming knowledge. Moreover, given the fuzziness of filtering criteria, a simple variant querying environment based on annotation values is simply not enough. The user needs to "explore" the data and not just query them. A querying mechanism requires a constant interaction between the UI (frontend) and the database (backend) which introduces a significant overhead on the exploration process. Even in the most sophisticated database designs, a complex query on a set of hundreds of thousands of variants is time-consuming enough to obstruct seamless data exploration. . A proper data exploration environment requires rich UI components with which the user can seamlessly interact by continuously applying multiple criteria with minimum delay. For this reason, and in contrary to similar environments, we make an important conceptual distinction which is also reflected on the UI. We split the environment in two parts. The first is a "traditional" online environment where the user applies queries with simple HTML forms, and the results are fetched from a database with a reasonable time-delay. We call this the "client-server" part. In this part the administrator can input every kind of data that he wants to manage (for example raw data from exome or whole genome sequence). In this thesis we chose to import annotate VCF files from different sources but for the same sample. Firstly we input the annotate file from the Ion Reporter platform (figure 13, case 1). The next scenario was to annotate the VCF file using the most widely used annotators which are VEP and ANNOVAR annotators (figure13, case 2). In the last scenario we built our custom clinVar Parser to gather the whole information from ClinVar for each variant. Then we used the VEP annotator as "platform" in order to annotate our VCF file with our ClinVar.vcf file (figure 13, case 3)

The second part of Zazz platform is a dynamic environment where the user can explore the data by interacting with a set of rich UI-components. On this part, filtering is happening in the browser and no client-server communication is required. The results from the filtering are shown almost immediately on the UI giving the sense of a zero-delay feedback interaction. This is possible with modern javascript libraries dc.js, crossfilter.js and d3.js. We call this the "dynamic" part. Data are fetched from the "client-server" part and fed to the "dynamic" part. Zazz is a "Single-page application", therefore both parts appear in a single web. page. The backend is built on the Django web framework whereas the frontend uses the AngularJS library.

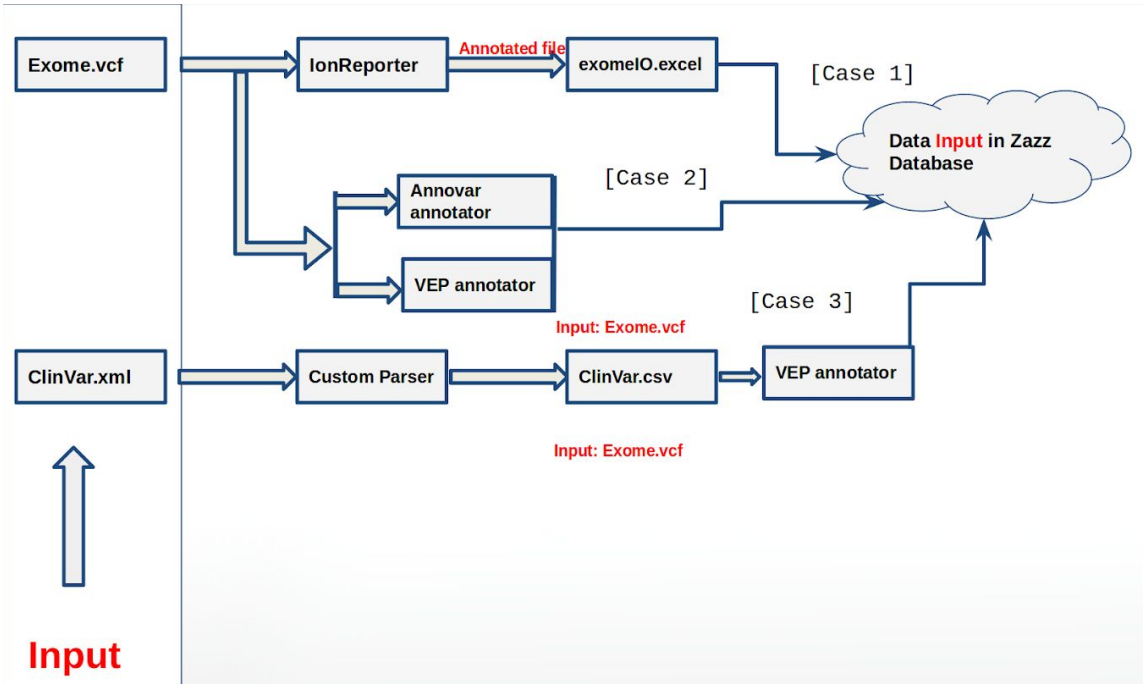


Figure 13: A schematic architectonic of the Zazz platform

4.5.1 Define the data

The primary concept in Zazz is the "field" which defines a distinct explorable column with annotation information. A user can define an arbitrary number of fields in a configuration file in JSON format. Each field has five main attributes: "data source", "item getter", "renderer", and "database" and "multi".

- **Data source:** In Zazz a data source can either be a file in CSV, Excel (Microsoft and OpenDocument) or HDF5 format. It can also accept data from online databases that have APIs that return data in JSON format. Finally users can import data programmatically through Pandas or with custom importers in python language.

- **Item getter:** This attribute defines how a field will be retrieved from a data source. It can contain quality control functions, type converters and any other post-processing actions in the python language. Item getters are functions that take as input a record of the data source and return the value of a field. For example a CSV file might contain the gene name of a variant as an ID of the HGNC (Human Genome Nomenclature Consortium). An item getter can be a function that converts this ID to the relevant gene name.

- **Renderer:** Renderers define which UI-components will be used for querying and exploring a field. Available values are: “categorical” which are rendered with dropdown menus supporting multiple selection, “piechart” which are rendered as pie-charts, “numerical” which are rendered as a slider in the “client-server” part and as a bar-charts on the “dynamic” part, and “freetext” which are rendered as text boxes allowing free text search

- **Database:** This attribute defines how the data will be stored internally on the database. Users can explicitly define the SQL parameters of the table column that will hold this field. On this attribute users can define SQL data types, maximum/minimum values or string lengths, default values and any other values are supported from the SQL table-column specification.

- **multi**: This attribute is described in the following paragraph.

4.5.1.1 Multiple fields

Most annotation fields contain singular values. Or else, a single value for a certain variant. For example, assume that an individual has the variant rs4925583. This variant has a global minor allele frequency (GMAF) of 0.25. This is a singular value meaning that each variant has only one value. Nevertheless, the same variant has two transcripts: NM_175911.2 and NM_001001963.1. On the first transcript, rs4925583, is an intronic variant whereas on the second it is an exonic variant that causes a missense variation. Moreover, this variant is present in 3 different Gene Ontology (GO) terms. Since this information regards a single variant, it is generated as a single record from a variant annotation software (Table 2). Yet, this form of annotation is unsuitable for querying since the information of which subfield corresponds to which is lost. For example, a query about transcripts that contain exonic variants will also return NM_175911.2 which is not correct.

Table 2: A typical record returned from an annotation tool.

dbSNP	GMAF	Transcripts	Location	Function	Gene Ontology
rs4925583	0.25	1 175911.2, NM 001001963.1	intronic, exonic	none, missense	GO:0005515, GO:0008150, GO:0003674

To resolve this issue, we need to correctly expand the fields containing multiple values and generate all possible records (Table 3).

Table 3: A typical record returned from an annotation tool.

dbSNP	GMAF	Transcripts	Location	Function	Gene Ontology
rs4925583	0.25	NM_175911.2, NM_001001963.1	intronic, exonic	none, missense	GO:0005515, GO:0008150, GO:0003674

In table 3 we observe that the two different values of Transcripts, Location and Function are coupled together. Also the same variant has three different Gene Ontology terms. Table 2 is unsuitable for querying. For example a search for the the transcript NM_175911.2 could return the value intronic, exonic for the location field which is misleading since the variant is only intronic for this transcript. Also, a search for exonic variants with the GO:0005516 term could also return the NM_175911.2 transcript which is not correct Internally, on the database level, we need to generate separate tables for each set of fields that contain multiple values and connect them with the main table with a "1 to N" relationship (Figure 14). When querying, we can apply a "CROSS JOIN" SQL function which generates the Cartesian product of two (or more) tables. This functionality can be configured with the "multi" attribute of a field. If left empty, the field is assumed to be a single valued field (like GMAF). Otherwise the user can define the name of the separate table that this field belongs as well as the special "item getter" that extracts and processes the sub-values of the field.

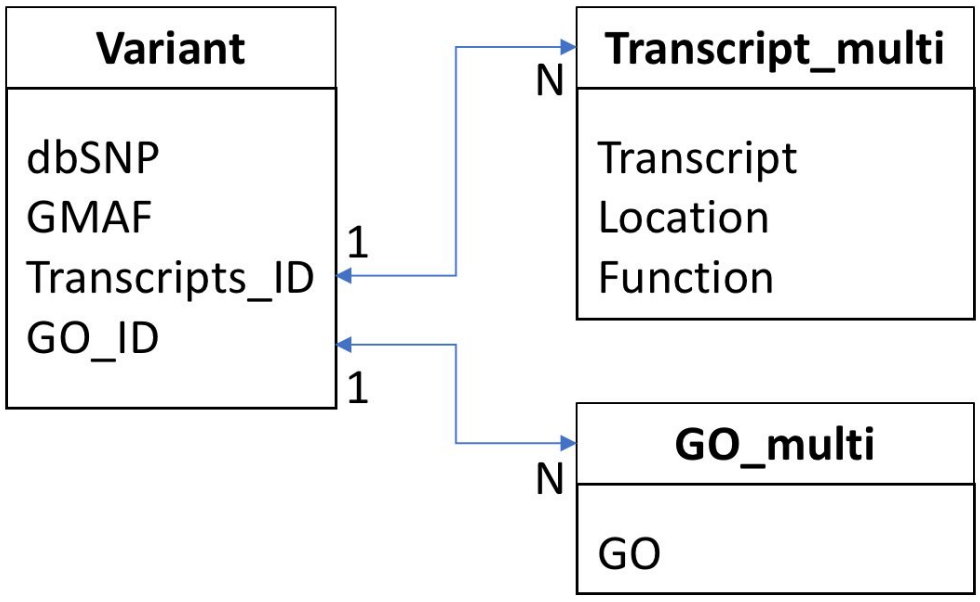


Figure 14: The database schema which allows multiple-value annotation for a single variant. By applying a "CROSS JOIN" SQL function on "Transcript_multi" and "GO_multi" tables, we can generate efficiently the contents of Table 3.

4.5.2 Genome Browser

Instead of using common UI-components to select a genomic region, in Zazz, we have integrated the D3GB genome browser [D3GB]. This browser contains a karyotype view of the genome where the user can select large chromosomal regions and also a zoomable chromosomal view for more detailed regions. Every variant that is not filtered-out from the user during exploration, is shown on the genome browser (Figure 15). Also, when the user selects a region from the browser, the rest of UI-components are automatically adjusted to portray this region.

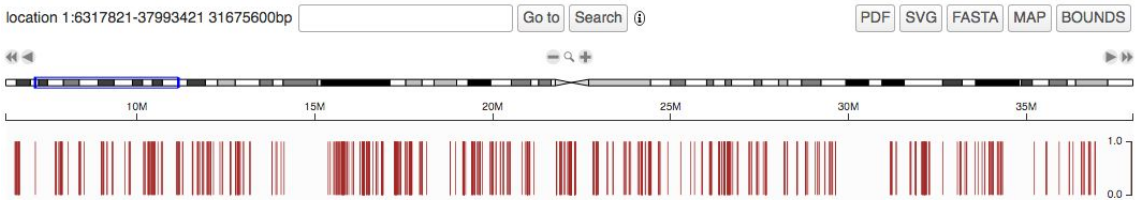


Figure 15: The D3GB genome browser integrated in Zazz. Red stripes indicate variants that pass the filters during exploration.

4.5.3 Importing Data

Once users create the JSON configuration file containing the field descriptions, they can run a python script that parses data inputs, creates the database, imports the data on the relevant tables and creates the frontend user interface. This script tries to minimize the time required for data import. To do that we have to minimize the number of SQL queries, in order to take advantage of the internal advanced methods of the database engine for massive data import. Initially the script separates “single” from “multi” fields. “single” fields include are stored in a unique table, which we call “main”, therefore we can use the “bulk import” feature of modern Database Engines. Single fields are split in chunks as large as the memory allows, and then added in bulk in the “main” table. Each chunk requires a single SQL query. The next step is to add the “multi” fields. Initially we are “bulk importing” into all relevant tables all multi fields. For example, all different Transcripts, Location and Function triplets are bulk imported into the “Transcripts_multi” table and all different Gene Ontology terms are bulk imported on the GO_multi table (Figure 13). During this process, we are storing in memory a data structure that links the primary key of the records in the “main” table with the set of primary keys of the records on “multi” tables. Then we are using this data structure to connect the primary keys of the “main” tables to the sets of records in the “multi” tables

Tables	main			Transcript_multi				GO_multi	
Fields	pk	dbSNP	GMAF	pk	Transcripts	Location	Function	pk	Gene Ontology
Records	1	rs4925583	0.25	1	NM_175911.2	intronic	none	1	GO:0005515
	1	rs4925583	0.25	1	NM_175911.2	intronic	none	2	GO:0008150
	1	rs4925583	0.25	1	NM_175911.2	intronic	none	3	GO:0003674
	1	rs4925583	0.25	2	NM_001001963.1	exonic	missense	1	GO:0005515
	1	rs4925583	0.25	2	NM_001001963.1	exonic	missense	2	GO:0008150
	1	rs4925583	0.25	2	NM_001001963.1	exonic	missense	3	GO:0003674
	2	rs1234567	0.10	3	NM_12345.1	exonic	missence	2	GO:0008150
Memory	pk:1			[1,2]				[1,2,3]	
	pk:2			[3]				[2]	

Figure 16: The upper part of this figure shows the expanded database records. Compared to Table 1, we have added another record (rs1234567). The lower part shows the data structure that stores the connections between the primary keys (pk) of the "main" table and the "multi" table. Notice that this is a compact representation of the complete table.

Django supports two mechanisms to make this connection. The first is to bulk import data in an intermediate table that controls the relationship between the "main" and the "multi" tables. The second is to add multiple records on each primary key of the main table. Our experiments show that the second method is slightly faster. To add the data shown in figure 16 we totally perform 7 SQL queries: 1 to add all data of table "main", 2 to add all data on the two "multi" tables and 4 to connect the two different primary keys of the "main" table with the two different set of values of the "multi" tables. Notice that if our dataset does not contain any "multi" field, then the complete data import takes 1 SQL query. Additionally, the complete design does not have any limit on the number of variants, samples or annotation fields.

4.5.3.1 Description of the existing IonTorrent annotated file

One of the most prominent sequencing platforms is Thermo-Ion Torrent. As with most sequencing platforms, sequencing is implemented in three stages. The first is the alignment of raw sequencing data (usually FASTQ) into a reference genome. This process generates aligned sequences in BAM format. The second stage contains three steps: (1) coverage analysis, which is the qualitative and quantitative analysis of the aligned regions (2) variant calling, which includes the detection of genetic regions that differ from the reference genome and (3) assessment of the quality metrics and statistics of the variant calling step. In the final, third

stage, of the sequencing workflow, all detected genetic variants for each sample are annotated through the Ion Reporter annotation software. This software fetches information from well-known genetic databases regarding the detected variants. In figure 16 we show the names of the databases and their respective versions for the Ion Reporter annotation tool.

Name	Version
5000Exomes	20161108
Canonical RefSeq Transcripts	v77
ClinVar	20170404
COSMIC	80
dbSNP	147
DGV	20160515
Disease Research Area	20170112
DrugBank	20161212
ExAC	031
Gene Ontology	20160928
Named Variants	1
OMIM	20161111
Pfam	30
PhyloP Scores	20160919
RefSeq Functional Canonical Transcripts Scores	6
RefSeq GeneModel	77

Figure 17: Names of the genetic databases and their versions, used for IonReporter v. 5.6

The final outcome of this process is an annotated VCF file that contains all detected variants along with the information fetched from the genomic databases. In table 4 we describe some of the annotated fields from the Ion Torrent output file that we used as the first input in Zazz database [see case 1 figure 13] having as example a variant which is located in chr1:16356501. In Appendix (table 1) we present analytically the fields, their type and their parameters.

Table 4: An example of the parameters used in Zazz platform in order to import annotated fields from Ion Torrent output file

Example Value	Description	Source	Django Field type	Django Field Parameters	Multi table	Formatter
chr1:16356501	Position of the variant.		Split in two fields: Chromosome --> CharField Position --> IntegerField	Chromosome --> {'max_length': '100'} Position --> {}	No	
SNV	Type of variant.	Ion Reporter™ Software Categories: SNV MNV CNV INDEL LONGDEL REF NOCALL FUSION EXPR_CONTROL RNA_HOTSPOT GENE_EXPRESSION RNAExonVariant ProcControl	CharField	{'max_length': '100'}	No	
G	The reference allele (hg19)	Ion Reporter™ Software	CharField	{'max_length': '255'}	No	
1	Size of the variant	Ion Reporter™ Software	IntegerField	{}	No	

4.5.3.2 Description of the existing annotated vcf files from VEP and Annovar

Variant classification is one of the most important phases in NGS analysis. As we read from previous chapters there are plenty of tools in the market that promise reliable Variant classification with the Annovar, SnpEff and Variant Effect Predictor (VEP) to be in the top of the most widely used annotation tools from the scientific community. Each tool fetches its annotation file output with information from the same databases but also fetch their annotation file output with extra information, so, for our platform which is an exploratory tool is important to be able to describe each variant with as much information we can. The first step was to

generate the input file for the two annotators. The input file which was generated via parsing from the existing Ion Reporter annotated file had the below information :

```
#CHROM  POS    ID    REF  ALT  QUAL FILTER  INFO
```

After we install the two annotators we used the below commands in order to generate the output annotated files:

For ANNOVAR

```
./table_annovar.pl /home/kantale/maria__input.vcf humandb/ -buildver hg19
-out koumakis -remove -protocol refGene,cytoBand,exac03,avsnp147,dbnsfp30a
-operation g,r,f,f,f -nastring . -vcfinput

INSTALL LATEST CLIVAR VERSION:
./annotate_variation.pl -buildver hg19 -downdb -webfrom annovar
clinvar_20190305 humandb/
```

For VEP

```
 time ./ensembl-vep-release-96/vep -i /home/kantale/koumakis_vep_input_2.vcf
--cache --dir_cache /home/kantale/VEP/homo_sapiens_vep_96_GRCh37/
--force_overwrite --offline --everything --fasta
/home/kantale/VEP/fasta/Homo_sapiens.GRCh37.dna.primary_assembly.fa.gz
--assembly GRCh37 --vcf -o output_2.vcf.
```

The last step is to input the annotated VCF files in Zazz platform. In table 5 and table 6 we describe some fields for each annotated VCF. In appendix A (table 5 and table 6) there is a detailed description for each field.

Table 5: Example of annotated fields from Annovar in Zazz platform

	Column	Examples	Description	Source	Django Field type	Django Field Parameters	Multi table	Formatter
	#locus	chr1:1558792						
1	ANN_ExonicFunc_refGene	nonsynonymous_SNV	Exonic variant function (e.g., nonsynonymous, synonymous)	ANNOVAR annotator.For more details: http://annovar.openbioinformatics.org/en/latest/user-guide/gene/	CharField	{'max_length': '200', 'null': 'True'}	No	

Table 6: Example of annotated fields from VEP in Zazz platform

	Column	Examples	Description	Source	Django Field type	Django Field Parameters	Multi table	Formatter
	#locus	chr1:1987993						
0	VEP_Allele	T/C	pair of alleles separated by a '/', with the reference allele first	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

4.5.3.3 ClinVar

Clinvar is a database that contains variants associated with a large range of diseases. Each variant can be associated with various types of conditions. For each variant-condition relationship, clinvar has a list of evidences named "assertion criteria". Assertion criteria contains information like publications, providers (for example institutes) and indications of the clinical significance of the variant for the development of a certain condition. From a computer science perspective, the main entry on this database is the variant. Each variant has a unique id which is called VCV (Variation ClinVar Record). For example the VCV record VCV000008678.2 refers to the SNP which is located to chromosome 11 at position 17388025 of the GRCh38 human assembly and encodes a nucleotide change from T to C. This variant is also present in the dbSNP database with the accession code: rs5219. This variant is associated with 14 conditions. For each condition ClinVar lists 5 main types of information:

1. A unique ID. This ID is called RCV (Reference ClinVar Record). Basically this ID "connects" a variant (RCV) with a condition. One VCV has many RCVs, but one RCV has one (one only one) VCV.
2. The name of the condition (for example Permanent neonatal diabetes mellitus). Each condition is linked with known phenotype databases (for example MedGen [117]. The interpretation of the condition. Clinvar uses 14 different representations of the clinical significance of a variant-condtion connection these are Benign, Likely benign, Uncertain significance, Likely pathogenic,

Pathogenic, drug response, association, risk factor, protective, Affects, conflicting data from submitters, other, not provided and '-'. The later is used in special conditions when a variant is submitted in combination with another variant and the interpretation value is missing. More information on the semantics of each "interpretation" value exists here:

3. The number of assertion criteria
4. The date when the association between the variant and condition was last evaluated.

As we mentioned before each variant-condition connection contains many "assertion criteria". For example one of the 14 conditions with which the variant VCV000008678.2 is associated is Permanent neonatal diabetes mellitus. This unique relationship is encoded with the RCV code: RCV000020356.2. This RCV is associated with 2 assertion criteria. Each assertion criteria has a unique ID which is called SCV (Submitted record in ClinVar). On[9]e RCV has many SCV and one SCV has one (and only one) RCV. For example the two SCVs that the RCV RCV000020356.2 contains are: SCV000040740 and SCV000483236. The information that ClinVar stores for each assertion criteria is:

1. The unique ID (or else the SCV) of the assertion criteria
2. The name of the submitter (usually a research facility)
3. The review status of the assertion. In Clinvar, each assertion, undergoes a review process. This process assigns a status that indicates the validity of this assertion. Moreover each status has a star-grading system that ranges from 0 to 4. 0 stars is for assertion titled "no assertion provided", "no assertion criteria provided", "no assertion for the individual variant". 1 star is for assertions titled "criteria provided, single submitter", "criteria provided, conflicting interpretations". 2 stars is for assertions titled "criteria provided, multiple submitters, no conflicts". 3 stars is for assertions titled "reviewed by expert panel" and 4 stars is for "practice guideline". For more information on this grading system see:[10]
4. The clinical significance of this assertion. This is the same as the "interpretation" field in RCV records.
5. The origin studied in this assertion. For example whether the mutation maternal inherited, paternal inherited, somatic or germline. More information here:[11]
6. The collection method with which this assertion was studies (or else how the samples were collected). For example, if this is a case-control study, a in-vitro experiment or a literature study. More information: [12] In brief, clinvar contains

[9] https://www.ncbi.nlm.nih.gov/clinvar/docs/clinsig/

[10] https://www.ncbi.nlm.nih.gov/clinvar/docs/review_status/

[11] https://www.ncbi.nlm.nih.gov/clinvar/docs/properties/

[12] https://www.ncbi.nlm.nih.gov/clinvar/docs/spreadsheet/

Variants (VSC). Each variant has many conditions (RCV) and each Variant-Conditon relationship has many assertions (SCV). Also each SCV belongs to a unique RCV and each RCV belongs to a unique VCV. More information regarding the IDs of ClinVar is available here: [13] ClinVar is community based database. Overall approximately 1000 different submitters have uploaded 680,000 variant-condition relationships for 430.000 unique variants [118]

4.5.3.3.1 Tools and Environments for ClinVar annotation and filtering

ClinVar is the defacto database for variant prioritization and annotation in clinical genetics. Nevertheless the web service that supports it from NCBI, offers only simple browsing and basic querying. This is one of the points for improvement that have been brought forward from ClinVar's development team itself [118]. For this purpose various tools and services have been developed in order to expand its functionality and augment the user-experience. Simple Clinvar [119] offers a google-like single-user-input field for querying the complete dataset. It also visualizes the query results and offers aggregate statistics. Clinvar Miner [120] is another service that offers simple filtering and data exploration capabilities. Both Simple Clinvar and Clinvar Miner suffer from the "submit" problem whereas they do not offer variant annotation. The only tool that also offers annotation is Clinotator [121].

4.5.3.3.2 Releases of ClinVar

ClinVar updates constantly these relationships and makes complete releases once every month. More info on ClinVar releases is available here [14]. These releases are in two different formats: VCF and XML. Releases in VCF contain only variants that also exist in dbSNP and have an rs# identifier. Also VCF contain only information regarding the VCV-RCV relationship and does not contain information regarding the assertion criteria. This might be problematic since information that lies in the assertion criteria can give insights regarding the pathogenicity of a variant. For example the VCF entry for the aforementioned variant (rs5219), assigns a unique clinical significance value, which is drug_response. Yet if we browse the web version of ClinVar for this variant (https://www.ncbi.nlm.nih.gov/clinvar/variation/8678/\#id_second) we will notice that there are conditions (Diabetes mellitus type 2) for which this variant is a risk factor. The VCF version contains a minimal and simplified version of the complete data model of ClinVar [206]. Yet this version is far easier to parse than the XML version which contains the complete data. This has an undesirable effect: Annotation software like VEP and

[13] https://www.ncbi.nlm.nih.gov/clinvar/docs/identifiers/#accessions

[14] https://www.ncbi.nlm.nih.gov/clinvar/docs/ftp_primer/

ANNOVAR provide only the ClinVar fields that exist in the VCF version. Indeed, adding a VCF file as an annotation track is trivial for both software. For this reason, the ClinVar fields that are included by default by VEP and ANNOVAR are incomplete for some clinical genetics tasks.

4.5.3.3.3 Parsing ClinVar XML

In order to get advantage of the complete data structure of ClinVar, we decided to build our own XML parser. This task has two main inherent difficulties. The first is that the XML schema does not follow the simple data model of ClinVar. Specifically, as we have described, ClinVar structures its data according to VCV-RCV-SCV relationship. Or else, each VCV (variant) has many RCV (conditions), and each RCV has many SCV (assertion criteria). Surprisingly, ClinVar's XML schema, places RCV on the top. Each RCV has the unique VCV which is associated followed by the list of SCV. For example, if we assume that one VCV has two RCVs (RCV1 and RCV2) and the first RCV has 2 SCVs (SCV1 and SCV2) and the second has three (SCV3, SCV4, SCV5), the XML would have the following structure:

```
<DATA>
  <RCV 1>
    <VCV></VCV>
    <SCV 1></SCV 1>
    <SCV 2></SCV 2>
  </RCV 1>
  <RCV 2>
    <VCV></VCV>
    <SCV 3></SCV 3>
    <SCV 4></SCV 4>
    <SCV 5></SCV 5>
  </RCV 2>
</DATA>
```

Notice in this schema that the same VCV is repeated twice in different RCVs. An expected and more intuitive schema would be:

```
<DATA>
```

```
  <VCV>
    <RCV 1>
      <SCV 1></SCV 1>
      <SCV 2></SCV 2>
    </RCV 1>
    </RCV 2>
      <SCV 3></SCV 3>
      <SCV 4></SCV 4>
      <SCV 5></SCV 5>
    </RCV 2>
  </VCV>
</DATA>
```

A second difficulty has to do with the size of the data itself. The latest XML release of ClinVar is a 830 Megabytes compressed file. This size makes prohibitive the use of DOM (Document Object Model) parsers. A DOM parser loads the complete XML tree in memory and provides an API for traversing the tree and accessing each element along with its attributes. This is very convenient since the programmer does not have to keep track of the relative position of an element in the tree. An alternative is to use a SAX (Simple Api for XML) parser. A SAX parser performs a serialized parsing of the XML file. Namely it calls a user-defined function for each element that reads from the file. In essence it performs a top-down depth-first parsing of the tree and calls a user defined function for each element. This function provides the name of the element, its attributes, and the event which triggered this call (element start or element end). SAX parsing do not keep track of the path of a node in the tree. An obvious advantage of this strategy is that the parsing has minimal memory footprint even for very large XML files. The main disadvantage is that the programmer should build custom methods in order to monitor the position of an element. In our implementation we followed a hybrid approach. Overall we parse the document with a SAX parser. For every RCV element, we build a custom DOM-like structure that contains the tree for this element. Then we extract the RCV, VCV and SCV data and we save them in a CSV file. When an RCV element ends, we destroy the tree, releasing thus, its memory and we continue with the next RCV element.The result of this processing is a CSV file where each VCV (variant) can exist multiple times due to the inconsistency of the XML schema presented above. Therefore this file is unsuitable for variant annotation. To

overcome this, we re-order the file according to the VCV column and we group rows with the save VCV together. This is done through the pandas framework which is very efficient with tabular data.

4.5.3.3.4 Annotating an exome with custom ClinVar data

The end result of the parsing procedure, is a CSV file that can be fed to any variant annotation software. The annotation tracks of this file contain a far more fine-grained and detailed version of ClinVar. For example both ANNOVAR [15] and VEP [16] offer this functionality. ANNOVAR requires a file with space separated values and VEP requires a file that can be in BED, GFF, GTF, VCF or BIGWIG format. In both tools, the end result is an annotated VCF file.

4.5.3.3.4 Importing custom ClinVar data to Zazz

The last step in this part of the analysis is to import the exome sequencing data annotated with our custom ClinVar parser into Zazz (figure 13 scenario 3). The overall purpose is to take advantage of the complete VCV-RCV-SCV abstraction. Zazz supports fields with multiple values, therefore since a VCV can have multiple RCV-SCV tuples, we define a "multi" field which encodes the VCV-RCV-SCV relationship. A disadvantage of this approach is that we have to expand all combinations of RCV-SCV for each variant. In a future version of Zazz we plan to support multi-fields of more than 2 dimensions (a multi field having multi fields, etc). In figure 18 we present a schematic representation of this abstraction.

---

15 http://annovar.openbioinformatics.org/en/latest/user-guide/filter/#generic-mutation-annotations

16 https://www.ensembl.org/info/docs/tools/vep/script/vep_custom.html

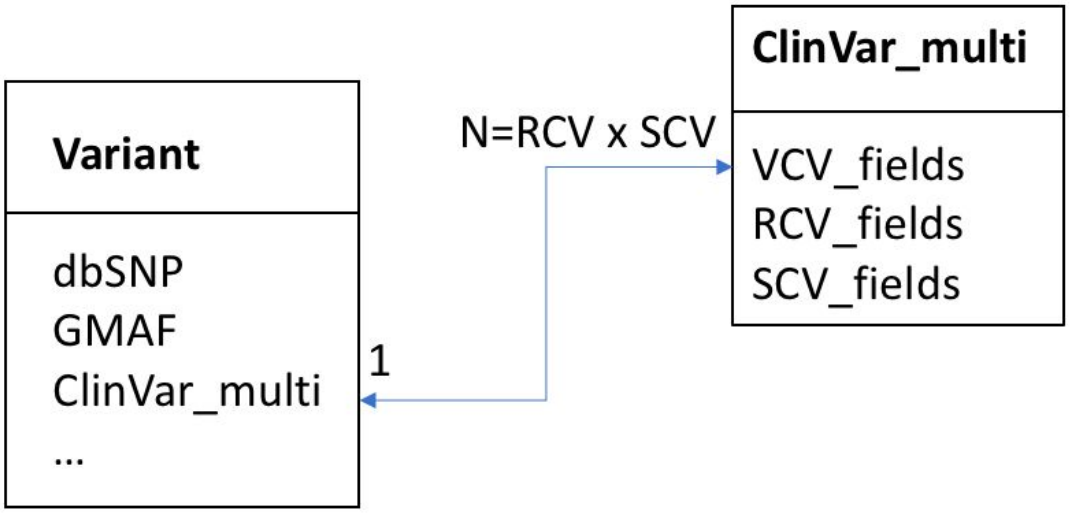


Figure 18 : A single variant contains a primary key to a table describing a VCV-RCV-SCV triplet.

4.5.4 *The User Interface*

The user interface is a single page that is split in two main parts. The first (client-server) is placed on the top of the page and it contains a list of all fields. Each field contains the UI component (i.e. select menu, slider) that was defined for querying. When a user makes any change on these components then Zazz performs a query to the database and returns only the number of variants that pass the filters. It also fetches the selected fields for only the first 1000 variants that passed the filters which are shown in a table. In a production environment we assume that the database contains hundreds of millions of variants, each having hundreds of fields. A generic query (i.e chromosome=1) that would fetch the complete set of results, would easily clutter the browser of the user. Even if we use paging to show small frames of the data, the user would still be overwhelmed from the enormous size of returned data. With this design, users have a good sense of the size and the dimensionality of the data returned from the set of filters and can adjust the queries accordingly without wondering whether or not their browser has enough resources to show the results. On this part of the interface, Zazz can query terabytes of data with the same efficiency, regardless of the abilities of the user's browsing device. Once users have selected an "interesting" subset of data through this initial filtering, they can press the "Explore" button. Then, the complete set of variants that passed the filters are fetched on the browser. These variants do not contain the complete set of fields but only the ones that the user selected. It is known that usually a clinical geneticist will explore variants based on 5 to 20 features that they consider "useful". In contrast, variant annotation software like VEP and ANNOVAR can easily fetch 50 fields of annotation information each. Exploring data on an interesting set of features can streamline easier the whole procedure. Nevertheless users can anytime select (or de-select) features from the "upper" part and update the resulting data.

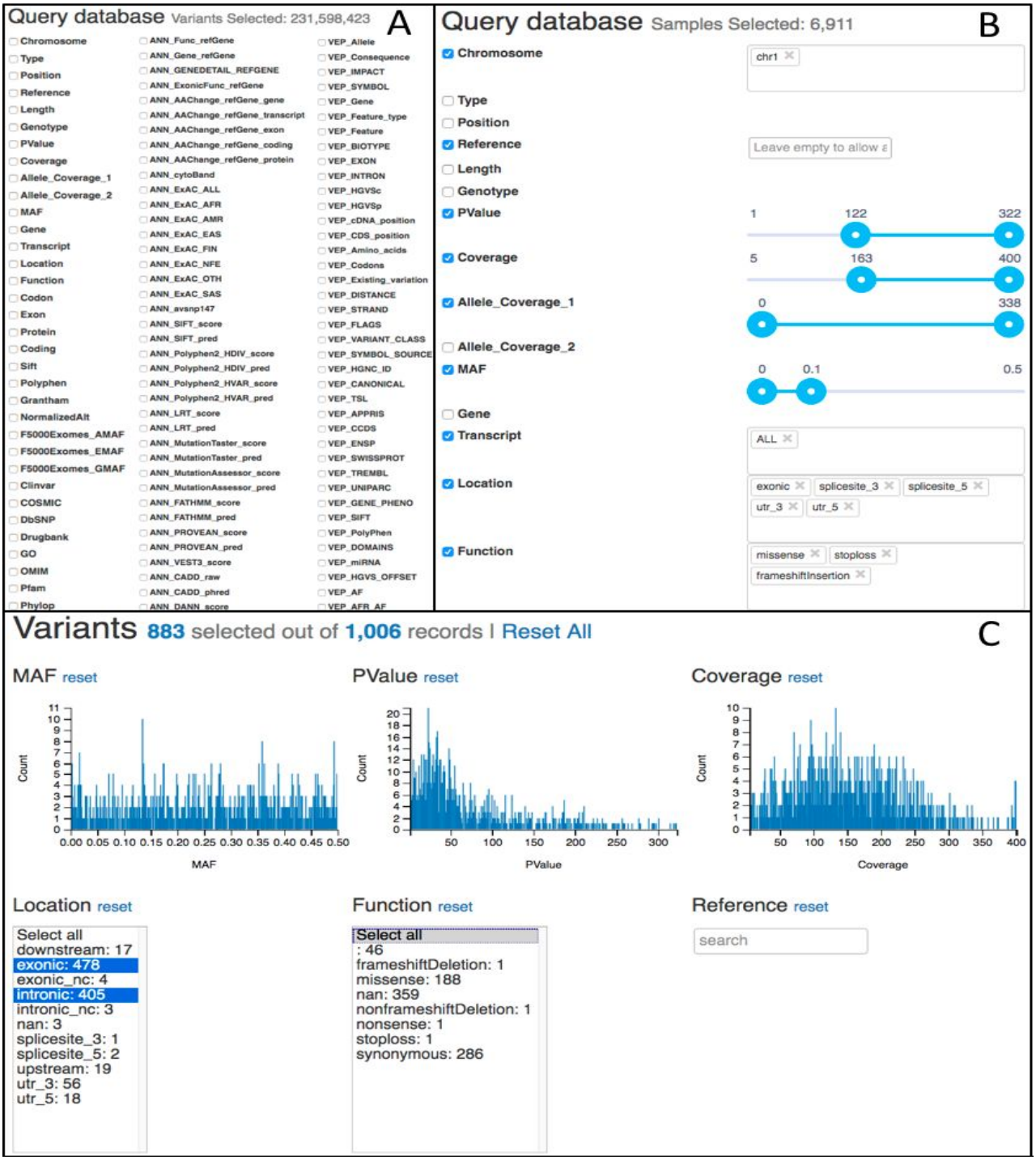


Figure 19: The main components of the interface of Zazz. Upper left (A) shows the complete list of fields in a configuration that contains all annotations from Ion Reporter Software, VEP and ANNOVAR (total 158 fields). Upper right (B) shows filters applied in the "client-server" part and the lower (C) subfigure shows the interactive components for data exploration through the dc.js library.

Once the “Explore” button is pressed the data are fed to the “dynamic” part which is placed lower on the interface. On this part, we use the dc.js javascript library to explore the data. Again users select which fields they want to “experiment” with and for each one an interactive chart is created. Charts on dc.js are interconnected. A filtering interaction in any chart updates the data representation in all the other. This part has a default dc.js components which is a table showing all variants that pass the current filters. We have also created a custom chart which updates the location of the filtered variants in a D3GB genome browser.

CHAPTER 5: CONCLUSION

Sequencing only the coding regions not only helps to discover variants that contribute in many mendelian diseases that would not have been otherwise possible but also helps the doctors to treat diseases with the appropriate medication for the patient. Although, there are several bioinformatic tools to help in the accurate identification, and interpretation of disease causing variants in exome sequencing experiments, as we get more into effects of genetic variants, this information will increase in complexity and size. This has as a result the need to build environments that will help clinical genetics to efficiently explore the variant annotation information to apply the appropriate ACMG reporting guidelines. Zazz is an environment set to fulfil this need. The main disruptive feature of Zazz is that it offers a meaningful balance between two design patterns, each with its pros and cons, that similar tools follow. The first pattern is the "client-server" model. On that model the complete information is stored in a large database and the user has to continuously submit complex and time-consuming queries in order to properly prioritize a list of genetic variants. The benefit of this model is that it has no constraints on the amount of information that can be queried. The disadvantage is that the client-server communication can introduce a delay which makes the whole process unsettling and non-interactive. The second model is the "dynamic" where the complete information is stored on the users' browser and the exploration is performed through interactive UI components. Modern browsers are very efficient in interactive data analysis and visualization even when presented with millions of data rows. Yet the size of annotated variants from NGS experiments is prohibitive for interactive exploration even when the user has unusual computational resources available. Zazz offers both environments in an online single page application. At present, Zazz is placed on one of the most sensitive places of modern NGS pipelines. This is right after variant annotation and right before reporting. For now, Zazz platform is the only free tool that supports quickly dynamic- user friendly exploration upon over 150 columns in order to provide a more complete variant exploration to the users. In the future we plan to offer docker containers that will contain besides Zazz, pre-installed variant annotation software (like VEP and ANNOVAR), annotation files from external databases (like ClinVar and Gencode) and python scripts that will automatically annotate a VCF file and add the results in Zazz.

BIBLIOGRAPHY

1. https://byjus.com/biology/mendelian-disorders/
2. https://ghr.nlm.nih.gov/primer/basics/dna
3. Galich, Nikolay E. "Invariance and noises of Shannon entropy for information on oxidative activity of DNA in all living cells for medical diagnostics." *American Journal of Operations Research* 4.02 (2014): 72.
4. Gonzaga-Jauregui, Claudia, James R. Lupski, and Richard A. Gibbs. "Human genome sequencing in health and disease." *Annual review of medicine* 63 (2012): 35-61.
5. Richards, Sue, et al. "Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology." *Genetics in medicine* 17.5 (2015): 405.
6. 1000 Genomes Project Consortium. "A global reference for human genetic variation." *Nature* 526.7571 (2015): 68.
7. Venter, J. Craig, et al. "The sequence of the human genome." *science* 291.5507 (2001): 1304-1351.
8. International Human Genome Sequencing Consortium. "Initial sequencing and analysis of the human genome." *nature* 409.6822 (2001): 860.
9. Allen, Hana Lango, et al. "Hundreds of variants clustered in genomic loci and biological pathways affect human height." *Nature* 467.7317 (2010): 832.
10. Robert, Francis, and Jerry Pelletier. "Exploring the Impact of Single-Nucleotide Polymorphisms on Translation." *Frontiers in genetics* 9 (2018).
11. Redon, Richard, et al. "Global variation in copy number in the human genome." *nature* 444.7118 (2006): 444.
12. Cheng, Ze, et al. "A genome-wide comparison of recent chimpanzee and human segmental duplications." *Nature* 437.7055 (2005): 88.
13. https://www.nlm.nih.gov/visibleproofs/education/dna/vntr.pdf
14. Waddington, Conrad H. "The epigenotype." *International journal of epidemiology* 41.1 (2011): 10-13.
15. Waddington, Conrad Hal. "Towards a theoretical biology." *Nature* 218.5141 (1968): 525.
16. Morris, J. R. "Genes, genetics, and epigenetics: a correspondence." *Science* 293.5532 (2001): 1103-1105.
17. Reik, Wolf. "Stability and flexibility of epigenetic gene regulation in mammalian development." *Nature* 447.7143 (2007): 425.

18. Siva, Nayanah. "1000 Genomes project." (2008): 256.
19. International HapMap Consortium. "The international HapMap project." *Nature* 426.6968 (2003): 789.
20. Sherry, Stephen T., et al. "dbSNP: the NCBI database of genetic variation." *Nucleic acids research* 29.1 (2001): 308-311.
21. Johannsen, Wilhelm. "The genotype conception of heredity." *The American Naturalist* 45.531 (1911): 129-159
22. Ilona Miko (2008). Gregor Mendel's principles of inheritance form the cornerstone of modern genetics. So just what are they? Gregor Mendel and the principles of inheritance. Nature Education 1(1):134
23. Chial, Heidi. "Rare genetic disorders: learning about genetic disease through gene mapping, SNPs, and microarray data." *Nature education* 1.1 (2008): 192.
24. Sanger, Frederick, Steven Nicklen, and Alan R. Coulson. "DNA sequencing with chain-terminating inhibitors." *Proceedings of the national academy of sciences* 74.12 (1977): 5463-5467. from Cambridge University awarded a Nobel Prize in Chemistry in 1980.
25. Maxam, Allan M., and Walter Gilbert. "A new method for sequencing DNA." *Proceedings of the National Academy of Sciences* 74.2 (1977): 560-564. from Harvard University.
26. Kchouk, Mehdi, Jean-François Gibrat, and Mourad Elloumi. "Generations of sequencing technologies: From first to next generation." *Biology and Medicine* 9.3 (2017).
27. Shendure, Jay, et al. "DNA sequencing at 40: past, present and future." *Nature* 550.7676 (2017): 345.
28. Totomoch-Serra, Armando, Manlio F. Marquez, and David E. Cervantes-Barragán. "Sanger sequencing as a first-line approach for molecular diagnosis of Andersen-Tawil syndrome." *F1000Research* 6 (2017).
29. AliciaGomesMS (2018). Chapter 5 - Genetic Testing Techniques. Pediatric Cancer Genetics 2018, Pages 47-64.
30. WANG, Xing-chun, et al. "High-throughput sequencing technology and its application." *China Biotechnology* 32.01 (2012): 109-114.
31. https://allseq.com/knowledge-bank/sequencing-platforms/ion-torrent/
32. Baker, Monya. "De novo genome assembly: what every biologist should know." (2012): 333.
33. Ng PC, Kirkness EF. Whole genome sequencing. Methods MolBiol 2010;628:215-26.
34. Wang, Zhong, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics." *Nature reviews genetics* 10.1 (2009): 57.

35. Settings M. Metagenomics versus Moore's law. Nat Methods 2009;6:623.
36. Kahvejian, Avak, John Quackenbush, and John F. Thompson. "What would you do if you could sequence everything?." *Nature biotechnology* 26.10 (2008): 1125.
37. Visscher, Peter M., et al. "10 years of GWAS discovery: biology, function, and translation." *The American Journal of Human Genetics* 101.1 (2017): 5-22.
38. Clarke, Geraldine M., et al. "Basic statistical analysis in genetic case-control studies." *Nature protocols* 6.2 (2011): 121.
39. Klein, Robert J., et al. "Complement factor H polymorphism in age-related macular degeneration." *Science* 308.5720 (2005): 385-389.
40. Ambrosone, Christine B. "The promise and limitations of genome-wide association studies to elucidate the causes of breast cancer." *Breast Cancer Research* 9.6 (2007): 114.
41. Reich, David E., and Eric S. Lander. "On the allelic spectrum of human disease." *TRENDS in Genetics* 17.9 (2001): 502-510.
42. Jorgenson, Eric, and John S. Witte. "Coverage and power in genomewide association studies."*The American Journal of Human Genetics* 78.5 (2006): 884-888.
43. Rabbani, Bahareh, Mustafa Tekin, and Nejat Mahdieh. "The promise of whole-exome sequencing in medical genetics." *Journal of human genetics* 59.1 (2014): 5.
44. Pabinger, Stephan, et al. "A survey of tools for variant analysis of next-generation genome sequencing data." *Briefings in bioinformatics* 15.2 (2014): 256-278.
45. https://www.ecseq.com/support/ngs/what-is-the-best-ngs-alignment-software
46. Anju Ramesh Ekre. 2016."Genome sequence alignment tools: A review". 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB).
47. Li, Ruiqiang, et al. "SOAP: short oligonucleotide alignment program." *Bioinformatics* 24.5 (2008): 713-714.
48. Jiang, Hui, and Wing Hung Wong. "SeqMap: mapping massive amount of oligonucleotides to the genome." *Bioinformatics* 24.20 (2008): 2395-2396.
49. Langmead, Ben, et al. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome biology* 10.3 (2009): R25.
50. Li, Heng, and Richard Durbin. "Fast and accurate short read alignment with Burrows–Wheeler transform." *bioinformatics* 25.14 (2009): 1754-1760.

51. Liu, Yongchao, Bertil Schmidt, and Douglas L. Maskell. "CUSHAW: a CUDA compatible short read aligner to large genomes based on the Burrows–Wheeler transform." *Bioinformatics* 28.14 (2012): 1830-1837.
52. Sandmann, Sarah, et al. "Evaluating variant calling tools for non-matched next-generation sequencing data." *Scientific reports* 7 (2017): 43169.
53. Larson, David E., et al. "SomaticSniper: identification of somatic point mutations in whole genome sequencing data." *Bioinformatics* 28.3 (2011): 311-317.
54. Wang, Weixin, et al. "FaSD-somatic: a fast and accurate somatic SNV detection algorithm for cancer genome sequencing data." *Bioinformatics* 30.17 (2014): 2498-2500.
55. Roth, Andrew, et al. "JointSNVMix: a probabilistic model for accurate detection of somatic mutations in normal/tumour paired next-generation sequencing data." *Bioinformatics* 28.7 (2012): 907-913.
56. Kim, Sangwoo, et al. "Virmid: accurate detection of somatic mutations with sample impurity inference." *Genome biology* 14.8 (2013): R90.
57. Liu, Yongchao, et al. "SNVSniffer: an integrated caller for germline and somatic single-nucleotide and indel mutations." *BMC systems biology* 10.2 (2016): 47.
58. Christoforides, Alexis, et al. "Identification of somatic mutations in cancer through Bayesian-based analysis of sequenced genome pairs." *BMC genomics* 14.1 (2013): 302.
59. Saunders, Christopher T., et al. "Strelka: accurate somatic small-variant calling from sequenced tumor–normal sample pairs." *Bioinformatics* 28.14 (2012): 1811-1817.
60. Saunders, Christopher T., et al. "Strelka: accurate somatic small-variant calling from sequenced tumor–normal sample pairs." *Bioinformatics* 28.14 (2012): 1811-1817.
61. Shiraishi, Yuichi, et al. "An empirical Bayesian framework for somatic mutation detection from cancer genome sequencing data." *Nucleic acids research* 41.7 (2013): e89-e89.
62. Cibulskis, Kristian, et al. "Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples." *Nature biotechnology* 31.3 (2013): 213.
63. Wang, Kai, Mingyao Li, and Hakon Hakonarson. "ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data." *Nucleic acids research* 38.16 (2010): e164-e164.
64. Hui Yang et al. (2015). Genomic variant annotation and prioritization with ANNOVAR and wANNOVAR.*Nature Protocols* volume 10, pages 1556–1566.
65. Cingolani, Pablo, et al. "A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3." *Fly* 6.2 (2012): 80-92.

66. Habegger, Lukas, et al. "VAT: a computational framework to functionally annotate variants in personal genomes within a cloud-computing environment." *Bioinformatics* 28.17 (2012): 2267-2269.
67. McLaren, William, et al. "The ensembl variant effect predictor." *Genome biology* 17.1 (2016): 122.
68. Pedersen, Brent S., Ryan M. Layer, and Aaron R. Quinlan. "Vcfanno: fast, flexible annotation of genetic variants." *Genome biology* 17.1 (2016): 118.
69. http://blog.goldenhelix.com/goldenadmin/the-sate-of-variant-annotation-a-comparison-of-annovar-snpeff-and-vep/
70. Yao, Jianchao, et al. "FamAnn: an automated variant annotation pipeline to facilitate target discovery for family-based sequencing studies." *Bioinformatics* 30.8 (2014): 1175-1176.
71. Xin, Jiwen, et al. "High-performance web services for querying gene and variant annotation." *Genome biology* 17.1 (2016): 91.
72. Sefid Dashti, Mahjoubeh Jalali, and Junaid Gamieldien. "A practical guide to filtering and prioritizing genetic variants." *Biotechniques* 62.1 (2017): 18-30.
73. Arnold, Matthias, et al. "SNiPA: an interactive, genetic variant-centered annotation browser." *Bioinformatics* 31.8 (2014): 1334-1336.
74. Sherry, Stephen T., et al. "dbSNP: the NCBI database of genetic variation." *Nucleic acids research* 29.1 (2001): 308-311.
75. Musumeci, Lucia, et al. "Single nucleotide differences (SNDs) in the dbSNP database may lead to errors in genotyping and haplotyping studies." *Human mutation* 31.1 (2010): 67-73.
76. https://wp.sanger.ac.uk/barrettgroup/2015/03/20/exac-its-big-and-easy-to-use/.
77. Hamosh, Ada, et al. "Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders." *Nucleic acids research* 33.suppl_1 (2005): D514-D517.
78. Landrum, Melissa J., et al. "ClinVar: public archive of interpretations of clinically relevant variants." *Nucleic acids research* 44.D1 (2015): D862-D868.
79. Forbes, Simon A., et al. "COSMIC: exploring the world's knowledge of somatic mutations in human cancer." *Nucleic acids research* 43.D1 (2014): D805-D811.
80. Thorn, Caroline F., Teri E. Klein, and Russ B. Altman. "PharmGKB: the pharmacogenomics knowledge base." *Pharmacogenomics*. Humana Press, Totowa, NJ, 2013. 311-320.
81. Fischer, Maria, et al. "SIMPLEX: cloud-enabled pipeline for the comprehensive analysis of exome sequencing data." *PloS one* 7.8 (2012): e41948.
82. Guo, Yunfei, et al. "SeqMule: automated pipeline for analysis of human exome/genome sequencing data." *Scientific reports* 5 (2015): 14283.

83. https://www.qiagenbioinformatics.com/products/ingenuity-variant-analysis/ .
84. database.bio: a web application for interpreting human variations.
85. Piñero, Janet, et al. "DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants." *Nucleic acids research* (2016): gkw943.
86. https://genome.ucsc.edu/
87. Buels, Robert, et al. "JBrowse: a dynamic web platform for genome visualization and analysis." *Genome biology* 17.1 (2016): 66.
88. Team, R. Core. "R: A language and environment for statistical computing." (2013): 201.
89. Anand, Lakshay. "chromoMap: An R package for Interactive Visualization and Annotation of Chromosomes." *bioRxiv* (2019): 605600.
90. http://d3gb.usal.es/index.html.
91. https://medium.com/@Marianattestad/one-genome-browser-to-rule-them-all-cc41e2daccd7.
92. Farwell, Kelly D., et al. "Enhanced utility of family-centered diagnostic exome sequencing with inheritance model–based analysis: results from 500 unselected families with undiagnosed genetic conditions." *Genetics in Medicine* 17.7 (2015): 578.
93. Iglesias, Alejandro, et al. "The usefulness of whole-exome sequencing in routine clinical practice." *Genetics in Medicine* 16.12 (2014): 922.
94. Joshi, Charuta, et al. "Reducing the cost of the diagnostic odyssey in early onset epileptic encephalopathies." *BioMed research international* 2016 (2016).
95. Iglesias, Alejandro, et al. "The usefulness of whole-exome sequencing in routine clinical practice." *Genetics in Medicine* 16.12 (2014): 922.
96. Niguidula, Nancy, et al. "Clinical whole‐exome sequencing results impact medical management." *Molecular genetics & genomic medicine* 6.6 (2018): 1068-1078.
97. Krier, Joel B., Sarah S. Kalia, and Robert C. Green. "Genomic sequencing in clinical practice: applications, challenges, and opportunities." *Dialogues in clinical neuroscience* 18.3 (2016): 299.
98. https://genohub.com/recommended-sequencing-coverage-by-application/
99. MacArthur, D. G., et al. "Guidelines for investigating causality of sequence variants in human disease." *Nature* 508.7497 (2014): 469.
100. Strom, Samuel P. "Current practices and guidelines for clinical next-generation sequencing oncology testing." *Cancer biology & medicine* 13.1 (2016): 3.
101. Ravichandran, Vignesh, et al. "Toward automation of germline variant curation in clinical cancer genetics." *Genetics in Medicine* (2019): 1.

102. http://www.acgs.uk.com/media/1140458/uk_practice_guidelines_for_variant_classification_2018_v1.0.pdf
103. Pandey, Kapil Raj, et al. "The curation of genetic variants: difficulties and possible solutions." *Genomics, proteomics & bioinformatics* 10.6 (2012): 317-325.
104. Jamuar, Saumya Shekhar, and Ene-Choo Tan. "Clinical application of next-generation sequencing for Mendelian diseases." *Human genomics* 9.1 (2015): 10.
105. Lee, Lawrence C., Florence Horn, and Fred E. Cohen. "Automatic extraction of protein point mutations using a graph bigram association." *PLoS computational biology* 3.2 (2007): e16.
106. Horn, Florence, Anthony L. Lau, and Fred E. Cohen. "Automated extraction of mutation data from the literature: application of MuteXt to G protein-coupled receptors and nuclear hormone receptors." *Bioinformatics* 20.4 (2004): 557-568.
107. http://www.semanticsoftware.info/open-mutation-miner-project
108. https://omictools.com/mutationfinder-tool
109. Almeida, Hayda, et al. "Machine learning for biomedical literature triage." *PLoS One* 9.12 (2014): e115892.
110. Lee, Kyubum, et al. "Scaling up data curation using deep learning: an application to literature triage in genomic variation resources." *PLoS computational biology* 14.8 (2018): e1006390.
111. Lee, Kyubum, et al. "Scaling up data curation using deep learning: an application to literature triage in genomic variation resources." *PLoS computational biology* 14.8 (2018): e1006390.
112. https://blog.bitsrc.io/11-javascript-charts-and-data-visualization-libraries-for-2018-f01a283a5727
113. https://dc-js.github.io/dc.js/.
114. https://medium.com/@louisn_23157/interactive-dashboard-crossfilter-dcjs-tutorial-7f3a3ea584c2
115. https://www.thoughtco.com/use-asynchronous-or-synchronous-ajax-2037228
116. https://gearheart.io/blog/top-10-web-development-frameworks-2019-2020/
117. Halavi, Maryam, et al. "MedGen." *The NCBI Handbook [Internet]. 2nd edition*. National Center for Biotechnology Information (US), 2018.
118. Landrum, Melissa J., and Brandi L. Kattman. "Clinvar at five years: delivering on the promise." *Human mutation* 39.11 (2018): 1623-1630.
119. Pérez-Palma, Eduardo, et al. "Simple ClinVar: an interactive web server to explore and retrieve gene and disease variants aggregated in ClinVar database." *Nucleic acids research* (2019).

120. Henrie, Alex, et al. "ClinVar Miner: Demonstrating utility of a Web‐based tool for viewing and filtering ClinVar data." *Human mutation* 39.8 (2018): 1051-1060.
121. Butler III, Robert R., and Pablo V. Gejman. "Clinotator: analyzing ClinVar variation reports to prioritize reclassification efforts." *F1000Research* 7 (2018).

APPENDIX A

Table 1: An analytic description of how Zazz manages the fields of an annotated file from Ion Reporter.
'Column' is the name of the field in annotated excel (or VCF) file, or else how IonReporter names each field. 'Example Value' is an example value for this field. 'Description' contains a description of the semantics of the field. 'Source' contains the database or the technique with which IonReporter acquired the value of the field. 'Django Field type' is the type of field according to Django. A complete list of Django Field Types is available here: https://docs.djangoproject.com/en/2.2/ref/models/fields/ . These fields are in essence an abstraction over SQL field types. 'Django Field Parameters' are the parameters passed to the Django field type. Usually these are: "max_length" (maximum string length for varchar SQL fields), 'null' whether this value takes null values. In Zazz we use null values (None in python) to indicate empty fields. 'Multi table' indicates whether this field is a 'multi' field. If yes, it contains the name of the table that contains the multiple values of this field. In Zazz implementation we create a Many-to-Many relationship from the main table to the 'multi' table. 'Formatter' contains the python function that converts the values seen in the excel (or VCF) to the ones stored in the database. For example the pvalue field contains p-value probabilities. In the database we store the -log(x) of these values which are easier to filter and visualize. Note: Some columns in the excel file are stored in more than one field in Zazz. For example the field with Column name '# locus' it contains values like 'chr1:16356501'. From this column we extract the values for two fields: Chromosome (i.e. 'chr1') and Position (i.e. 16456501).

Table 1:

	Column	Example Value	Description	Source	Django Field type	Django Field Parameters	Multi table	Formatter
0	# locus	chr1:16356501	Position of the variant.		Split in two fields: Chromosome --> CharField Position --> IntegerField	Chromosome --> {'max_length': '100'} Position --> {}	No	
1	type	SNV	Type of variant.	Ion Reporter™ Software Categories: SNV MNV CNV INDEL LONGDEL REF NOCALL FUSION EXPR_CONTROL RNA_HOTSPOT GENE_EXPRESSION RNAExonVariant ProcControl	CharField	{'max_length': '100'}	No	
2	ref	G	The reference allele (hg19)	Ion Reporter™ Software	CharField	{'max_length': '255'}	No	
3	length	1	Size of the variant	Ion Reporter™ Software	IntegerField	{}	No	
4	genotype	G/A	Genotype of the sample in each position	Ion Reporter™ Software	CharField	{'max_length': '255'}	No	

5	pvalue	4.75E-41	*p-value of the variant call: logarithmic transformation of the made by the VariantCaller. For example, a VariantCaller quality score of 20 is associated with a p-value of 0.01. A 30 is associated with a p-value is 0.001.*	Ion Reporter™ Software Threshold: 0.00001 to 0.99999	IntegerField	{'null': True}	No	-LOG10(x)
6	coverage	236	Total coverage for a variant	Ion Reporter™ Software	IntegerField	{}	No	
7	allele_coverage	129,107	Number of reads supporting the called allele	Ion Reporter™ Software	Split in two fields: allele_coverage_1 --> IntegerField allele_coverage_2 --> IntegerField	allele_coverage_1 --> {'null': True} allele_coverage_2 --> {'null': True}	No	re.split(r'\,', x)
8	maf	0.367	Population frequency information from the 1000 genomes project. MAF numbers are provided by the dbSNP in Ion Reporter™ Software, which gets the MAF numbers from 1000 genomes. Therefore, the version of dbSNP annotation sources used within the Ion Reporter™ analysis may impact these MAF values.	1000 Genomes Range: 0.0-0.5	FloatField	{'null': True}	No	Sometimes this field contains 2 or more values (i.e. 0.306:0.469). It is unclear why this happens. In our implementation in this case we take the first
9	gene	CLCNKA	Set of genes that overlap with the variant	https://www.ncbi.nlm.nih.gov/refseq/ http://www.ensembl.org/index.html	CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
10	transcript	NM_004070.3	Preferred transcripts used to determine coding regions of genes. If you include a transcript file, only transcripts that are present in your selection of canonical transcripts are reported. Other transcripts are filtered out.	RefSeq canonical; Ensembl canonical Format: GENE_NAME transcript_accession_id1 , transcript_accession_id2 , ...	CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
11	location	exonic	Position of the variant.	Ion Reporter™ Software Categories: Intronic Exonic UTR_3, UTR_5 Splice_5 Splice_3	CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')

				Upstream Downstream Exonic_nc				
12	function	missense	The effect of the variant on the coding sequence.	Ion Reporter™ Software Categories: RefAllele Unknown Synonymous Missense NonframeshiftInsertion NonframeshiftDeletion nonframeshiftBlockSubstitution frameshiftinsertion frameshiftDeletion nonsense stoploss	CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
13	codon	ACC	HGVS notation that represents an amino acid change	Ion Reporter™ Software	CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
14	exon	14			CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
15	protein	p.Ala447Thr			CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
16	coding	c.1339G>A	HGVS notation that represents a nucleotide change	Ion Reporter™ Software	CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
17	sift	0.14	Prediction of the functional effect of a variant on a protein	JCVI http://provean.jcvi.org/index.php Threshold: 0.0-0.05 deleterious 0.05-1.0 benign	FloatField	{'null': 'True'}	Transcripts	x.split('|')
18	polyphen	0.002	Prediction of the functional effect of a variant on a protein	Harvard University http://genetics.bwh.harvard.edu/pph2/ Threshold: 0.0-0.15 benign 0.15-1.0 possibly damaging 0.85-1.0 damaging	FloatField	{'null': True}	Transcripts	x.split('|')

19	grantham	58	A measure of evolutionary distance. A lower Grantham score reflects less evolutionary distance. A higher Grantham score reflects a greater evolutionary distance. Higher Grantham scores are considered more deleterious	Ion Reporter™ Software Threshold: 5-215	FloatField	{'null': True}	Transcripts	x.split('|')
20	normalizedAlt	A			CharField	{'max_length': '200', 'null': 'True'}	Transcripts	x.split('|')
21	5000Exomes	AMAF=0.2873:EMAF=0.4557:GMAF=0.3987	Population frequency information from the 5000 exomes project	NHLBI ESP https://evs.gs.washington.edu/EVS/ Threshold: 0.0-0.5	Split in 3 fields: F5000Exomes_AMAF --> FloatField F5000Exomes_EMAF --> FloatField F5000Exomes_GMAF --> FloatField	F5000Exomes_AMAF --> {'null': 'True'} F5000Exomes_EMAF --> {'null': 'True'} F5000Exomes_GMAF --> {'null': 'True'}	No	f5000Exomes_AMAF --> x.split('|')[0].split('=')[1] F5000Exomes_EMAF --> x.split('|')[1].split('=')[1] F5000Exomes_GMAF --> x.split('|')[2].split('=')[1]
22	NamedVariants	M470V	A list of known variants in the CFTR gene panel	Ion Reporter™ Software	CharField	{'max_length': '200', 'null': 'True'}	No	
23	clinvar	non-pathogenic	Assessment of the impact of the variant observed from NCBI ClinVar database	ClinVar https://www.ncbi.nlm.nih.gov/clinvar/ Categories: Pathogenic Likely pathogenic Benign Likely benign Variant of Uncertain Significance (VUS) Drug response	CharField	{'max_length': '200', 'null': 'True'}	No	
24	cosmic	1344642:1344643	Catalog of somatic mutations in tumor tissue	COSMIC https://cancer.sanger.ac.uk/cosmic/	CharField	{'max_length': '200', 'null': 'True'}	COSMIC	x.split(':')

25	dbsnp	rs201746872:rs3748596	Single Nucleotide Polymorphism database. The dbSNP annotation source in Ion Reporter™ Software contains a flag carried by a subset of its SNPs that have been curated by UCSC to be "UCSC Common SNPs". In order for a variant to be annotated as a UCSC Common SNP, the variant is first annotated as being present in dbSNP, and it might also then have be classified as a UCSC Common SNP.	dbSNP https://www.ncbi.nlm.nih.gov/snp/	CharField	{'max_length': '200', 'null': 'True'}	DBSNP	x.split(':')
26	dgv		Database of Genomic Variants: A curated database of human genomic structural variation	http://dgv.tcag.ca/dgv/app/home	This field was empty for all variants			
27	drugbank	Niflumic Acid	List of drugs known to target the gene(s) affected by the variant	DrugBank https://www.drugbank.ca/ Note: When you create an hg19 annotation set, do not use the annotation source DrugBank version 20150107. Use instead annotation source DrugBank version 1 or DrugBank version 20161212 or 20180731, which the latest version available in Ion Reporter™ Software 5.12 for hg19. If you use the DrugBank version 20150107 in an hg19 annotation set, you will not be able to create a filter chain of DrugBank for any analysis that uses the annotation set.	CharField	{'max_length': '200', 'null': 'True'}	DRUGBANK	x.split(':')
28	GO	GO:0005737:GO:0016787:GO:0005634	Standardized ontology for gene and gene products. For example, functional role or localization.	GO http://geneontology.org/	CharField	{'max_length': '200', 'null': 'True'}	GO	x.split(':')

29	omim	614282:615291	Online Mendelian Inheritance in Man®	OMIM https://www.ncbi.nlm.nih.gov/omim	CharField	{'max_length': '200', 'null': 'True'}	OMIM	x.split(':')
30	pfam	PF00536:PF07647	Protein domain families in the coded protein	Pfam https://pfam.sanger.ac.uk/ Category: Pfam-A: "A is curated and contains well-characterized protein domain families with high quality alignments, which are maintained by using manually checked seed alignments and HMMs to find and align all members"	CharField	{'max_length': '200', 'null': 'True'}	PFAM	x.split(':')
31	phylop	-1.04,-0.31	Measure of conservation of the protein across a wide range of organisms	Cornell University http://compgen.cshl.edu/phast/ Threshold: -20 to 30 Positive scores — Measure conservation, which is slower evolution than expected, at sites that are predicted to be conserved. Negative scores — Measure acceleration, which is faster evolution than expected, at sites that are predicted to be fast-evolving.	FloatField	{'null': 'True'}	PHUYLOP	x.split(',')

Table 2: An analytic description how Zazz manages the annotated fields from a file which is output of Annovar [1] [2] [3]. The description of each column is the same as in Table 1.

	Column	Examples	Description	Source	Django Field type	Django Field Parameters	Multi table	Formatter

[1] http://www.programmersought.com/article/5214984688/
[2] https://brb.nci.nih.gov/seqtools/colexpanno.html
[3] https://github.com/WGLab/doc-ANNOVAR/blob/master/docs/user-guide/filter.md

	#locus	chr1:1558792	Chr1: 10689678						
0	ANN_Func_refGene	exonic	exonic	Annotate the region where the variant site is located (exonic, splicing, UTR5, UTR3, intronic, ncRNA_exonic, ncRNA_intronic, ncRNA_UTR3, ncRNA_UTR5, ncRNA_splicing, upstream, downstream, intergenic)	ANNOVAR annotator. For more Details : http://annovar.openbioinformatics.org/en/latest/user-guide/gene/	CharField	{'max_length': '200', 'null': 'True'}	No	
1	ANN_Gene_refGene	MIB2	PEX14	List the transcripts associated with this variant site (only transcripts that function in accordance with the Func column are listed). If Func is intergenic, the gene names on both sides are listed here	ANNOVAR annotator.	CharField	{'max_length': '200', 'null': 'True'}	No	
2	ANN_GENEDETAIL_REFGENE	NaN	NaN	Describe the variation in UTR, splicing, ncRNA_splicing, or intergenic regions. When the value of the Func column is exonic, ncRNA_exonic, intronic, ncRNA_intronic, upstream, downstream, upstream; downstream, ncRNA_UTR3, ncRNA_UTR5, the column is empty; when the value of the Func column is intergenic, the column format is dist=1366; dist =22344, indicating the distance of the variant site from the genes on both sides	ANNOVAR annotator.	CharField	{'max_length': '500', 'null': 'True'}	ANN_GeneDetail_refGene	x.replace('\\x3d', '=').split('\\x3b')
3	ANN_ExonicFunc_refGene	nonsynonymous_SNV	synonymous_SNV	Exonic variant function (e.g., nonsynonymous, synonymous)	ANNOVAR annotator.For more details: http://annovar.openbioinformatics.org/en/latest/user-guide/gene/	CharField	{'max_length': '200', 'null': 'True'}	No	
4	ANN_AAChange_refGene_gene	MIB2	PEX14:NM_004565:exon9:c.G768A:p.V256V	Amino acid change; gene name region	ANNOVAR annotator.	CharField	{'max_length': '100', 'null': 'True'}	ANN_AAChange_refGene	
5	ANN_AAChange_refGene_transcript	NM_001170686	PEX14:NM_004565:exon9:c.G768A:p.V256V	Amino acid change:Known RefSeq accession	ANNOVAR annotator.	CharField	{'max_length': '100', 'null': 'True'}	ANN_AAChange_refGene	
6	ANN_AAChange_refGene_exon	exon3	PEX14:NM_004565:exon9:c.G768A:p.V256V	Amino acid change:region	ANNOVAR annotator.	CharField	{'max_length': '100', 'null': 'True'}	ANN_AAChange_refGene	

7	ANN_AAChange_refGene_coding	c.T305C	PEX14:NM_004565:exon9:c.G768A:p.V256V	Amino acid change:cDNA level change	ANNOVAR annotator.	CharField	{'max_length': '100', 'null': 'True'}	ANN_AAChange_refGene	
8	ANN_AAChange_refGene_protein	p.M102T	PEX14:NM_004565:exon9:c.G768A:p.V256V	Amino acid change:protein level change	ANNOVAR annotator.	CharField	{'max_length': '100', 'null': 'True'}	ANN_AAChange_refGene	
9	ANN_cytoBand	1p36.33	1p36.22	The chromosome segment in which the variant site is located (observed by Giemas staining)	ANNOVAR annotator.	CharField	{'max_length': '200', 'null': 'True'}	No	
10	ANN_ExAC_ALL	0.8797	0.0124	The allele frequency of the variant in the Exome Aggregation Consortium (ExAC) database, all populations	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	
11	ANN_ExAC_AFR	0.7673	0.003	The allele frequency of the variant in the ExAC database, African/African American population	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	
12	ANN_ExAC_AMR	0.8125	0.0064	The allele frequency of the variant in the ExAC database, Latino population.	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	
13	ANN_ExAC_EAS	0.6241	0	The allele frequency of the variant in the ExAC database, East Asian population.	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	
14	ANN_ExAC_FIN	0.9004	0.0059	Allele frequency of the variant in the ExAC database, Finnish population.	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	
15	ANN_ExAC_NFE	0.9411	0.0191	The allele frequency of the variant in the ExAC database, non-Finnish European population	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	

16	ANN_ExAC_OTH	0.9115	0.0067	The allele frequency of the variant in the ExAC database, other populations	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	
17	ANN_ExAC_SAS	0.8605	0.0051	The allele frequency of the variant in the ExAC database, South Asian population	ANNOVAR annotator.	FloatField	{'null': 'True'}	No	
18	ANN_avsnp147	rs12755088	rs36083022	Annotate the variants with dbSNP identifiers	ANNOVAR annotator.	CharField	'parameters': {'max_length': '100', 'null': 'True'} 'component': 'freetext'	No	
19	ANN_SIFT_score	0.297	NaN	The SIFT score indicates the effect of the mutation on the protein sequence. The smaller the SIFT score, the more "bad", indicating that the SNP is likely to cause a change in protein structure or function;	ANNOVAR annotator (SIFT Algorithm) For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
20	ANN_SIFT_pred	T	NaN	D: Deleterious (sift<=0.05); T: tolerated (sift>0.05)）	ANNOVAR annotator (SIFT Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
21	ANN_Polyphen2_HDIV_score	0.013	NaN	Using PolyPhen2 based on HumanDiv database to predict the effect of this mutation on protein sequence, for complex diseases, the larger the value, the more "harmful", indicating that the SNP leads to a large possibility of protein structure or function change; damaging (0.453<=pp2_hdiv<=0.956) ; B: benign (pp2_hdiv<=0.452)) The score ranges from 0.0 (tolerated) to 1.0 (deleterious)	ANNOVAR annotator (Polyphen v2 Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	

22	ANN_Polyphen2_HDIV_pred	B	NaN	D or P or B (D: Probably damaging (>=0.957), P:may	ANNOVAR annotator (Polyphen v2 Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
23	ANN_Polyphen2_HVAR_score	0.004	NaN	PolyPhen2 was used to predict the effect of this mutation on protein sequences based on the HumanVar database for single-gene genetic diseases. The larger the value, the more "harmful", indicating that the SNP is likely to cause a change in protein structure or function;	ANNOVAR annotator (Polyphen v2 Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
24	ANN_Polyphen2_HVAR_pred	B	NaN	D or P or B (D: Probably damaging (>=0.909), P: possibly damaging (0.447<=pp2_hvar<=0.909); B: benign (pp2_hvar<=0.446))	ANNOVAR annotator (Polyphen v2 Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
25	ANN_LRT_score	0.135	NaN	The LRT score indicates the effect of the mutation on the protein sequence. The larger the value, the more "bad", indicating that the SNP is likely to cause a change in protein structure or function.	ANNOVAR annotator (LRT Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
26	ANN_LRT_pred	U	NaN	D, N or U (D: Deleterious; N: Neutral; U: Unknown).	ANNOVAR annotator (LRT Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
27	ANN_MutationTaster_score	1	NaN	The MutationTaster score indicates the effect of the mutation on the protein sequence. The larger the value, the more "bad", indicating that the SNP is likely to cause a change in protein structure or function. ("polymorphism_automatic"	ANNOVAR annotator (MutationTaster Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
28	ANN_MutationTaster_pred	P	NaN	A ("disease_causing_automatic"); "D" ("disease_causing");"N" ("polymorphism"); "P"	ANNOVAR annotator (MutationTaster Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	

29	ANN_MutationAssessor_pred	N	NaN	MutationAssessor predicts the classification according to the threshold: H is a pathogenic site with higher confidence, M is a moderately cripable pathogenic site, L is a low-confidence pathogenic site, and N is a harmless site. Point	ANNOVAR annotator (MutationAssessor Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
30	ANN_FATHMM_score	1.44	NaN	Pathogenicity score predicted by FATHMM software	ANNOVAR annotator (FATHMM Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
31	ANN_FATHMM_pred	T	NaN	Classification of FATHMM based on threshold: D is a highly credible pathogenic site, and P is a pathogenic site with a high degree of confidence.	ANNOVAR annotator (FATHMM Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
32	ANN_PROVEAN_score	0.21	NaN	predicts whether an amino acid substitution or indel has an impact on the biological function of a protein	ANNOVAR annotator (Protein Variation Effect Analyzer) https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
33	ANN_PROVEAN_pred	N	NaN	D: Deleterious;N: Neutral higher values are more deleterious	ANNOVAR annotator (Protein Variation Effect Analyzer) https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
34	ANN_VEST3_score	0.13	NaN	higher values are more deleterious	ANNOVAR annotator (VEST V3 Algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
35	ANN_CADD_raw_score	-1.067	NaN	Raw values do have relative meaning, with higher values indicating that a variant is more likely to be simulated and therefore more likely to have deleterious effects	ANNOVAR annotator (CADD Combined annotation dependent depletion). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	

36	ANN_CADD_phred	0.012	NaN	Raw values have been PHRED-scaled by expressing the rank in order of magnitude terms rather than the precise rank itself. For example, reference genome single nucleotide variants at the 10th-% of CADD scores are assigned to CADD-10, top 1% to CADD-20, top 0.1% to CADD-30, etc. The results of this transformation are the scaled CADD scores	ANNOVAR annotator (CADD Combined annotation dependent depletion). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
37	ANN_DANN_score	0.724	NaN	Deleterious Annotation of genetic variants using Neural Networks, higher values are more deleterious	ANNOVAR annotator (DANN algoirthm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
38	ANN_fathmm_MKL_coding_score	0.002	NaN	Predict the Functional Consequences of Non-Coding and Coding Single Nucleotide Variants (SNVs).Predictions are given as p-values in the range [0, 1]: values above 0.5 are predicted to be deleterious, while those below 0.5 are predicted to be neutral or benign	ANNOVAR annotator (FATHMM-MKL algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
39	ANN_fathmm_MKL_coding_pred	N	NaN	D: Deleterious;T: Tolerated Score >= 0.5: D; Score < 0.5: T	ANNOVAR annotator (FATHMM-MKL algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
40	ANN_MetaSVM_score	-1	NaN	Large value means the SNV is more likely to be damaging. Scores range from -2 to 3 in dbNSFP	ANNOVAR annotator (MetaSVM algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
41	ANN_MetaSVM_pred	T	NaN	D: Deleterious; T: Tolerated; higher scores are more deleterious	ANNOVAR annotator (MetaSVM algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	

42	ANN_MetaLR_score	0	NaN	Scoring method for deleterious missense mutations. The range of the score is between 0 and 1	ANNOVAR annotator (MetaLR algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
43	ANN_MetaLR_pred	T	NaN	D: Deleterious;T: Tolerated;higher scores are more deleterious	ANNOVAR annotator (MetaLR algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	CharField	{'max_length': '10', 'null': 'True'}	No	
44	ANN_integrated_fitCons_score	0.706	NaN	FitCons algorithm estimates the probability that a point mutation at each position in a genome will influence fitness. These fitness consequence (fitCons) scores serve as evolution-based measures of potential genomic function	ANNOVAR annotator (fitCons algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
45	ANN_integrated_confidence_value	0	NaN	A confidence interval is an interval in which a measurement or trial falls corresponding to a given probability	ANNOVAR annotator (fitCons algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	IntegerField	{'null': 'True'}	No	
46	ANN_GERP	-2.47	NaN	RS score, the larger the score, the more conserved the site	ANNOVAR annotator(Genome Evolutionary Rate Profiling ++). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
47	ANN_phyloP7way_vertebrate	-0.004	NaN	phyloP (phylogenetic p-values) conservation score based on the multiple alignments of 7 vertebrate genomes (including human). The larger the score, the more conserved the site	ANNOVAR annotator(PhyloP algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
48	ANN_phyloP20way_mammalian	-1.025	NaN	Genomes of 20 mammals, The larger the score, the more conserved the site	ANNOVAR annotator(PhyloP algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	

49	ANN_MutationAssessor_score	0	NaN	MutationAssessor predicted pathogenic score	ANNOVAR annotator(MutationAssessor algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
50	ANN_phastCons7way_vertebrate'phastCons	0.001	NaN	conservation score based on the multiple alignments of 7 vertebrate genomes (including human). The larger the score, the more conserved the site	ANNOVAR annotator(phastCons algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
51	ANN_phastCons20way_mammalian	0.002	NaN	genomes of 20 mammals (including human). The larger the score, the more conserved the site	ANNOVAR annotator(phastCons algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
52	ANN_SiPhy_29way_logOdds	4.729	NaN	The SiPhy score based on 29 mammals genomes. The larger the score, the more conserved the site. Scores ranges from 0 to 37.9718.	ANNOVAR annotator(SiPhy algorithm). For more info https://brb.nci.nih.gov/seqtools/colexpanno.html#dbnsfp	FloatField	{'null': 'True'}	No	
53	ANN_CLNALLELEID	Nan	249355	The ClinVar Allele ID for the variant	ANNOVAR annotator (ClinVar Database)	CharField	{'max_length': '100', 'null': 'True'}	No	
54	ANN_CLNDN	Nan	Zellweger_syndrome|not_specified|not_provided	Clinical Disease Name. A string consisting of the disease name used by the database specified by CLNDISDB	ANNOVAR annotator	CharField	{'max_length': '10', 'null': 'True'}	ANN_CLINVAR	x.split('|')
55	ANN_CLNDISDB	Nan	MedGen:C0043459,OMIM:214100,Orphanet:ORPHA912,SNOMED_CT:88469006|MedGen:CN169374|MedGen:CN517202	Clinical Source Database and ID. A string that includes pairs of the variant's disease database name and identifier separated by a colon (:), with each pair delimited by a vertical bar (|)	ANNOVAR annotator	CharField	{'max_length': '200', 'null': 'True'}	ANN_CLINVAR	x.split('|')

56	ANN_CLNREVSTAT	Nan	criteria_provided,_conflicting_interpretations;	Clinical Review Status. Integer that represents ClinVar Review Status. One of the following values may be assigned: 1 no_assertion (No asserion provided), 2 no_criteria (No assertion criteria provided), 3 single (Criteria provided single submitter), 4 mult (Criteria provided multiple submitters no conflict) 5 conf (Criteria provided conflicting interpretations), 6 exp (Reviewed by expert panel), 7 guideline (Practice guideline)	ANNOVAR annotator (ClinVar Database)	CharField	{'max_length': '200', 'null': 'True'}	No	
57	ANN_CLNSIG	Nan	Conflicting_interpretations_of_pathogenicity	Clinical Significance. A string that describes the variant's clinical significance. One of the following values may be assigned: 0 unknown, 1 untested, 2 nonpathogenic, 3 probable-nonpathogenic, 4 probable-pathogenic, 5 pathogenic, 6 drug-response, 7 histocompatibility, 255 other	ANNOVAR annotator (ClinVar Database)	CharField	{'max_length': '200', 'null': 'True'}	No	

Table 3: **An analytic description how Zazz manages the annotated fields from a file which is output of VEP [4] [5]. The description of each column is the same as in Table 1.**

T

	Column	Example Value		Description	Source	Django Field type	Django Field Parameters	Multi table	Formatter
	#locus	chr1:1987993	Chr1: 100949860						
0	VEP_Allele	T/C	G/A	pair of alleles separated by a '/', with the reference allele first	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
1	VEP_AA_AF	C	A	The variant allele used to calculate the consequence	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
2	VEP_Consequence	synonymous_variant	synonymous_variant	Consequence of this variant on the protein sequence (e.g. stop gained, missense, stop lost, frameshift)	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
3	VEP_IMPACT	LOW	LOW	The impact of a variant on a transcript (modifier or low for example)	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
4	VEP_SYMBOL	PRKCZ	CDC14A	The gene symbol	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
5	VEP_Gene	ENSG00000067606	ENSG00000079335	Ensembl stable ID of affected gene	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

[4] http://avianbase.narf.ac.uk/info/docs/tools/vep/vep_formats.html
[5] https://www.ebi.ac.uk/gene2phenotype/output.txt

6	VEP_Feature_type	Transcript	Transcript	Ensembl stable ID of feature	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
7	VEP_Feature	ENST00000378567	ENST00000336454	Type of feature. Currently one of Transcript, RegulatoryFeature, MotifFeature	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
8	VEP_BIOTYPE	protein_coding	protein_coding	Biotype of transcript or regulatory feature	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
9	VEP_EXON	3/18	11 /16	The exon number (out of total number)	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
10	VEP_INTRON	NaN	NaN	The intron number (out of total number)	VEP Annnotator	CharField	max_length': '100', 'null': 'True'	VEP_MULTI	x.split('|')
11	VEP_HGVSc	ENST00000378567.3:c.264T>C	ENST00000336454.3:c.990G>A	The HGVS coding sequence name	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
12	VEP_HGVSp	ENSP00000367830.3:p.Asp88%3D	ENSP00000336739.3:p.Ser330%3D	The HGVS protein sequence name	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
13	VEP_cDNA_position	425	1345	The relative position of base pair in cDNA sequence	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
14	VEP_CDS_position	264	990	The relative position of base pair in coding sequence	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

15	VEP_Amino_acids	D	S	Reference and variant amino acids	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
16	VEP_Codons	gaT/gaC	tcG/tcA	Reference and variant codon sequence	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
17	VEP_Existing_variation	rs12184	rs2270694	Identifier(s) of co-located known variants	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
18	VEP_DISTANCE	NaN	NaN	Shortest distance from variant to transcript	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
19	VEP_STRAND	1	1	Defined as + (forward) or - (reverse	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
20	VEP_FLAGS	NaN	NaN	Transcript quality flags	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
21	VEP_VARIANT_CLASS	SNV	SNV	They define the class of a variant according to its component alleles and its mapping to the reference genome (for example SNV, copy_number_loss etc)	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
22	VEP_SYMBOL_SOURCE	HGNC	HGNC	Gene_symbol_source	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

23	VEP_HGNC_ID	9412	1718	identifier for a gene from the HGNC database	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
24	VEP_CANONICAL	YES	YES	A flag indicating if the transcript is denoted as the canonical transcript for this gene	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
25	VEP_TSL	NaN	NaN	The Transcript Support Level (TSL) is a method to highlight the well-supported and poorly-supported transcript models for users, based on the type and quality of the alignments used to annotate the transcript.	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
26	VEP_APPRIS	NaN	NaN	APPRIS is a system to annotate alternatively spliced transcripts based on a range of computational methods to identify the most functionally important transcript(s) of a gene.	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

27	VEP_CCDS	CCDS37.1	CCDS770.1	The CCDS identifier shows when the variant have been annotated.The lower the CCDS identifier, the earlier it was annotated. It is connected with the APPRIS field. When, APPRIS core modules are unable to choose a clear principal variant and there more than one of the variants have distinct CCDS identifiers, APPRIS selects the variant with lowest CCDS value	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
28	VEP_ENSP	ENSP00000367830	ENSP00000359142	the Ensembl protein identifier of the affected transcript	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
29	VEP_SWISSPROT	Q05513	Q9UNH5	UniProtKB/Swiss-Prot identifier of protein product	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
30	VEP_TREMBL	J3KRP7	NaN	UniProtKB/TrEMBL identifier of protein product	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
31	VEP_UNIPARC	UPI0000169EB7	UPI000006FD73	UniParc identifier of protein product	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

32	VEP_GENE_PHENO	NaN	1	Indicates if the overlapped gene is associated with a phenotype, disease or trait	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
33	VEP_SIFT	NaN	NaN	Indicates if the overlapped gene is associated with a phenotype, disease or trait.	VEP Annnotator (sift="sift5.2.2")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
34	VEP_PolyPhen	NaN	NaN	PolyPhen tool is applied for human only. It is a tool which predicts possible impact of an amino acid substitution on the structure and function of a human protein using straightforward physical and comparative considerations.VEP can output the prediction term, score or both.	VEP Annnotator (polyphen="2.2.2")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
35	VEP_DOMAINS	PIRSF_domain:PIRSF000554	hmmpanther:PTHR23339:SF62	The source and identifer of any overlapping protein domains	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
36	VEP_miRNA	NaN	NaN	A small RNA (~22bp) that silences the expression of target mRNA	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

37	**VEP_HGVS_OFFSET**	**NaN**	**NaN**	**The flag HGVS_OFFSET is set to the number of bases by which the variant has shifted, relative to the input genomic coordinates**	**VEP Annnotator**	**CharField**	**{'max_length': '100', 'null': 'True'}**	**VEP_MULTI**	**x.split('|')**
38	**VEP_AF**	**NaN**	**0.0883**	**The global allele frequency (AF) from 1000 Genomes Phase 3 data for any known co-located variant to the output.**	**VEP Annnotator**	**CharField**	**{'max_length': '100', 'null': 'True'}**	**VEP_MULTI**	**x.split('|')**
39	**VEP_AFR_AF**	**0.8873**	**0.059**	**The allele frequency for the African population from 1000 Genomes.**	**VEP Annnotator**	**CharField**	**{'max_length': '100', 'null': 'True'}**	**VEP_MULTI**	**x.split('|')**
40	**VEP_AMR_AF**	**0.402**	**0.0965**	**The allele frequency for the American population from 1000 Genomes**	**VEP Annnotator**	**CharField**	**{'max_length': '100', 'null': 'True'}**	**VEP_MULTI**	**x.split('|')**
41	**VEP_EAS_AF**	**0.9454**	**0.0625**	**The allele frequency for the East Asian population from 1000 Genomes**	**VEP Annnotator**	**CharField**	**{'max_length': '100', 'null': 'True'}**	**VEP_MULTI**	**x.split('|')**
42	**VEP_EUR_AF'**	**0.3429**	**0.1441**	**The allele frequency for the European population from 1000 Genomes**	**VEP Annnotator**	**CharField**	**{'max_length': '100', 'null': 'True'}**	**VEP_MULTI**	**x.split('|')**
43	**VEP_SAS_AF**	**0.6186**	**0.091**	**The allele frequency for the South Asian and African populations from 1000 Genomes**	**VEP Annnotator**	**CharField**	**{'max_length': '100', 'null': 'True'}**	**VEP_MULTI**	**x.split('|')**

44	VEP_AA_AF	0.8191|	0.05838	The allele frequency for American and African population from NHLBI-ESP for	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
45	VEP_EA_AF	0.3436	0.1331	The allele frequency for African and from NHLBI-ESP for European_American and African population	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
46	VEP_gnomAD_AF	0.4571	0.117	Frequency of existing variant in gnomAD exomes combined population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
47	VEP_gnomAD_AFR_AF	0.8382	0.05718	Frequency of existing variant in gnomAD exomes African/American population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
48	VEP_gnomAD_AMR_AF	0.4405	0.2138	Frequency of existing variant in gnomAD exomes American population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
49	VEP_gnomAD_ASJ_AF	0.4447	0.08684	Frequency of existing variant in gnomAD exomes Ashkenazi Jewish population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
50	VEP_gnomAD_EAS_AF	0.9519	0.2138	Frequency of existing variant in gnomAD exomes East Asian population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
51	VEP_gnomAD_FIN_AF	0.3321	0.06245	Frequency of existing variant in gnomAD exomes Finnish population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

52	VEP_gnomAD_NFE_AF	0.3279	0.1685	Frequency of existing variant in gnomAD exomes Non-Finnish European population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
53	VEP_gnomAD_OTH_AF	0.3279	0.1299	Genome aggregation database exomes: other and African populations	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
54	VEP_gnomAD_SAS_AF	0.5555	0.1178	Frequency of existing variant in gnomAD exomes South Asian population	VEP Annnotator (gnomAD="r2.1")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
55	VEP_MAX_AF	0.9519	0.08647	Report the highest allele frequency observed in any population from 1000 genomes, ESP or gnomAD	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
56	VEP_MAX_AF_POPS	gnomAD_EAS	gnomAD_ASJ	Populations in which maximum allele frequency was observed		CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
57	VEP_CLIN_SIG	NaN	benig N	ClinVar clinical significance of the dbSNP variant	VEP Annnotator (ClinVar="201810")	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
58	VEP_SOMATIC	NaN	NaN	Somatic status of existing variant(s); multiple values correspond to multiple values in the Existing_variation field	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')

59	VEP_PHENO	NaN	NaN	Indicates if existing variant is associated with a phenotype, disease or trait; multiple values correspond to multiple values in the Existing_variation field	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
60	VEP_PUBMED	NaN	NaN	Pubmed ID(s) of publications that cite existing variant	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
61	VEP_MOTIF_NAME	NaN	NaN	The source and identifier of a transcription factor binding profile aligned at this position	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
62	VEP_MOTIF_POS	NaN	NaN	The relative position of the variation in the aligned TFBP	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
63	VEP_HIGH_INF_POS	NaN	NaN	A flag indicating if the variant falls in a high information position of a transcription factor binding profile (TFBP)	VEP Annnotator	CharField	{'max_length': '100', 'null': 'True'}	VEP_MULTI	x.split('|')
64	VEP_MOTIF_SCORE_CHANGE	NaN	NaN	The difference in motif score of the reference and variant sequences for the TFBP	VEP Annnotator	FloatField	{'null': 'True'}	VEP_MULTI	x.split('|')